

Algorithms for Approximation

A. Iske J. Levesley
Editors

Algorithms for Approximation

Proceedings of the 5th International
Conference, Chester, July 2005

With 85 Figures and 21 Tables

 Springer

Armin Iske
Universität Hamburg
Department Mathematik
Bundesstraße 55
20146 Hamburg, Germany
E-mail: iske@math.uni-hamburg.de

Jeremy Levesley
University of Leicester
Department of Mathematics
University Road
Leicester LE1 7RH, United Kingdom
E-mail: jl1@mcs.le.ac.uk

The contribution by Alistair Forbes “Algorithms for Structured Gauss-Markov Regression” is reproduced by permission of the Controller of HMSO, © Crown Copyright 2006

Mathematics Subject Classification (2000): 65Dxx, 65D15, 65D05, 65D07, 65D17

Library of Congress Control Number: 2006934297

ISBN-10 3-540-33283-9 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-33283-1 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable for prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media
springer.com
© Springer-Verlag Berlin Heidelberg 2007

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting by the authors using a Springer L^AT_EX macro package
Cover design: *design & production* GmbH, Heidelberg

Printed on acid-free paper SPIN: 11733195 46/SPi 5 4 3 2 1 0

Preface

Approximation methods are of vital importance in many challenging applications from computational science and engineering. This book collects papers from world experts in a broad variety of relevant applications of approximation theory, including pattern recognition and machine learning, multiscale modelling of fluid flow, metrology, geometric modelling, the solution of differential equations, and signal and image processing, to mention a few.

The 30 papers in this volume document new trends in approximation through recent theoretical developments, important computational aspects and multidisciplinary applications, which makes it a perfect text for graduate students and researchers from science and engineering who wish to understand and develop numerical algorithms for solving their specific problems. An important feature of the book is to bring together modern methods from statistics, mathematical modelling and numerical simulation for solving relevant problems with a wide range of inherent scales. Industrial mathematicians, including representatives from Microsoft and Schlumberger make contributions, which fosters the transfer of the latest approximation methods to real-world applications.

This book grew out of the fifth in the conference series on *Algorithms for Approximation*, which took place from 17th to 21st July 2005, in the beautiful city of Chester in England. The conference was supported by the National Physical Laboratory and the London Mathematical Society, and had around 90 delegates from over 20 different countries.

The book has been arranged in six parts:

- Part I.** Imaging and Data Mining;
- Part II.** Numerical Simulation;
- Part III.** Statistical Approximation Methods;
- Part IV.** Data Fitting and Modelling;
- Part V.** Differential and Integral Equations;
- Part VI.** Special Functions and Approximation on Manifolds.

Part I grew out of a workshop sponsored by the London Mathematical Society on *Developments in Pattern Recognition and Data Mining* and includes contributions from Donald Wunsch, the President of the *International Neural Networks Society* and Chris Burges from Microsoft. The numerical solution of differential equations lies at the heart of practical application of approximation theory. The next two parts contain contributions in this direction. Part II demonstrates the growing trend in the transfer of approximation theory tools to the simulation of physical systems. In particular, radial basis functions are gaining a foothold in this regard. Part III has papers concerning the solution of differential equations, and especially delay differential equations. The realisation that statistical Kriging methods and radial basis function interpolation are two sides of the same coin has led to an increase in interest in statistical methods in the approximation community. Part IV reflects ongoing work in this direction. Part V contains recent developments in traditional areas of approximation theory, in the modelling of data using splines and radial basis functions. Part VI is concerned with special functions and approximation on manifolds such as spheres.

We are grateful to all the authors who have submitted for this volume, especially for their patience with the editors. The contributions to this volume have all been refereed, and thanks go out to all the referees for their timely and considered comments. Finally, we very much appreciate the cordial relationship we have had with Springer-Verlag, Heidelberg, through Martin Peters.

Leicester, June 2006

*Armin Iske
Jeremy Levesley*

Contents

Part I Imaging and Data Mining

Ranking as Function Approximation <i>Christopher J.C. Burges</i>	3
Two Algorithms for Approximation in Highly Complicated Planar Domains <i>Nira Dyn, Roman Kazinnik</i>	19
Computational Intelligence in Clustering Algorithms, With Applications <i>Rui Xu, Donald Wunsch II</i>	31
Energy-Based Image Simplification with Nonlocal Data and Smoothness Terms <i>Stephan Didas, Pavel Mrázek, Joachim Weickert</i>	51
Multiscale Voice Morphing Using Radial Basis Function Analysis <i>Christina Orphanidou, Irene M. Moroz, Stephen J. Roberts</i>	61
Associating Families of Curves Using Feature Extraction and Cluster Analysis <i>Jane L. Terry, Andrew Crampton, Chris J. Talbot</i>	71

Part II Numerical Simulation

Particle Flow Simulation by Using Polyharmonic Splines <i>Armin Iske</i>	83
--	----

VIII Contents

Enhancing SPH using Moving Least-Squares and Radial Basis Functions <i>Robert Brownlee, Paul Houston, Jeremy Levesley, Stephan Rosswog</i>	103
Stepwise Calculation of the Basin of Attraction in Dynamical Systems Using Radial Basis Functions <i>Peter Giesl</i>	113
Integro-Differential Equation Models and Numerical Methods for Cell Motility and Alignment <i>Athena Makroglou</i>	123
Spectral Galerkin Method Applied to Some Problems in Elasticity <i>Chris J. Talbot</i>	135
<hr/>	
Part III Statistical Approximation Methods	
<hr/>	
Bayesian Field Theory Applied to Scattered Data Interpolation and Inverse Problems <i>Chris L. Farmer</i>	147
Algorithms for Structured Gauss-Markov Regression <i>Alistair B. Forbes</i>	167
Uncertainty Evaluation in Reservoir Forecasting by Bayes Linear Methodology <i>Daniel Busby, Chris L. Farmer, Armin Iske</i>	187
<hr/>	
Part IV Data Fitting and Modelling	
<hr/>	
Integral Interpolation <i>Rick K. Beatson, Michael K. Langton</i>	199
Shape Control in Powell-Sabin Quasi-Interpolation <i>Carla Manni</i>	219
Approximation with Asymptotic Polynomials <i>Philip Cooper, Alistair B. Forbes, John C. Mason</i>	241
Spline Approximation Using Knot Density Functions <i>Andrew Crampton, Alistair B. Forbes</i>	249
Neutral Data Fitting by Lines and Planes <i>Tim Goodman, Chris Tofallis</i>	259

Approximation on an Infinite Range to Ordinary Differential Equations Solutions by a Function of a Radial Basis Function
Damian P. Jenkinson, John C. Mason 269

Weighted Integrals of Polynomial Splines
Mladen Rogina 279

Part V Differential and Integral Equations

On Sequential Estimators for Affine Stochastic Delay Differential Equations
Uwe Küchler, Vyacheslav Vasiliev 287

Scalar Periodic Complex Delay Differential Equations: Small Solutions and their Detection
Neville J. Ford, Patricia M. Lumb 297

Using Approximations to Lyapunov Exponents to Predict Changes in Dynamical Behaviour in Numerical Solutions to Stochastic Delay Differential Equations
Neville J. Ford, Stewart J. Norton 309

Superconvergence of Quadratic Spline Collocation for Volterra Integral Equations
Darja Saveljeva 319

Part VI Special Functions and Approximation on Manifolds

Asymptotic Approximations to Truncation Errors of Series Representations for Special Functions
Ernst Joachim Weniger 331

Strictly Positive Definite Functions on Generalized Motion Groups
Wolfgang zu Castell, Frank Filbir 349

Energy Estimates and the Weyl Criterion on Compact Homogeneous Manifolds
Steven B. Damelin, Jeremy Levesley, Xingping Sun 359

Minimal Discrete Energy Problems and Numerical Integration on Compact Sets in Euclidean Spaces
Steven B. Damelin, Viktor Maymeskul 369

Numerical Quadrature of Highly Oscillatory Integrals Using Derivatives <i>Sheehan Olver</i>	379
Index	387

List of Contributors

Rick K. Beatson
University of Canterbury
Dept of Mathematics and Statistics
Christchurch 8020, New Zealand
R.Beatson@math.canterbury.ac.nz

Christopher J.C. Burges
Microsoft Research
One Microsoft Way
Redmond, WA 98052-6399, U.S.A.
cburges@microsoft.com

Daniel Busby
Schlumberger
Abingdon Technology Center
Abingdon OX14 1UJ, UK
dbusby4@slb.com

Robert Brownlee
University of Leicester
Department of Mathematics
Leicester LE1 7RH, UK
r.brownlee@mcs.le.ac.uk

Wolfgang zu Castell
GSF - National Research Center for
Environment and Health
D-85764 Neuherberg, Germany
castell@gsf.de

Philip Cooper
University of Huddersfield
School of Computing and Engineering
Huddersfield HD1 3DH, UK
p.cooper@hud.ac.uk

Andrew Crampton
University of Huddersfield
School of Computing and Engineering
Huddersfield HD1 3DH, UK
a.crampton@hud.ac.uk

Steven B. Damelin
University of Minnesota
Institute Mathematics & Applications
Minneapolis, MN 55455, U.S.A.
damelin@ima.umn.edu

Stephan Didas
Saarland University
Mathematics and Computer Science
D-66041 Saarbrücken, Germany
didas@ia.uni-saarland.de

Nira Dyn
Tel-Aviv University
School of Mathematical Sciences
Tel-Aviv 69978, Israel
niradyn@post.tau.ac.il

Chris L. Farmer
Schlumberger
Abingdon Technology Center
Abingdon OX14 1UJ, UK
farmer5@slb.com

XII List of Contributors

Frank Filbir

GSF - National Research Center for
Environment and Health
D-85764 Neuherberg, Germany
filbir@gsf.de

Alistair B. Forbes

National Physical Laboratory
Teddington TW11 0LW, UK
alistair.forbes@npl.co.uk

Neville J. Ford

University of Chester
Department of Mathematics
Chester CH1 4BJ, UK
njford@chester.ac.uk

Peter Giesl

Munich University of Technology
Department of Mathematics
D-85747 Garching, Germany
giesl@ma.tum.de

Tim Goodman

University of Dundee
Department of Mathematics
Dundee DD1 5RD, UK
tgoodman@maths.dundee.ac.uk

Paul Houston

University of Nottingham
School of Mathematical Sciences
Nottingham NG7 2RD, UK
paul.houston@nottingham.ac.uk

Armin Iske

University of Hamburg
Department of Mathematics
D-20146 Hamburg, Germany
iske@math.uni-hamburg.de

Damian P. Jenkinson

University of Huddersfield
School of Computing and Engineering
Huddersfield HD1 3DH, UK
d.p.jenkinson@hud.ac.uk

Roman Kazinnik

Tel-Aviv University
School of Mathematical Sciences
Tel-Aviv 69978, Israel
romank@post.tau.ac.il

Uwe K uchler

Humboldt University Berlin
Institute of Mathematics
D-10099 Berlin, Germany
kuechler@math.hu-berlin.de

Michael K. Langton

University of Canterbury
Dept of Mathematics and Statistics
Christchurch 8020, New Zealand

Jeremy Levesley

University of Leicester
Department of Mathematics
Leicester LE1 7RH, UK
j.levesley@mcs.le.ac.uk

Patricia M. Lumb

University of Chester
Department of Mathematics
Chester CH1 4BJ, UK
p.lumb@chester.ac.uk

Athena Makroglou

University of Portsmouth
Department of Mathematics
Portsmouth, Hampshire PO1 3HF, UK
athena.makroglou@port.ac.uk

Carla Manni

University of Rome "Tor Vergata"
Department of Mathematics
00133 Roma, Italy
manni@mat.uniroma2.it

John C. Mason

University of Huddersfield
School of Computing and Engineering
Huddersfield HD1 3DH, UK
j.c.mason@hud.ac.uk

Viktor Maymeskul

Georgia Southern University
Department of Mathematical Sciences
Georgia 30460, U.S.A.
vmaymesk@georgiasouthern.edu

Irene M. Moroz

University of Oxford
 Industrial and Applied Mathematics
 Oxford OX1 3LB, UK
 moroz@maths.ox.ac.uk

Pavel Mrázek

Upek R&D s.r.o., Husinecka 7
 130 00 Prague 3, Czech Republic
 pavel.mrazek@upek.com

Stewart J. Norton

University of Chester
 Department of Mathematics
 Chester CH1 4BJ, UK
 s.norton@chester.ac.uk

Sheehan Olver

University of Cambridge
 Applied Mathematics & Theor. Physics
 Cambridge CB3 0WA, UK
 S.Olver@damtp.cam.ac.uk

Christina Orphanidou

University of Oxford
 Industrial and Applied Mathematics
 Oxford OX1 3LB, UK
 orphanid@maths.ox.ac.uk

Stephen J. Roberts

University of Oxford
 Pattern Analysis & Machine Learning
 Oxford OX1 3PJ, UK
 sjrob@robots.ox.ac.uk

Mladen Rogina

University of Zagreb
 Department of Mathematics
 10002 Zagreb, Croatia
 rogina@math.hr

Stephan Rosswog

International University Bremen
 School of Engineering and Science
 D-28759 Bremen, Germany
 s.rosswog@iu-bremen.de

Darja Saveljeva

University of Tartu
 Institute of Applied Mathematics
 Tartu 50409, Estonia
 darja.saveljeva@ut.ee

Xingping Sun

Missouri State University
 Department of Mathematics
 Springfield, MO 65897, U.S.A.
 XSun@MissouriState.edu

Chris J. Talbot

University of Huddersfield
 School of Computing and Engineering
 Huddersfield HD1 3DH, UK
 c.j.talbot@hud.ac.uk

Jane L. Terry

University of Huddersfield
 School of Computing and Engineering
 Huddersfield HD1 3DH, UK
 j.l.terry@hud.ac.uk

Chris Tofallis

University of Hertfordshire
 Business School
 Hatfield, Herts AL10 9AB, UK
 c.tofallis@herts.ac.uk

Vyacheslav Vasiliev

University of Tomsk
 Applied Mathematics and Cybernetics
 634050 Tomsk, Russia
 vas@mail.tsu.ru

Joachim Weickert

Saarland University
 Mathematics and Computer Science
 D-66041 Saarbrücken, Germany
 weickert@mia.uni-saarland.de

Ernst Joachim Weniger

University of Regensburg
 Physical and Theoretical Chemistry
 D-93040 Regensburg, Germany
 joachim.weniger@chemie.uni-regensburg.de

Donald Wunsch II

University of Missouri
 Applied Computational Intelligence Lab
 Rolla, MO 65409-0249, U.S.A.
 dwunsch@umr.edu

Rui Xu

University of Missouri
 Applied Computational Intelligence Lab
 Rolla, MO 65409-0249, U.S.A.
 rxu@umr.edu

Part I

Imaging and Data Mining

Ranking as Function Approximation

Christopher J.C. Burges

Microsoft Research, One Microsoft Way, Redmond, WA 98052-6399, U.S.A.,
cburges@microsoft.com

Summary. An overview of the problem of learning to rank data is given. Some current machine learning approaches to the problem are described. The cost functions used to assess the quality of a ranking algorithm present particular difficulties: they are non-differentiable (as a function of the scores output by the ranker) and multivariate (in the sense that the cost associated with one ranked object depends on its relations to several other ranked objects). I present some ideas on a general framework for training using such cost functions; the approach has an appealing physical interpretation. The paper is tutorial in the sense that it is not assumed that the reader is familiar with the methods of machine learning; my hope is that the paper will encourage applied mathematicians to explore this topic.

1 Introduction

The field of machine learning draws from many disciplines, but ultimately the task is often one of function approximation: for classification, regression estimation, time series estimation, clustering, or more complex forms of learning, an attempt is being made to find a function that meets given criteria on some data. Because the machine learning enterprise is multi-disciplinary, it has much to gain from more established fields such as approximation theory, statistical and mathematical modeling, and algorithm design. In this paper, in the hope of stimulating more interaction between our communities, I give a review of approaches to one problem of growing interest in the machine learning community, namely, ranking. Ranking is needed whenever an algorithm returns a set of results upon which one would like to impose an order: for example, commercial search engines must rank millions of URLs in real time to help users find what they are looking for, and automated Question-Answering systems will often return a few top-ranked answers from a long list of possible answers. Ranking is also interesting in that it bridges the gap between traditional machine learning (where, for example, a sample is to be classified into one of two classes), and another area that is attracting growing interest, namely that of modeling structured data (as inputs, outputs, or both), for

example for data structures such as graphs. In this light, I will also present some new ideas on models for handling structured output data.

1.1 Notation

To make the discussion concrete and to establish notation, I will use the example of ranking search results. There, the task is the following: a query Q is issued by a user. Q may be thought of as a text string, but it may also contain other kinds of data. The search engine examines a large set of previously gathered documents, and for each document D , constructs a feature vector $F(Q, D) \in \mathbb{R}^n$. Thus, the i th element of F is itself a function $f_i : \{Q, D\} \mapsto \mathbb{R}$, and f_i has been constructed to encapsulate some aspect of how relevant the document D is to the query Q ¹. The feature vector F is then input to a ranking algorithm \mathcal{A} , which outputs a scalar “score”: $\mathcal{A} : F \in \mathbb{R}^n \mapsto s \in \mathbb{R}$. We will denote the number of queries for a given dataset by N_Q and the number of documents returned for the i ’th query by n_i . During the training phase, a set of labeled data $\{Q_i, D_{ij}, l_{ij}, i = 1, \dots, N_Q, j = 1, \dots, n_i\}$ is used to minimize a cost function C . Here the labels l encode the relevance of document D_{ij} for the query Q_i , and take integer values, where for a given query Q , $l_1 > l_2$ means that the document with label l_1 is more relevant to Q than that with label l_2 (note that the labels l really attach to document-query pairs, since a given document may be relevant for one query but not for another). The form that the cost function C takes varies from one algorithm to another, but its range is always the reals; the training process aims to find those parameters in the function \mathcal{A} that minimize the sample expectation of the cost over the training set. Once such a function \mathcal{A} has been found, its parameters are fixed, and its output scores s are used to map feature vectors F to the reals, where $A(F(Q, D_1)) > A(F(Q, D_2))$ is taken to mean that, for query Q , document D_1 is to be ranked higher than document D_2 . We will encapsulate this last relation using the symbol \triangleright , so that $A(F(Q, D_1)) > A(F(Q, D_2)) \Rightarrow D_1 \triangleright D_2$.

1.2 Representing the Ranking Problem as a Graph

[11] provide a very general framework for ranking using directed graphs, where an arc from A to B means that A is to be ranked higher than B. Note that for ranking algorithms that train on pairs, all such sets of relations can be captured by specifying a set of training pairs, which amounts to specifying the arcs in the graph. This approach can represent arbitrary ranking functions, in particular, ones that are inconsistent - for example $A \triangleright B$, $B \triangleright C$, $C \triangleright A$. Such inconsistent rankings can easily arise when mapping multivariate measurements to one dimensional ranking, as the following toy example illustrates:

¹ In fact, some elements of the feature vector may depend only on the document D , in order to capture the notion that some documents are unlikely to be relevant for any possible query.

imagine that a psychologist has devised an aptitude test². Mathematician A is considered stronger than mathematician B if, given three particular theorems, A can prove at least two theorems faster than B . The psychologist finds the measurements shown in Table 1.

	Minutes Per Proof		
<i>Mathematician</i>	<i>Theorem 1</i>	<i>Theorem 2</i>	<i>Theorem 3</i>
Archimedes	8	1	6
Bryson	3	5	7
Callippus	4	9	2

Table 1. Archimedes is stronger than Bryson; Bryson is stronger than Callippus; but Callippus is stronger than Archimedes.

2 Measures of Ranking Quality

In the information retrieval literature, there are many methods used to measure the quality of ranking results. Here we briefly describe four. We observe that there are two properties that are shared by all of these cost functions: none are differentiable, and all are multivariate, in the sense that they depend on the scores of multiple documents. The non-differentiability presents particular challenges to the machine learning approach, where cost functions are almost always assumed to be smooth. Recently, some progress has been made tackling the latter property using support vector methods [19]; below, we will outline an alternative approach.

Pair-wise Error

The pair-wise error counts the number of pairs that are in the incorrect order, as a fraction of the maximum possible number of such pairs.

Normalized Discounted Cumulative Gain (NDCG)

The normalized discounted cumulative gain measure [17] is a cumulative measure of ranking quality (so a suitable cost would be 1-NDCG). For a given query Q_i the NDCG is computed as

$$\mathcal{N}_i \equiv N_i \sum_{j=1}^L (2^{r^{(j)}} - 1) / \log(1 + j)$$

² Of course this “magic-square” example is not serious, although it illustrates the perils of one-dimensional thinking.

where $r(j)$ is the relevance level of the j 'th document, and where the normalization constant N_i is chosen so that a perfect ordering would result in $\mathcal{N}_i = 1$. Here L is the ranking level at which the NDCG is computed. The \mathcal{N}_i are then averaged over the query set.

Mean Reciprocal Rank (MRR)

This metric applies to the binary relevance task, where for a given query, and for a given document returned for that query, label “1” means “relevant” and “0”, “not relevant”. If r_i is the rank of the highest ranking relevant document for the i 'th query, then the reciprocal rank measure for that query is $1/r_i$, and the MRR is just the reciprocal rank, averaged over queries:

$$\text{MRR} = \frac{1}{N_Q} \sum_{i=1}^{N_Q} 1/r_i$$

MRR was used, for example, in TREC evaluations of Question Answering systems, before 2002 [25].

Winner Takes All (WTA)

This metric also applies to the binary relevance task. If the top ranked document for a given query is relevant, the WTA cost is zero, otherwise it is one; for N_Q queries we again take the mean:

$$\text{WTA} = \frac{1}{N_Q} \sum_{i=1}^{N_Q} \delta(l_{i1}, 1)$$

where δ here is the Kronecker delta. WTA is used, for example, in TREC evaluations of Question Answering systems, after 2002 [26].

3 Support Vector Ranking

Support vector machines for ordinal regression were proposed by [13] and further explored by [18] and more recently by [7]. The approach uses pair-based training. For convenience let us write the feature vector for a given query-document pair as $\mathbf{x} \equiv F(Q, D)$, where indices Q and D on \mathbf{x} are understood, and let us represent the training data as a set of pairs $\{\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}\}$, $i = 1, \dots, N$, where N is the total number of pairs in the training set, together with labels $z_i \in \{\pm 1\}$, $i = 1, \dots, N$, where $z_i = 1$ (-1) if $\mathbf{x}_i^{(1)}$ is to be ranked higher (lower) than $\mathbf{x}_i^{(2)}$. Note that each query can generate training pairs (and that a given feature vector \mathbf{x} can appear in several pairs), but that once the pairs have been generated, all that is needed for training is the set of pairs and their labels.

To solve the ranking problem we solve the following QP:

$$\min_{\mathbf{w}, \xi_i} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_i \xi_i \right\}$$

subject to:

$$\begin{aligned} z_i \mathbf{w} \cdot (\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)}) &> 1 - \xi_i \\ \xi_i &\in \mathbb{R}_+ \end{aligned}$$

In the separable case, by minimizing $\|\mathbf{w}\|$, we are maximizing the gap, projected along \mathbf{w} , between items that are to be ranked differently; the slack variables ξ_i allow for non-separable data, and their sum gives a bound on the number of errors. This is similar to the original formulation of Support Vector Machines for classification [10, 5], and enjoys the same advantages: the algorithm can be implicitly mapped to a feature space using the kernel trick (see, for example, [22]), which gives the model a great deal of expressive freedom, and uniform bounds on generalization performance can be given [13].

4 Perceptron Ranking

[9] propose a ranker based on the Perceptron ('PRank'), which maps a feature vector $\mathbf{x} \in \mathbb{R}^d$ to the reals with a learned vector $\mathbf{w} \in \mathbb{R}^d$ and increasing thresholds³ $b_r = 1, \dots, N$ such that the output of the mapping function is just $\mathbf{w} \cdot \mathbf{x}$, and such that the declared rank of \mathbf{x} is $\min_r \{\mathbf{w} \cdot \mathbf{x} - b_r < 0\}$. An alternative way to view this is that the rank of \mathbf{x} is defined by the bin into which $\mathbf{w} \cdot \mathbf{x}$ falls. The learning step is modeled after the Perceptron update rule (see [9] for details): a newly presented example \mathbf{x} results in a change in \mathbf{w} (and in the b_r) only if it falls in the wrong bin, given the current values of \mathbf{w} and the b_r . If this occurs, \mathbf{w} is updated by a quantity proportional to \mathbf{x} , and those thresholds whose movement could result in \mathbf{x} being correctly ranked are also updated. The linear form of PRank is an online algorithm⁴, in that it learns (that is, it updates the vector \mathbf{w} , and the thresholds that define the rank boundaries) using one example at a time. However, PRank can be, and has been, compared to batch ranking algorithms, and a quadratic kernel version was found to outperform all such algorithms described in [13]. [12] has proposed a simple but very effective extension of PRank, which approximates finding the Bayes point (that point which would give the minimum achievable generalization error) by averaging over PRank models.

³ Actually the last threshold is pegged at infinity.

⁴ The general kernel version is not, since the support vectors must be saved.

5 Neural Network Ranking

In this Section we describe a recent neural net based ranking algorithm that is currently used in one of the major commercial search engines [3]. Let's begin by defining a suitable cost.

5.1 A Probabilistic Cost

As we have observed, most machine learning algorithms require differentiable cost functions, and neural networks fall in this class. To this end, in [3] the following probabilistic model was proposed for modeling posteriors, where each training pair $\{A, B\}$ has associated posterior $P(A \triangleright B)$. The probabilistic model is an important feature of the approach, since ranking algorithms often model preferences, and the ascription of preferences is a much more subjective process than the ascription of, say, classes. (Target probabilities could be measured, for example, by measuring multiple human preferences for each pair.) We consider models where the learning algorithm is given a set of pairs of samples $[A, B]$ in \mathbb{R}^d , together with target probabilities \bar{P}_{AB} that sample A is to be ranked higher than sample B . As described above, this is a general formulation, in that the pairs of ranks need not be complete (in that taken together, they need not specify a complete ranking of the training data), or even consistent. We again consider models $\mathcal{A} : \mathbb{R}^d \mapsto \mathbb{R}$ such that the rank order of a set of test samples is specified by the real values that \mathcal{A} takes, specifically, $\mathcal{A}(\mathbf{x}_1) > \mathcal{A}(\mathbf{x}_2)$ is taken to mean that the model asserts that $\mathbf{x}_1 \triangleright \mathbf{x}_2$.

Denote the modeled posterior $P(\mathbf{x}_i \triangleright \mathbf{x}_j)$ by P_{ij} , $i, j = 1, \dots, m$, and let \bar{P}_{ij} be the desired target values for those posteriors. The cost function is a function of the difference of the system's outputs for each member of a pair of examples, which encapsulates the observation that for any given pair, an arbitrary offset can be added to the outputs without changing the final ranking. Define $o_i \equiv \mathcal{A}(\mathbf{x}_i)$ and $o_{ij} \equiv \mathcal{A}(\mathbf{x}_i) - \mathcal{A}(\mathbf{x}_j)$. The cost is a cross entropy cost function

$$C_{ij} \equiv C(o_{ij}) = -\bar{P}_{ij} \log P_{ij} - (1 - \bar{P}_{ij}) \log (1 - P_{ij})$$

where the map from outputs to probabilities are modeled using a logistic function

$$P_{ij} \equiv \frac{1}{1 + e^{-o_{ij}}}$$

The cross entropy cost has been shown to result in neural net outputs that model probabilities [6]. C_{ij} then becomes

$$C_{ij} = -\bar{P}_{ij} o_{ij} + \log(1 + e^{o_{ij}}) \quad (1)$$

Note that C_{ij} asymptotes to a linear function; for problems with noisy labels this is likely to be more robust than a quadratic cost. Also, when $\bar{P}_{ij} = \frac{1}{2}$

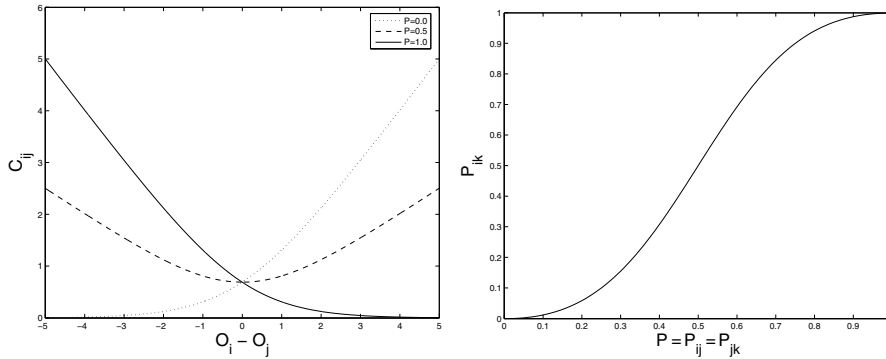


Fig. 1. Left: the cost function, for three values of the target probability. Right: combining probabilities.

(when no information is available as to the relative rank of the two patterns), C_{ij} becomes symmetric, with its minimum at the origin. This gives us a principled way of training on patterns that are desired to have the same rank. We plot C_{ij} as a function of o_{ij} in the left hand panel of Figure 1, for the three values $\bar{P} = \{0, 0.5, 1\}$.

Combining Probabilities

The above model puts consistency requirements on the \bar{P}_{ij} , in that we require that there exist ‘ideal’ outputs \bar{o}_i of the model such that

$$\bar{P}_{ij} \equiv \frac{1}{1 + e^{-\bar{o}_{ij}}} \quad (2)$$

where $\bar{o}_{ij} \equiv \bar{o}_i - \bar{o}_j$. This consistency requirement arises because if it is not met, then there will exist no set of outputs of the model that give the desired pairwise probabilities. The consistency condition leads to constraints on possible choices of the \bar{P} 's. For example, given \bar{P}_{ij} and \bar{P}_{jk} , Eq. (2) gives

$$\bar{P}_{ik} = \frac{\bar{P}_{ij}\bar{P}_{jk}}{1 + 2\bar{P}_{ij}\bar{P}_{jk} - \bar{P}_{ij} - \bar{P}_{jk}}$$

This is plotted in the right hand panel of Figure 1, for the case $\bar{P}_{ij} = \bar{P}_{jk} = P$. We draw attention to some appealing properties of the combined probability \bar{P}_{ik} . First, $\bar{P}_{ik} = P$ at the three points $P = 0$, $P = 0.5$ and $P = 1$, and only at those points. For example, if we specify that $P(A \triangleright B) = 0.5$ and that $P(B \triangleright C) = 0.5$, then it follows that $P(A \triangleright C) = 0.5$; complete uncertainty propagates. Complete certainty ($P = 0$ or $P = 1$) propagates similarly. Finally confidence, or lack of confidence, builds as expected: for $0 < P < 0.5$, then $\bar{P}_{ik} < P$, and for $0.5 < P < 1.0$, then $\bar{P}_{ik} > P$ (for example, if $P(A \triangleright B) = 0.6$, and $P(B \triangleright C) = 0.6$, then $P(A \triangleright C) > 0.6$). These considerations raise the

following question: given the consistency requirements, how much freedom is there to choose the pairwise probabilities? We have the following⁵

Theorem 1. *Given a sample set \mathbf{x}_i , $i = 1, \dots, m$ and any permutation \mathcal{Q} of the consecutive integers $\{1, 2, \dots, m\}$, suppose that an arbitrary target posterior $0 \leq \bar{P}_{kj} \leq 1$ is specified for every adjacent pair $k = \mathcal{Q}(i), j = \mathcal{Q}(i+1)$, $i = 1, \dots, m-1$. Denote the set of such \bar{P} 's, for a given choice of \mathcal{Q} , a set of 'adjacency posteriors'. Then specifying any set of adjacency posteriors is necessary and sufficient to uniquely identify a target posterior $0 \leq \bar{P}_{ij} \leq 1$ for every pair of samples $\mathbf{x}_i, \mathbf{x}_j$.*

Proof: Sufficiency: suppose we are given a set of adjacency posteriors. Without loss of generality we can relabel the samples such that the adjacency posteriors may be written $\bar{P}_{i,i+1}$, $i = 1, \dots, m-1$. From Eq. (2), \bar{o} is just the log odds:

$$\bar{o}_{ij} = \log \frac{\bar{P}_{ij}}{1 - \bar{P}_{ij}}$$

From its definition as a difference, any \bar{o}_{jk} , $j \leq k$, can be computed as $\sum_{m=j}^{k-1} \bar{o}_{m,m+1}$. Eq. (2) then shows that the resulting probabilities indeed lie in $[0, 1]$. Uniqueness can be seen as follows: for any i, j , \bar{P}_{ij} can be computed in multiple ways, in that given a set of previously computed posteriors $\bar{P}_{im_1}, \bar{P}_{m_1m_2}, \dots, \bar{P}_{m_nj}$, then \bar{P}_{ij} can be computed by first computing the corresponding \bar{o}_{kl} 's, adding them, and then using (2). However since $\bar{o}_{kl} = \bar{o}_k - \bar{o}_l$, the intermediate terms cancel, leaving just \bar{o}_{ij} , and the resulting \bar{P}_{ij} is unique. Necessity: if a target posterior is specified for every pair of samples, then by definition for any \mathcal{Q} , the adjacency posteriors are specified, since the adjacency posteriors are a subset of the set of all pairwise posteriors. \square

Although the above gives a straightforward method for computing \bar{P}_{ij} given an arbitrary set of adjacency posteriors, it is instructive to compute the \bar{P}_{ij} for the special case when all adjacency posteriors are equal to some value P . Then $\bar{o}_{i,i+1} = \log(P/(1-P))$, and $\bar{o}_{i,i+n} = \bar{o}_{i,i+1} + \bar{o}_{i+1,i+2} + \dots + \bar{o}_{i+n-1,i+n} = n\bar{o}_{i,i+1}$ gives $P_{i,i+n} = \Delta^n/(1+\Delta^n)$, where Δ is the odds ratio $\Delta = P/(1-P)$. The expected strengthening (or weakening) of confidence in the ordering of a given pair, as their difference in ranks increases, is then captured by:

Lemma 1. *Let $n > 0$. If $P > \frac{1}{2}$, then $P_{i,i+n} \geq P$ with equality when $n = 1$, and $P_{i,i+n}$ increases strictly monotonically with n . If $P < \frac{1}{2}$, then $P_{i,i+n} \leq P$ with equality when $n = 1$, and $P_{i,i+n}$ decreases strictly monotonically with n . If $P = \frac{1}{2}$, then $P_{i,i+n} = \frac{1}{2}$ for all n .*

⁵ A similar argument can be found in [21]; however there the intent was to uncover underlying class conditional probabilities from pairwise probabilities; here, we have no analog of the class conditional probabilities.

Proof: Assume that $n > 0$. Since $P_{i,i+n} = 1/(1 + (\frac{1-P}{P})^n)$, then for $P > \frac{1}{2}$, $\frac{1-P}{P} < 1$ and the denominator decreases strictly monotonically with n ; and for $P < \frac{1}{2}$, $\frac{1-P}{P} > 1$ and the denominator increases strictly monotonically with n ; and for $P = \frac{1}{2}$, $P_{i,i+n} = \frac{1}{2}$ by substitution. Finally if $n = 1$, then $P_{i,i+n} = P$ by construction. \square

We end this section with the following observation. In [16] and [4], the authors consider models of the following form: for some fixed set of events A_1, \dots, A_k , pairwise probabilities $P(A_i|A_i \text{ or } A_j)$ are given, and it is assumed that there is a set of probabilities \hat{P}_i such that $P(A_i|A_i \text{ or } A_j) = \hat{P}_i/(\hat{P}_i + \hat{P}_j)$. This is closely related to the model described here, where for example one can model \hat{P}_i as $N \exp(o_i)$, where N is an overall normalization.

5.2 RankNet: Learning to Rank with Neural Nets

The above cost function is general, in that it is not tied to any particular learning model; here we explore using it in neural network models. Neural networks provide us with a large class of easily learned functions to choose from. Let us remind the reader of the general back-prop equations⁶ for a two layer net with q output nodes [20]. For training sample \mathbf{x} , denote the outputs of net by o_i , $i = 1, \dots, q$, the targets by t_i , $i = 1, \dots, q$, let the transfer function of each node in the j th layer of nodes be g^j , and let the cost function be $\sum_{i=1}^q C(o_i, t_i)$. If α_k are the parameters of the model, then a gradient descent step amounts to $\delta\alpha_k = -\eta_k \frac{f}{\alpha_k}$, where the η_k are positive learning rates. This network embodies the function

$$o_i = g^3 \left(\sum_j w_{ij}^{32} g^2 \left(\sum_k w_{jk}^{21} x_k + b_j^2 \right) + b_i^3 \right) \equiv g_i^3$$

where for the weights w and offsets b , the upper indices index the node layer, and the lower indices index the nodes within each corresponding layer. Taking derivatives of C with respect to the parameters gives

$$\begin{aligned} \frac{C}{b_i^3} &= \frac{C}{o_i} g_i^3 \equiv \Delta_i^3 & (3) \\ \frac{C}{w_{in}^{32}} &= \Delta_i^3 g_n^2 \\ \frac{C}{b_m^2} &= g_m^2 \left(\sum_i \Delta_i^3 w_{im}^{32} \right) \equiv \Delta_m^2 \\ \frac{C}{w_{mn}^{21}} &= x_n \Delta_m^2 \end{aligned}$$

⁶ *Back-prop* gets its name from the propagation of the Δ 's backwards through the network (cf. Eq. 3), by analogy to the 'forward prop' of the node activations.

where x_n is the n th component of the input. Thus, ‘backProp’ consists of a forward pass, during which the activations, and their derivatives, for each node are stored; Δ_1^3 is computed for the output layer, and is then used to update the bias b for the output node; the weight updates for the w^{32} are then computed by simply multiplying Δ_1^3 by the outputs of the hidden nodes; the Δ_m^2 are then computed using the activation gradients and the current weight values; and the process repeats for the layer below. This procedure generalizes in the obvious way for more general networks.

Turning now to a net with a single output, the above is generalized to the ranking problem as follows [3]. Recall that the cost function is a function of the difference of the outputs of two consecutive training samples: $C(o_2 - o_1)$. Here it is assumed that the first pattern is known to rank higher than, or equal to, the second (so that, in the first case, C is chosen to be monotonic increasing). Note that C can include parameters encoding the importance assigned to a given pair. A forward prop is performed for the first sample; each node’s activation and gradient value are stored; a forward prop is then performed for the second sample, and the activations and gradients are again stored. The gradient of the cost is then

$$\frac{C}{\alpha} = \left(\frac{o_2}{\alpha} - \frac{o_1}{\alpha} \right) C'$$

where C' is just the derivative of C with respect to $o_2 - o_1$. We use the same notation as before but add a subscript, 1 or 2, denoting which pattern is the argument of the given function, and we drop the index on the last layer. Thus we have

$$\begin{aligned} \frac{C}{b^3} &= f'(g_2^3 - g_1^3) \equiv \Delta_2^3 - \Delta_1^3 \\ \frac{C}{w_m^{32}} &= \Delta_2^3 g_{2m}^2 - \Delta_1^3 g_{1m}^2 \\ \frac{C}{b_m^2} &= \Delta_2^3 w_m^{32} g_{2m}^2 - \Delta_1^3 w_m^{32} g_{1m}^2 \\ \frac{C}{w_{mn}^{21}} &= \Delta_{2m}^2 g_{2n}^1 - \Delta_{1m}^2 g_{1n}^1 \end{aligned}$$

Note that the terms always take the form⁷ of the difference of a term depending on \mathbf{x}_1 and a term depending on \mathbf{x}_2 , ‘coupled’ by an overall multiplicative factor of C' , which depends on both. A sum over weights does not appear because we are considering a two layer net with one output, but for more layers the sum appears as above; thus training RankNet is accomplished by a straightforward modification of the back-prop algorithm.

⁷ One can also view this as a weight sharing update for a Siamese-like net[2]. However Siamese nets use a cosine similarity measure for the cost function, which results in a different form for the update equations.

6 Ranking as Learning Structured Outputs

Let’s take a step back and ask: are the above algorithms solving the right problem? They are certainly attempting to learn an ordering of the data. However, in this Section I argue that, in general, the answer is no. Let’s revisit the cost metrics described in Section 2. We assume throughout that the documents have been ordered by decreasing score.

These metrics present two key challenges. First, they all depend on not just the output s for a single feature vector F , but on the outputs of all feature vectors, for a given query; for example for WTA, we must compare all the scores to find the maximum. Second, none are differentiable functions of their arguments; in fact they are flat over large regions of parameter space, which makes the learning problem much more challenging. By contrast, note that the algorithms described above have the property that, in order to make the learning problem tractable, they use smooth costs. This smoothness requirement is, in principle, not necessarily a burden, since in the ideal case, when the algorithm can achieve zero cost on the some dataset, it has also achieved zero cost using any of the above measures. Hence, the problems that arise from using a simple, smooth approximation to one of the above cost functions, arise because in practice, learning algorithms cannot achieve perfect generalization. This itself has several root causes: the amount of available labeled data may be insufficient; the algorithms themselves have finite capacity to learn (and if the amount of training data is limited, as is often the case, this is a very desirable property [24]); and due to noise in the data and/or the labels, perfect generalization is often not even theoretically possible.

For a concrete example of where using an approximate cost can lead to problems, suppose that we use a smooth approximation to pair-wise error (such as the RankNet cost function), but that what we really want to minimize is the WTA cost. Consider a training query with 1,000 returned documents, and suppose that there are two relevant documents D_1 and D_2 , and 998 irrelevant documents, and that the ranker puts D_1 in position 1 and D_2 in position 1000. Then the ranker can reduce the pair-wise error, for that query, by 996 errors, by moving D_2 up to rank 3 and by moving D_1 down to rank 2. However the WTA error has gone from zero to one. A huge decrease in the pairwise error rate has resulted in the maximum possible increase in the WTA cost.

The need for the ability to handle multivariate costs is not limited to traditional ranking problems. For example, one measure of quality for document retrieval, or in fact of classifiers in general, is the “AUC”, the area under the ROC curve [1]. Maximizing the AUC amounts to learning using a multivariate cost and is in fact also exactly a binary ranking problem: see, for example, [8, 15]. Similarly, optimizing measures that depend on precision and recall can be viewed as optimizing a multivariate cost [19, 15].

In order to learn using a multivariate, non-differentiable cost function, we propose a general approach, which for the ranking problem we call LambdaRank.

We describe the approach in the context of learning to rank using gradient descent. Here a general multivariate cost function for a given query takes the form $C(s_{ij}, l_{ij})$, where i indexes the query and j indexes a returned document for that query. Thus, in general the cost function may take a different number of arguments, depending on the query (some queries may get more documents returned than others). In general, finding a smooth cost function that has the desired behaviour is very difficult. Take the above WTA example. It is much more important to keep D_1 in the top position than to move D_2 up 997 positions and D_1 down one: the optimal WTA cost is achieved when either D_1 or D_2 is in the top position. Notice how the finite capacity of the learning algorithm is playing a crucial role here. In this particular case, to better approximate WTA, one approach would be to steeply discount errors that occur low in the ranking. Now imagine that C is a smooth approximation to the desired cost function that accomplishes this, and assume that at the current learning iteration, \mathcal{A} produces an ordering for a given Q where D_1 is in position 2 and D_2 is in position 1000. Then if $s_i \equiv \mathcal{A}(\mathbf{x}_i)$, $i = 1, 2$, we require that

$$\left| \frac{\partial C}{\partial s_1} \right| \gg \left| \frac{\partial C}{\partial s_2} \right|$$

Notice that we've captured a desired property of C by imposing a constraint on its derivatives. The idea of LambdaRank is to extend this by replacing the requirement of specifying C itself, by the task of specifying its derivative with respect to each s_j , $j = 1, \dots, n_i$, for each query Q_i . Those derivatives can then be used to train \mathcal{A} using gradient descent, just as the derivatives of C normally would be. The point is that it can be much easier, given an instance of a query and its ranked documents, to specify how you would like those documents to move, in order to reduce a non-differentiable cost, than to specify a smooth approximation of that (multivariate) cost. As a simple example, consider a single query with just two returned documents D_1 and D_2 , and suppose they have labels $l_1 = 1$ (relevant) and $l_2 = 0$ (not relevant), respectively. We imagine that there is some $C(s_1, l_1, s_2, l_2)$ such that

$$\begin{aligned} \frac{\partial C}{\partial s_1} &= -\lambda_1(s_1, l_1, s_2, l_2) \\ \frac{\partial C}{\partial s_2} &= -\lambda_2(s_1, l_1, s_2, l_2) \end{aligned}$$

We would like the λ 's to take the form shown in Figure 2, for some chosen margin $\delta \in \mathbb{R}$: thinking of the documents as lying on a vertical line, where higher scores s correspond to higher points on the line, then D_1 (D_2) gets a constant gradient up (or down) as long as it is in the incorrect position, and the gradient goes smoothly to zero until the margin is achieved. Thus the

learning algorithm \mathcal{A} will not waste capacity moving D_1 further away from D_2 if they are in the correct position by more than δ , and having nonzero δ ensures robustness to small changes in the scores s_i .

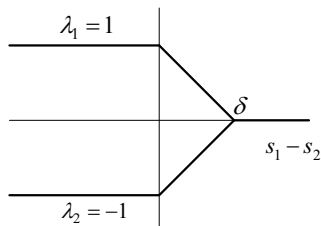


Fig. 2. Choosing the lambda's for a query with two documents.

Letting $x \equiv s_1 - s_2$, the λ 's may be written

$$\begin{aligned} x < 0 : \lambda_1 = 1 = -\lambda_2 \\ 0 \leq x \leq \delta : \lambda_1 = \delta - x = -\lambda_2 \\ x > \delta : \lambda_1 = \lambda_2 = 0 \end{aligned}$$

In this case a corresponding cost function exists:

$$\begin{aligned} x < 0 : C(s_1, l_1, s_2, l_2) &= s_2 - s_1 \\ 0 \leq x \leq \delta : C(s_1, l_1, s_2, l_2) &= \frac{1}{2}(s_1 - s_2)^2 - \delta(s_1 - s_2) \\ x > \delta : C(s_1, l_1, s_2, l_2) &= -\frac{1}{2}\delta^2 \end{aligned}$$

Note that in addition the Hessian of C is positive semidefinite, so the cost function takes a unique minimum value (although the s 's for which C attains its minimum are not unique). In general, when the number of documents for a given query is much larger than two, and where the rules for writing down the λ 's depend on the scores, labels and ranks of all the documents, then C can become prohibitively complicated to write down explicitly.

There is still a great deal of freedom in this model, namely, how to choose the λ 's to best model a given (multivariate, non-differentiable) cost function. Let's call this choice the λ -function. We will not explore here how, given a cost function, to find a particular λ -function, but instead will answer two questions which will help guide the choice: first, for a given choice of the λ 's, under what conditions does there exist a cost function C for which they are the negative derivatives? Second, given that such a C exists, under what conditions is C convex? The latter is desirable to avoid the problem that local minima in the cost function itself will present to any algorithm used for training \mathcal{A} . To

address the first question, we can use a well-known result from multilinear algebra [23]:

Theorem 2. (*Poincaré Lemma*): *If $S \subset \mathbb{R}^n$ is an open set that is star-shaped with respect to 0, then every closed form on S is exact.*

Note that since every exact form is closed, it follows that on an open set that is star-shaped with respect to 0, a form is closed if and only if it is exact. Now for a given query Q_i and corresponding set of returned D_{ij} , the n_i λ 's are functions of the scores s_{ij} , parameterized by the (fixed) labels l_{ij} . Let dx^i be a basis of 1-forms on \mathbb{R}^n and define the 1-form

$$\boldsymbol{\lambda} \equiv \sum_i \lambda_i dx^i$$

Then assuming that the scores are defined over \mathbb{R}^n , the conditions for Theorem 2 are satisfied and $\boldsymbol{\lambda} = dC$ for some function C if and only if $d\boldsymbol{\lambda} = 0$ everywhere. Using classical notation, this amounts to requiring that

$$\frac{\partial \lambda_i}{\partial s_j} = \frac{\partial \lambda_j}{\partial s_i} \quad \forall i, j \quad (4)$$

Thus we have a simple test on the λ 's to determine if there exists a cost function for which they are the derivatives: the Jacobian (that is, the matrix $J_{ij} \equiv \partial \lambda_i / \partial s_j$) must be symmetric. Furthermore, given that such a cost function C does exist, the condition that it be convex is that the Jacobian be positive semidefinite everywhere. Under these constraints, the Jacobian is beginning to look very much like a kernel matrix! However, there is a difference: the value of the i 'th, j 'th element of a kernel matrix depends on two vectors $\mathbf{x}_i, \mathbf{x}_j$ (where for example $\mathbf{x} \in \mathbb{R}^d$ for some d , although in general they may be elements of an abstract vector space), whereas the value of the i 'th, j 'th element of the Jacobian depends on all of the scores s_i .

For choices of the λ 's that are piecewise constant, the above two conditions (symmetric and positive semidefinite⁸) are trivially satisfied. For other choices of symmetric J , positive definiteness can be imposed by adding regularization terms of the form $\lambda_i \mapsto \lambda_i + \alpha s_i$, $\alpha_i > 0$, which amounts to adding a positive constant along the diagonal of the Hessian.

Finally, we observe that LambdaRank has a clear physical analogy. Think of the documents returned for a given query as point masses. Each λ then corresponds to a force on the corresponding point. If the conditions of Eq. (4) are met, then the forces in the model are conservative, that is, the forces may be viewed as arising from a potential energy function, which in our case is the cost function. For example, if the λ 's are linear in the outputs s , then

⁸ Some authors define the property of positive semi-definiteness to include the property of symmetry: see [14].

this corresponds to a spring model, with springs that are either compressed or extended. The requirement that the Jacobian is positive semidefinite amounts to the requirement that the system of springs have a unique global minimum of the potential energy, which can be found from any initial conditions by gradient descent (this is not true in general, for arbitrary systems of springs).

Acknowledgement

I thank J. Levesley and J. Platt for useful discussions.

References

1. D. Bamber: The area above the ordinal dominance graph and the area below the receiver operating characteristic graph. *Journal of Mathematical Psychology* **12**, 1975, 387–415.
2. J. Bromley, J.W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Sackinger, and R. Shah: Signature verification using a “siamese” time delay neural network. In: *Advances in Pattern Recognition Systems using Neural Network Technologies*, Machine Perception Artificial Intelligence 7, I. Guyon and P.S.P. Wang (eds.), World Scientific, 1993, 25–44.
3. C.J.C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender: Learning to rank using gradient descent. In: *Proceedings of the Twenty Second International Conference on Machine Learning*, Bonn, Germany, 2005.
4. R. Bradley and M. Terry: The rank analysis of incomplete block designs 1: the method of paired comparisons. *Biometrika* **39**, 1952, 324–245.
5. C.J.C. Burges: A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* **2**(2), 1998, 121–167.
6. E.B. Baum and F. Wilczek: Supervised learning of probability distributions by neural networks. In: *Neural Information Processing Systems*, D. Anderson (ed.), American Institute of Physics, 1988, 52–61.
7. W. Chu and S.S. Keerthi: New approaches to support vector ordinal regression. In: *Proceedings of the Twenty Second International Conference on Machine Learning*, Bonn, Germany, 2005.
8. C. Cortes and M. Mohri: Confidence intervals for the area under the ROC curve. In *Advances in Neural Information Processing Systems 18*. MIT Press, 2005.
9. K. Crammer and Y. Singer: Pranking with ranking. In: *Advances in Neural Information Processing Systems 14*. MIT Press, 2002.
10. C. Cortes and V. Vapnik: Support vector networks. *Machine Learning* **20**, 1995, 273–297.
11. O. Dekel, C.D. Manning, and Y. Singer: Log-linear models for label-ranking. In: *Advances in Neural Information Processing Systems 16*. MIT Press, 2004.
12. E.F. Harrington: Online ranking/collaborative filtering using the perceptron algorithm. In: *Proceedings of the Twentieth International Conference on Machine Learning*, 2003.

13. R. Herbrich, T. Graepel, and K. Obermayer: Large margin rank boundaries for ordinal regression. In: *Advances in Large Margin Classifiers*, A.J. Smola, P.L. Bartlett, B. Schölkopf, and D. Schuurmans (eds.), MIT Press, 2000, 115–132.
14. R.A. Horn and C.R. Johnson: *Matrix Analysis*. Cambridge University Press, 1985.
15. A. Herschtal and B. Raskutti: Optimising area under the ROC curve using gradient descent. In: *Proceedings of the Twenty First International Conference on Machine Learning*, Banff, Canada, 2004.
16. T. Hastie and R. Tibshirani: Classification by pairwise coupling. In: *Advances in Neural Information Processing Systems*, vol. 10, M.I. Jordan, M.J. Kearns, and S.A. Solla (eds.), MIT Press, 1998.
17. K. Jarvelin and J. Kekalainen: IR evaluation methods for retrieving highly relevant documents. In: *Proceedings of the 23rd annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, New York, 2000, 41–48.
18. T. Joachims: Optimizing search engines using clickthrough data. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-02)*, D. Hand, D. Keim, and R. Ng (eds.), ACM Press, New York, 2002, 132–142.
19. T. Joachims: A support vector method for multivariate performance measures. In: *Proceedings of the 22nd International Conference on Machine Learning*, L. De Raedt and S. Wrobel (eds.), 2005, 377–384.
20. Y. LeCun, L. Bottou, G.B. Orr, and K.-R. Müller: Efficient backprop. In: *Neural Networks: Tricks of the Trade*, G.B. Orr and K.-L. Müller (eds.), Springer, 1998, 9–50.
21. P. Refregier and F. Vallet: Probabilistic approaches for multiclass classification with neural networks. In: *International Conference on Artificial Neural Networks*, Elsevier, 1991, 1003–1006.
22. B. Schölkopf and A. Smola: *Learning with Kernels*. MIT Press, 2002.
23. M. Spivak: *Calculus on Manifolds*. Addison-Wesley, 1965.
24. V. Vapnik: *The Nature of Statistical Learning Theory*. Springer, New York, 1995.
25. E.M. Voorhees: Overview of the TREC 2001 question answering track. In: *TREC*, 2001.
26. E.M. Voorhees: Overview of the TREC 2002 question answering track. In *TREC*, 2002.

Two Algorithms for Approximation in Highly Complicated Planar Domains

Nira Dyn and Roman Kazinnik

School of Mathematical Sciences, Tel-Aviv University, Tel-Aviv 69978, Israel,
{niradyn,romank}@post.tau.ac.il

Summary. Motivated by an adaptive method for image approximation, which identifies “smoothness domains” of the image and approximates it there, we developed two algorithms for the approximation, with small encoding budget, of smooth bivariate functions in highly complicated planar domains. The main application of these algorithms is in image compression. The first algorithm partitions a complicated planar domain into simpler subdomains in a recursive binary way. The function is approximated in each subdomain by a low-degree polynomial. The partition is based on both the geometry of the subdomains and the quality of the approximation there. The second algorithm maps continuously a complicated planar domain into a k -dimensional domain, where approximation by one k -variate, low-degree polynomial is good enough. The integer k is determined by the geometry of the domain. Both algorithms are based on a proposed measure of domain singularity, and are aimed at decreasing it.

1 Introduction

In the process of developing an adaptive method for image approximation, which determines “smoothness domains” of the image and approximates it there [5, 6], we were confronted by the problem of approximating a smooth function in highly complicated planar domains. Since the adaptive approximation method is aimed at image compression, an important property required from the approximation in the complicated domains is a low encoding budget, namely that the approximation is determined by a small number of parameters. We present here two algorithms. The first algorithm approximates the function by piecewise polynomials. The algorithm generates a partition of the complicated domain to a small number of less complicated subdomains, where low-degree polynomial approximation is good enough. The partition is a binary space partition (BSP), driven by the geometry of the domain and is encoded with a small budget. This algorithm is used in the compression method of [5, 6]. The second algorithm is based on mapping a complicated

domain continuously into a k -dimensional domain in which one k -variate low-degree polynomial provides a good enough approximation to the mapped function. The integer k depends on the geometry of the complicated domain. The approximant generated by the second algorithm is continuous, but is not a polynomial. The suggested mapping can be encoded with a small budget, and therefore also the approximant.

Both algorithms are based on a new measure of domain singularity, concluded from an example, showing that in complicated domains the smoothness of the function is not equivalent to the approximation error, as is the case in convex domains [4], and that the quality of the approximation depends also on geometric properties of the domain. The outline of the paper is as follows: In Section 2, first we discuss some of the most relevant theoretical results on polynomial approximation in planar domains. Secondly, we introduce our example violating the Jackson-Bernstein inequality, which sheds light on the nature of domain singularities for approximation.

Subsequently in Section 3 we propose a measure for domain singularity. The first algorithm is presented and discussed in Section 4, and the second in Section 5.

Several numerical examples, demonstrating various issues discussed in the paper, are presented. In the examples, the approximated bivariate functions are images, defined on a set of pixels, and the approximation error is measured by PSNR, which is proportional to the logarithm of the inverse of the discrete L_2 -error.

2 Some Facts about Polynomial Approximation in Planar Domains

This section reviews relevant results on L_2 bivariate polynomial approximation in planar domains. By analyzing an example of a family of polynomial approximation problems, we arrive at an understanding of the nature of domain singularities for approximation by polynomials. This understanding is the basis for the measure of domain singularity proposed in the next section, and used later in the two algorithms.

2.1 L_2 -Error

The error of L_2 bivariate polynomial approximation in convex and ‘almost-convex’ planar domains $\Omega \subset \mathbb{R}^2$ can be characterized by the smoothness of the function in the domain (see [3, 4]). These results can be formulated in terms of the moduli of continuity/smoothness of the approximated function, or of its weak derivatives. Here we cite results on general domains.

Let $\Omega \subset \mathbb{R}^2$ be a bounded domain and let $f \in L_2(\Omega)$. For $m \in \mathbb{N}$, the m -th difference operator is:

$$\Delta_h^m(f, \Omega)(x) = \begin{cases} \sum_{k=0}^m (-1)^{m+k} \binom{m}{k} f(x+kh), & \text{for } [x, x+mh] \subset \Omega, \\ 0, & \text{otherwise,} \end{cases}$$

where $h \in \mathbb{R}^2$, and $[x, y]$ denotes the line segment connecting the two points $x, y \in \mathbb{R}^2$. The m -th order $L_2(\Omega)$ modulus of smoothness is defined for $t > 0$ as

$$\omega_m(f, t, \Omega)_2 = \sup_{|h| < t} \|\Delta_h^m(f, \Omega)\|_{L_2(\Omega)},$$

with $|h|$ the Euclidean norm of $h \in \mathbb{R}^2$.

Denote by Π_n the linear space of bivariate polynomials of total degree $n-1$, then the L_2 approximation error on Ω , is defined as

$$E_n(f, \Omega)_2 = \inf_{p \in \Pi_n} \|f - p\|_{L_2(\Omega)}.$$

This quantity is equivalent in Lipschitz domains to the modulus of smoothness of f , namely there exist $C_1, C_2 > 0$ such that

$$C_1 \omega_n(f, \text{diam}(\Omega), \Omega)_2 \leq E_n(f, \Omega)_2 \leq C_2 \omega_n(f, \text{diam}(\Omega), \Omega)_2 \quad (1)$$

(see [4] for further details). While the constant C_1 depends only on n , the constant C_2 depends on both n and the geometry of Ω . For example, in the case of a *star-shaped domain* the constant C_2 depends on the *chunkiness parameter* $\gamma = \inf_{B \subset \Omega} \frac{\text{diam}(\Omega)}{\text{radius}(B)}$, with B a disc ([1]). In particular, the Bramble-Hilbert lemma states that for $f \in W_2^m(\Omega)$, $m \in \mathbb{N}$, where $W_2^m(\Omega)$ is the Sobolev space of functions with all weak derivatives of order m in $L_2(\Omega)$, there exists a polynomial $p_n \in \Pi_n$ for which

$$|f - p_n|_{k,2} \leq C(n, m, \gamma) \text{diam}(\Omega)^{m-k} |f|_{m,2},$$

where $k = 0, 1, \dots, m$ and $|\cdot|_{m,2}$ denotes the Sobolev semi-norm. It is important to note that in [4] the dependence on the geometry of Ω in case of convex domains is eliminated.

When the geometry of the domain is complicated then the smoothness of the function inside the domain does not guarantee the quality of the approximation. Figure 1 shows an example of a smooth function, which is poorly approximated in a highly non-convex domain.

2.2 An Instructive Example

Here we show that (1) cannot hold with a constant C_2 independent of the domain, by an example that “blows-up” the constant C_2 in (1). For this example we construct a smooth function f and a family of planar domains $\{\Omega_\epsilon\}$, such that for any positive t and n , $\omega_n(f, t, \Omega_\epsilon)_2 \rightarrow 0$ as $\epsilon \rightarrow 0$, while $E_n(f, \Omega_\epsilon)_2 = O(1)$.

Let S denote the open the square with vertices $(\pm 1, \pm 1)$, and let R_ϵ denote the closed rectangle with vertices $(\pm(1 - \epsilon), \pm\frac{1}{2})$. The domains of approximation are $\{\Omega_\epsilon = S \setminus R_\epsilon\}$. The function f is smooth in S , and satisfies

$$f(x) = \begin{cases} 1, & \text{for } x \in S \cap \{x : x_2 > \frac{1}{2}\}, \\ 0, & \text{for } x \in S \cap \{x : x_2 < -\frac{1}{2}\}, \end{cases}$$

where $x = (x_1, x_2)$.

It is easy to verify that $\omega_n(f, t, \Omega_\epsilon)_2 \rightarrow 0$ as $\epsilon \rightarrow 0$. We claim that $E_n(f, \Omega_\epsilon)_2$ for small ϵ is bounded below by a positive constant. To prove the claim assume that it is false. Then there exists a sequence $\{\epsilon_k\}$, tending to zero, such that $E_n(f, \Omega_{\epsilon_k})_2 \rightarrow 0$. Denote by $p_k \in \Pi_n$ the polynomial satisfying $E_n(f, \Omega_{\epsilon_k}) = \|f - p_k\|_{L_2(\Omega_{\epsilon_k})}$. Since there is a convergent subsequence of $\{p_k\}$, with a limit denoted by p^* , then $\|f - p^*\|_{L_2(\Omega_0)} = 0$, which is impossible.

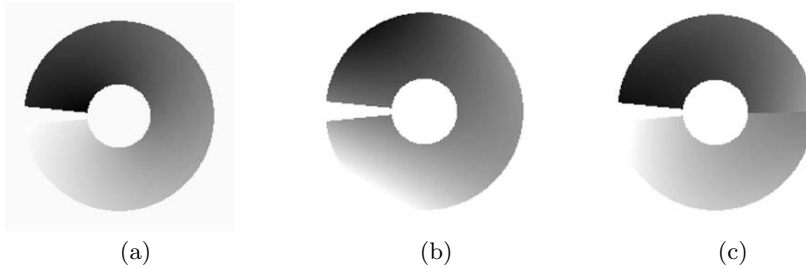


Fig. 1. (a) given smooth function, (b) “poor” approximation with a quadratic polynomial over the entire domain (PSNR=21.5 dB), (c) approximation improves once the domain is partitioned into “simpler” subdomains (PSNR=33 dB).

The relevant conclusion from this example is that the quality of bivariate polynomial approximation depends both on the smoothness of the approximated function and on the geometry of the domain. Yet, in convex domains the constant C_2 in (1) is geometry independent [4].

Defining the *distance defect ratio* of a pair of points $x, y \in cl(\Omega) = \Omega \cup \partial\Omega$ (with $\partial\Omega$ the boundary of Ω) by

$$\mu(x, y)_\Omega = \frac{\rho(x, y)_\Omega}{|x - y|} \quad (2)$$

where $\rho(x, y)_\Omega$ is the length of the shortest path inside $cl(\Omega)$ connecting x and y , we observe that in the domains $\{\Omega_\epsilon\}$ of the example, there exist pairs of points with distance defect ratio growing as $\epsilon \rightarrow 0$.

Note that there is no upper bound for the distance defect ratio of arbitrary domains, while in convex domains the distance defect ratio is 1.

For a domain Ω with $x, y \in cl(\Omega)$, such that $\mu(x, y)_\Omega$ is large, and for a smooth f in Ω , with $|f(x) - f(y)|$ large, the approximation by a polynomial is poor (see e.g. Figure 1). This is due to the fact that a polynomial cannot change significantly between the close points x, y , if it changes moderately in Ω (as an approximation to a smooth function in Ω).



Fig. 2. (a) cameraman image, (b) example of segmentation curves, (c) complicated domains generated by the segmentation in (b).

3 Distance Defect Ratio as a Measure for Domain Singularity

It is demonstrated in Section 2.2 that the ratio between the L_2 -error of bivariate polynomial approximation and the modulus of smoothness of the approximated function, can be large due to the geometry of the domain. In a complicated domain the quality of the approximation might be very poor, even for very smooth functions inside the domain, as is illustrated by Figure 1.

Since in convex domains this ratio is bounded independently of the geometry of the domains, a potential solution would be to *triangulate* a complicated domain, and to approximate the function separately in each triangle. However the triangulation is not optimal in the sense that it may produce an excessively large amount of triangles. In practice, since reasonable approximation can often be achieved in mildly nonconvex domains, one need not force partitioning into convex regions, but try to reduce the singularities of a domain.

Here we propose a measure of the *singularity* of a domain, assuming that convex domains have no singularity. Later, we present two algorithms which aim at reducing the singularities of the domain where the function is approximated; one by partitioning it into subdomains with smaller singularities, and the other by mapping it into a less singular domain in higher dimension.

The measure of domain singularity we propose, is defined for a domain Ω , such that $\rho(x, y)_\Omega < \infty$, for any $x, y \in \partial\Omega$. Denote the convex hull of Ω by

H , and the complement of Ω in H by

$$\mathcal{C} = H \setminus \Omega .$$

The set \mathcal{C} may consist of a number of disjoint components $\mathcal{C} = \bigcup \mathcal{C}_i$.

A complicated planar domain $\tilde{\Omega}$, the corresponding sets H and \mathcal{C} , the latter consisting of several disjoint components $\{\mathcal{C}_i\}$, are shown in Figure 4. Note that each \mathcal{C}_i can potentially impede the polynomial approximation, independently of the other components, as is indicated by the example in Section 2.2.



Fig. 3. (a) example of a subdomain in the cameraman initial segmentation, (b) example of one geometry-driven partition with a straight line.

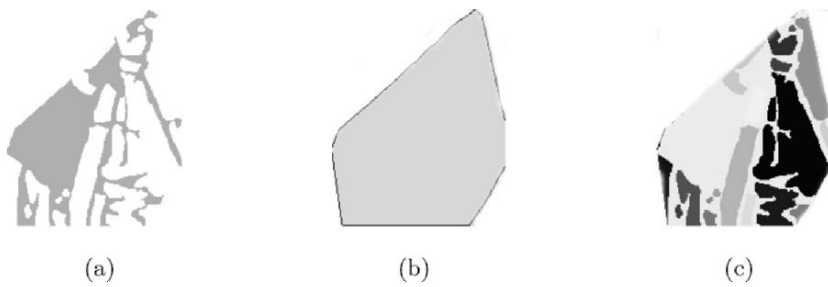


Fig. 4. (a) a subdomain $\tilde{\Omega}$ generated by the partition in Figure 3, (b) its convex hull H , (c) the corresponding disjoint components $\{\mathcal{C}_i\}$ of $H \setminus \tilde{\Omega}$.

For a component \mathcal{C}_i we define its corresponding *measure of geometric singularity* relative to Ω by

$$\mu(\mathcal{C}_i)_\Omega = \max_{x,y \in \partial\mathcal{C}_i \cap \partial\Omega} \mu(x,y)_\Omega, \quad (3)$$

with $\mu(x,y)_\Omega$ the distance defect ratio defined in (2). We denote by $\{P_1^i, P_2^i\}$ a pair of points at which the maximum in (3) is attained. The *measure of geometric singularity* of the domain Ω we propose is

$$\mu(\Omega) = \max_i \mu(\mathcal{C}_i)_\Omega.$$

Since every component \mathcal{C}_i introduces a *singularity* of the domain Ω , we refer to the i -th (*geometric*) *singularity component* of the domain Ω as the triplet: the component \mathcal{C}_i , the distance defect ratio $\mu(\mathcal{C}_i)_\Omega$, and the pair of points $\{P_1^i, P_2^i\}$.

4 Algorithm 1: Geometry-Driven Binary Partition

We presently describe the *geometry-driven binary partition* algorithm for approximating a function in complicated domains. We demonstrate the application of the algorithm on a planar domain from the segmentation of the cameraman image, as shown in Figure 2(c), and on a domain with one domain singularity, as shown in Figure 8(a), and Figure 8(b).

Our algorithm employs the measure of domain singularity introduced in Section 3, and produces geometry-driven partition of a complicated domain, which targets at efficient piecewise polynomial approximation with low-budget encoding cost. The algorithm constructs recursively a binary space partition (BSP) tree, improving gradually the corresponding piecewise polynomial approximation and discarding the domain singularities. The decisions taken during the performance of the algorithm are based on both the quality of the approximation and the measure of geometric singularity.

4.1 Description of the Algorithm

The algorithm constructs the binary tree recursively. The root of the tree is the initial domain Ω , and its nodes are subdomains of Ω . The leaves of the tree are subdomains where the polynomial approximation is good enough. For a subdomain $\tilde{\Omega} \subset \Omega$ at a node of the binary tree, first a least-squares polynomial approximation to the given function is constructed. If the approximation error is below the prescribed allowed error, then the node becomes a leaf. If not, then the domain $\tilde{\Omega}$ is partitioned.

The partitioning step: the algorithm constructs the components $\{\tilde{\mathcal{C}}_i\}$ of the complement of $\tilde{\Omega}$ in its convex hull, and selects $\tilde{\mathcal{C}}_i$ with the largest $\mu(\tilde{\mathcal{C}}_i)_{\tilde{\Omega}}$. Then the algorithm partitions $\tilde{\Omega}$ with a *ray*, which is a straight line perpendicular to $\partial\tilde{\mathcal{C}}_i$, cast from the point $P \in \partial\tilde{\mathcal{C}}_i \cap \partial\tilde{\Omega}$, chosen such that $\rho(P, P_1^i) = \rho(P, P_2^i)$, where $\{P_1^i, P_2^i\}$ are the pair of points of the singularity

component $\tilde{\mathcal{C}}_i$, as defined in Section 3. We favor the partition along a straight line since a straight line does not create new non-convexities and is coded with a small budget. By this partition we discard the worst singularity component (the one with the largest distance defect ratio).

It may happen that $\tilde{\mathcal{C}}_i$ lies entirely “inside” $\tilde{\Omega}$. Then two rays in two directions are needed in order to partition $\tilde{\Omega}$ in a way that eliminates the singularity of $\tilde{\mathcal{C}}_i$. These two rays are perpendicular to $\partial\tilde{\mathcal{C}}_i \cap \partial\tilde{\Omega}$ at the two points P_1^i, P_2^i .

In Figure 5 partition by ray casting is demonstrated schematically, for the case of a singularity component “outside” the domain with one ray, and for the case of a singularity domain “inside” the domain with two rays.

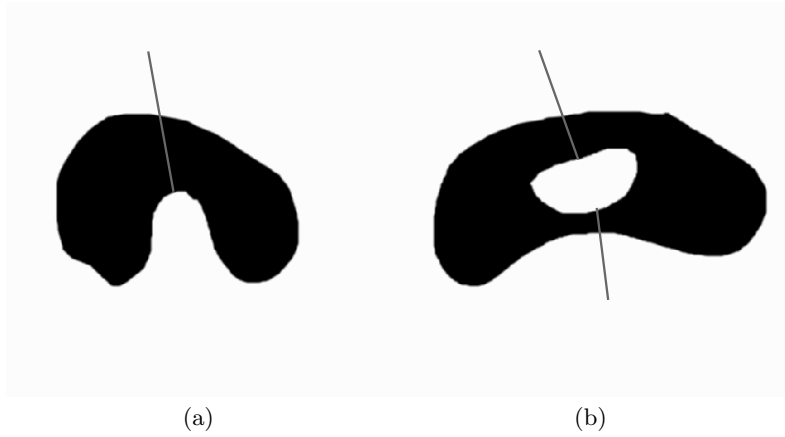


Fig. 5. Partition of a domain by ray casting. (a) by one ray for a singularity component “outside” the domain, (b) by two rays for a singularity component “inside” the domain.

For the construction of the convex hull H and the components $\{\mathcal{C}_i\}$ of a domain, we employ the *sweep* algorithm of [2] (see [5]), which is a scan based algorithm for finding connected components in a domain defined by a discrete set of pixels.

4.2 Two Examples

In this section we demonstrate the performance of the algorithm on two examples. We show the first steps in the performance of the algorithm on the domain Ω in Figure 3 (a). Figure 3 (b) illustrates the first partition of the domain, generating two subdomains. Next we consider the subdomain $\tilde{\Omega}$ shown in Figure 4 (a), its convex hull H , shown in Figure 4 (b), and the components $\{\mathcal{C}_i\}$ of $H \setminus \tilde{\Omega}$, shown in Figure 4 (c). The algorithm further partitions

$\tilde{\Omega}$, in order to reduce its measure of singularity and to improve the piecewise polynomial approximation.

The second example demonstrates in Figure 8(a), 8(b) a partition of a domain with one singularity, and the corresponding piecewise polynomial approximation.

4.3 A Modification of the Partitioning Step

Here is a small modification of the partitioning step of our algorithm that we find to be rather efficient. We select a small number $((2^k - 1)$ with $1 < k \leq 3$) of components $\{\mathcal{C}_i\}$, having the largest $\{\mu(\mathcal{C}_i)\}$, prompt the partitioning procedure for each of the selected components, and compute the resulting piecewise polynomial approximation. For the actual partitioning step, we select the component corresponding to the maximal reduction in the error of approximation. Thus, the algorithm performs dyadic partitions, based both on the measure of geometric singularity and on the quality of the approximation. This modification is encoded with k extra bits.

5 Algorithm 2: Dimension-Elevation

We now introduce a novel approach to 2-D approximation in complicated domains, which is *not* based on partitioning the domain. This algorithm challenges the problem of finding continuous approximants which can be encoded with a small budget.

5.1 The Basic Idea

We explain the main idea on a domain Ω with one singularity component \mathcal{C} , and later extend it straightforwardly to the case of multiple singularity components.

Roughly speaking, we suggest to raise up one point from the pair of points $\{P_1, P_2\}$ of the singularity component \mathcal{C} , along the additional dimension axis, to increase its Euclidean distance between P_1 and P_2 . This is demonstrated in Figure 6.

Once the domain Ω is *continuously* mapped to a 3-D domain $\tilde{\Omega} = \Phi(\Omega)$, and the domain singularity is resolved, the given function f is mapped to the tri-variate function $f(\Phi^{-1}(\cdot))$ defined on $\tilde{\Omega}$, which is approximated by a tri-variate polynomial p , minimizing the $L_2(\tilde{\Omega})$ -norm of the approximation error. The polynomial p , is computed in terms of orthonormal tri-variate polynomials relative to $\tilde{\Omega}$. The approximant of f in Ω is $P \circ \Phi$.

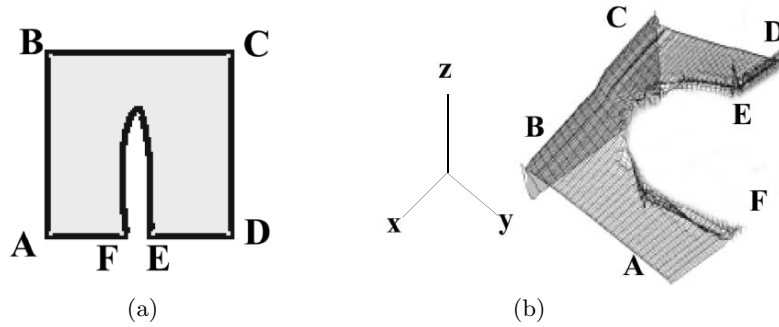


Fig. 6. (a) domain with one singularity component, (b) the domain in 3-D resulting from the continuous mapping of the planar domain.

5.2 The Dimension-Elevation Mapping

For a planar domain Ω with one singularity component, the algorithm employs a continuous one-to-one mapping $\Phi : \Omega \rightarrow \tilde{\Omega}$, $\Omega \subset \mathbb{R}^2$, $\tilde{\Omega} \subset \mathbb{R}^3$, such that for any two points in $\Phi(\Omega)$ the distance inside the domain is of the same magnitude as the Euclidean distance.

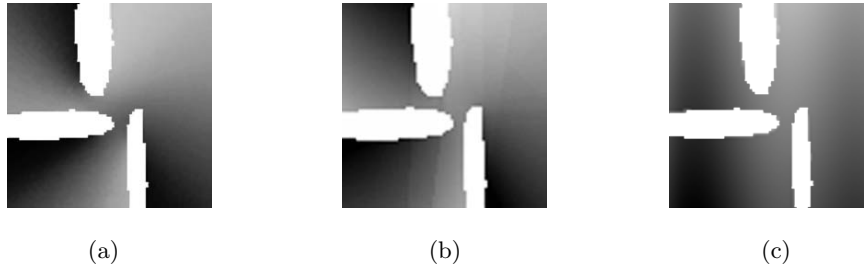


Fig. 7. (a) the original image, defined over a domain with three singularity components, (b) approximation with one 5-variate linear polynomial using a continuous 5-D mapping achieves PSNR=28.6 dB, (c) approximation using one bivariate linear polynomial produces PSNR=16.9 dB.

The continuous mapping we use is so designed to eliminate the singularity of the pair $\{P_1, P_2\}$, corresponding to the unique singularity component $\mathcal{C} = H \setminus \Omega$. The mapping $\Phi(P)$, for $P = (P_x, P_y) \in \Omega$ is

$$\Phi(P) = (P_x, P_y, h(P)) ,$$

with $h(P) = \rho(P, P_C)_\Omega$, where P_C is one of the pair of points $\{P_1, P_2\}$. Note that the mapping is continuous and one-to-one.

An algorithm for the computation of $h(P)$ is presented in [5]. This algorithm is based on the idea of *view frustum* [2], which is used in 3D graphics for culling away 3D objects. In [5], it is employed to determine a finite sequence of “source points” $\{Q_i\}$ starting from P_C , and a corresponding partition of Ω , $\{\Omega_i\}$. Each source point is the farthest visible point on $\partial\Omega$ from its predecessor in the sequence. The sequence of source points determines a partition of Ω into subdomains, such that each subdomain Ω_i is the maximal region in $\Omega \setminus \cup_{j=1}^{i-1} \Omega_j$ which is visible from Q_i . Then for $P \in \Omega_i$ we have $h(P) = |P - P_i| + \sum_{j=1}^{i-1} |P_{j+1} - P_j|$.

For a domain with multiple singularity components, we employ N additional dimensions to discard the N singularity components $\{\mathcal{C}_i, i = 1, \dots, N\}$. For each singularity component \mathcal{C}_i , we construct a mapping

$$\Phi_i(P) = (P_x, P_y, h_i(P)), \quad i = 1, \dots, N,$$

where in the definition of Φ_i we ignore the other components $\mathcal{C}_j, j \neq i$, and regard \mathcal{C}_i as a unique singularity component. The resulting mapping $\Phi : \Omega \rightarrow \tilde{\Omega}$, $\Omega \subset \mathbb{R}^2, \tilde{\Omega} \subset \mathbb{R}^{2+N}$, is defined as

$$\Phi(P) = \{P_x, P_y, h_1(P), \dots, h_N(P)\},$$

and is one-to-one and continuous.

After the construction of the mapping Φ , we compute the best $(N + 2)$ -variate polynomial approximation to $f \circ \Phi^{-1}$, in the $L_2(\Phi(\Omega))$ -norm. In case of a linear polynomial approximation, the approximating polynomial has N more coefficients than a linear bivariate polynomial. For coding purposes only these coefficients have to be encoded, since the mapping Φ is determined by the geometry of Ω , which is known to the decoder. Note that by this construction the approximant is continuous, but is not a polynomial.

5.3 Two Examples

In Figure 7 we demonstrate the operation of our algorithm in case of three domain singularities. This example indicates that the approximant generated by the dimension-elevation algorithm is superior to the bivariate polynomial approximation, in particular along the boundaries of the domain singularities.

Figure 8 displays an example, showing that the approximant generated by the dimension-elevation algorithm is better than the approximant generated by the geometry-driven binary partition algorithm, and that it has a better visual quality (by avoiding the introduction of the artificial discontinuities along the partition lines).

Acknowledgement

The authors wish to thank Shai Dekel for his help, in particular with Section 2.

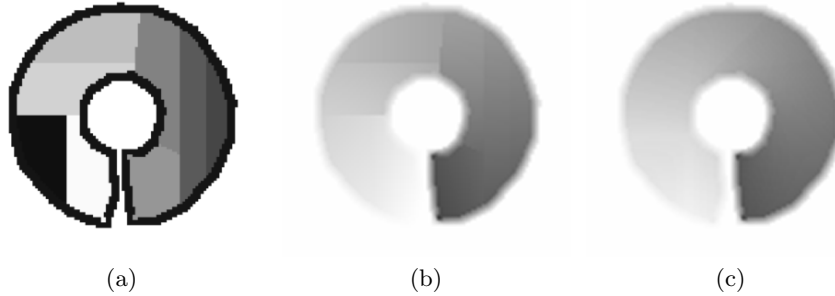


Fig. 8. Comparison of the two algorithms, approximating the smooth function $f(r, \theta) = r \cdot \theta$ in a domain with one singularity component. (a) eight subdomains are required to approximate by piecewise linear (bivariate) polynomials, (b) the piecewise linear approximant on the eight subdomains approximates with PSNR of 25.6 dB, (c) similar approximation error (25.5 dB) is achieved with one tri-variate linear polynomial using our mapping.

References

1. J.H. Bramble and S.R. Hilbert: Estimation of linear functionals on Sobolev spaces with applications to Fourier transforms and spline interpolation. *SIAM J. Numerical Analysis* **7**, 1970, 113–124.
2. M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf: *Computational Geometry Algorithms and Applications*. Springer, 1997.
3. S. Dekel and D. Leviatan: The Bramble-Hilbert lemma for convex domains. *SIAM J. Math. Anal.* **35**, 2004, 1203–1212.
4. S. Dekel and D. Leviatan: Whitney estimates for convex domains with applications to multivariate piecewise polynomial approximation. *Found. Comput. Math.* **4**, 2004, 345–368.
5. R. Kazinnik: *Image Compression using Geometric Piecewise Polynomials*. Ph.D. thesis, School of Mathematics, Tel Aviv University, in preparation.
6. R. Kazinnik, S. Dekel, and N. Dyn: Low-bit rate image coding using adaptive geometric piecewise polynomial approximation. Preprint, 2006.

Computational Intelligence in Clustering Algorithms, With Applications

Rui Xu and Donald Wunsch II

Applied Computational Intelligence Laboratory, University of Missouri, Rolla, MO 65409-0249, U.S.A., {rxu,dwunsch}@umr.edu

Summary. Cluster analysis plays an important role for understanding various phenomena and exploring the nature of obtained data. A remarkable diversity of ideas, in a wide range of disciplines, has been applied to clustering research. Here, we survey clustering algorithms in computational intelligence, particularly based on neural networks and kernel-based learning. We further illustrate their applications in five real world problems. Substantial portions of this work were first published in [87].

1 Introduction

Clustering, in contrast to supervised classification, involves problems where no labeled data are available [18, 22, 28, 45]. The goal is to separate a finite unlabeled data set into a finite and discrete set of “natural”, hidden data structures, rather than provide an accurate characterization of unobserved samples generated from the same probability distribution [4, 18]. One of the important properties of clustering is the subjectivity, which precludes an absolute judgment as to the relative efficacy of all clustering algorithms [4, 46].

Clustering algorithms partition data into a certain number of clusters (groups, subsets, or categories). There is no universally agreed upon definition [28]. Most researchers describe a cluster by considering the internal homogeneity and the external separation [34, 40, 45], i.e., patterns in the same cluster should be similar to each other, while patterns in different clusters should not. Both the similarity and the dissimilarity should be examinable in a clear and meaningful way. Here, we give the simple mathematical descriptions of partitional clustering and hierarchical clustering, based on [40].

Given a set of N input patterns $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_j, \dots, \mathbf{x}_N\}$, where $\mathbf{x}_j = (x_{j1}, x_{j2}, \dots, x_{jd})^T \in \mathbb{R}^d$ and each x_{ji} measure is said to be a feature (attribute, dimension, or variable),

- (Hard) partitional clustering attempts to seek a K -partition of \mathbf{X} , $C = \{C_1, \dots, C_K\}$ ($K \leq N$), such that
 - $C_i \neq \phi, i = 1, \dots, K$;
 - $\bigcup_{i=1}^K C_i = \mathbf{X}$;
 - $C_i \cap C_j = \phi, i, j = 1, \dots, K$ and $i \neq j$.
- Hierarchical clustering attempts to construct a tree-like nested structure partition of \mathbf{X} , $H = \{H_1, \dots, H_Q\}$ ($Q \leq N$), such that $C_i \in H_m, C_j \in H_l$, and $m > l$ imply $C_i \subset C_j$ or $C_i \cap C_j = \phi$ for all $i, j \neq i, m, l = 1, \dots, Q$.

Clustering consists of four basic steps:

1. *Feature selection or extraction.* As pointed out in [9] and [46], feature selection chooses distinguishing features from a set of candidates, while feature extraction utilizes some transformations to generate useful and novel features.
2. *Clustering algorithm design or selection.* The step is usually combined with the proximity measure selection and the criterion function construction. The proximity measure directly affects the formation of the resulting clusters. Once it is chosen, the clustering criterion construction makes the partition of clusters an optimization problem, which is well defined mathematically.
3. *Cluster validation.* Effective evaluation standards and criteria are important to provide the users with a degree of confidence for the clustering results derived from the used algorithms.
4. *Results interpretation.* Experts in the relevant fields interpret the data partition. Further analysis, even experiments, may be required to guarantee the reliability of extracted knowledge.

The remainder of the paper is organized as follows. In Section 2, we briefly review major clustering techniques rooted in machine learning, computer science, and statistics. More discussions on computational intelligence technologies based clustering are given in Section 3 and 4. We illustrate five important applications of the clustering algorithms in Section 5. We conclude the paper and summarize the potential challenges in Section 6.

2 Clustering Algorithms

Different objects and criteria usually lead to different taxonomies of clustering algorithms [28, 40, 45, 46]. A rough but widely agreed frame is to classify clustering techniques as hierarchical clustering and partitional clustering [28, 46], as described in Section 1.

Hierarchical clustering (HC) algorithms organize data objects with a sequence of partitions, either from singleton clusters to a cluster including all individuals or vice versa [28]. The results of HC are usually depicted by a binary tree or dendrogram. The root node of the dendrogram represents the

whole data set and each leaf node is regarded as a data object. The intermediate nodes thus describe the extent that the objects are proximal to each other; and the height of the dendrogram usually expresses the distance between each pair of objects or clusters, or an object and a cluster. The ultimate clustering results can be obtained by cutting the dendrogram at different levels. This representation provides very informative descriptions and visualization for the potential data clustering structures, especially when real hierarchical relations exist in the data. However, classical HC algorithms lack robustness and are sensitive to noise and outliers. The computational complexity for most of HC algorithms is at least $O(N^2)$ and this high cost limits their application in large-scale data.

In contrast to hierarchical clustering, partitional clustering assigns a set of objects into a pre-specified K clusters without a hierarchical structure. The principally optimal partition is infeasible in practice, due to the expensive computation [28]. Therefore, heuristic algorithms have been developed in order to seek approximate solutions. One of the important factors in partitional clustering is the criterion function [40], and the sum of squared error function is one of the most widely used, which aims to minimize the cost function. The K -means algorithm is the best-known squared error-based clustering algorithm, which is very simple and can be easily implemented in solving many practical problems [54]. It can work very well for compact and hyperspherical clusters. The time complexity of K -means is $O(NKd)$, which makes it scale well for large data sets. The major disadvantages of K -means lie in its dependence on the initial partitions and the identification of the number of clusters, the convergence problem, and the sensitivity to noise. Many variants of K -means have been proposed to address these problems, as summarized in [87]. Particularly, the stochastic optimization methods, such as the genetic algorithms, can explore the solution space more flexibly and efficiently and find the approximate global optimum [38]. However, the potential price are the difficulty of parameter selection and expensive computational complexity [87].

Hard or crisp clustering only assigns an object to one cluster. However, a pattern may also be allowed to belong to all clusters with a degree of membership, $u_{i,j} \in [0, 1]$, which represents the j^{th} membership coefficient of the i^{th} object in the cluster and satisfies the following two constraints: $\sum_{i=1}^c u_{i,j} = 1, \forall j$ and $\sum_{j=1}^N u_{i,j} < N, \forall i$, as introduced in fuzzy set theory [89]. This is particularly useful when the boundaries among the clusters are not well separated and ambiguous. Moreover, the memberships may help us discover more sophisticated relations between a given object and the disclosed clusters. The typical example is Fuzzy c -Means algorithm, together with its numerous variants [8, 43, 87].

In the probabilistic view, data points in different clusters are assumed to be generated according to different probability distributions. The mixture probability density for the whole data set is expressed as $p(\mathbf{x}|\eta) = \sum_{i=1}^K p(\mathbf{x}|C_i, \eta_i)P(C_i)$, where $\eta = (\eta_1, \dots, \eta_K)$ is the parameter vector, $P(C_i)$

is the prior probability and $\sum_{i=1}^K P(C_i) = 1$, and $p(\mathbf{x}|C_i, \eta_i)$ is the conditional probability density. The component density can be different types of functions, or the same family, but with different parameters. If these distributions are known, finding the clusters of a given data set is equivalent to estimating the parameters of several underlying models, where Maximum Likelihood (ML) estimation can be used [22]. In the case that the solutions of the likelihood equations of ML cannot be obtained analytically, the Expectation-Maximization (EM) algorithm can be utilized to approximate the ML estimates through an iterative procedure [56]. As long as the parameter vector is decided, the posterior probability for assigning a data point to a cluster can be easily calculated with Bayes's theorem.

3 Neural Networks-Based Clustering

In competitive neural networks, active neurons reinforce their neighborhood within certain regions, while suppressing the activities of other neurons (so-called on-center/off-surround competition). Typical examples include Learning Vector Quantization (LVQ) and Self-Organizing Feature Maps (SOFM) [48, 49]. Intrinsically, LVQ performs supervised learning, and is not categorized as a clustering algorithm [49, 61]. But its learning properties provide an insight to describe the potential data structure using the prototype vectors in the competitive layer. By pointing out the limitations of LVQ, including sensitivity to initiation and lack of a definite clustering object, Pal, Bezdek and Tsao proposed a general LVQ algorithm for clustering, known as GLVQ [61]. They constructed the clustering problem as an optimization process based on minimizing a loss function, which is defined on the locally weighted error between the input pattern and the winning prototype. They also showed the relations between LVQ and the online K -means algorithm.

The objective of SOFM is to represent high-dimensional input patterns with prototype vectors that can be visualized in a usually two-dimensional lattice structure [48, 49]. Each unit in the lattice is called a neuron, and adjacent neurons are connected to each other, which gives the clear topology of how the network fits itself to the input space. Input patterns are fully connected to all neurons via adaptable weights, and during the training process, neighboring input patterns are projected into the lattice, corresponding to adjacent neurons. In this sense, some authors prefer to think of SOFM as a method to displaying latent data structure in a visual way rather than a clustering approach [61]. Basic SOFM training goes through the following steps and a variety of variants of SOFM can be found in [49].

1. Define the topology of the SOFM; Initialize the prototype vectors $\mathbf{m}_i(0)$, $i = 1, \dots, K$ randomly;
2. Present an input pattern \mathbf{x} to the network; Choose the winning node J that is closest to \mathbf{x} , i.e. $J = \arg \min_j \{\|\mathbf{x} - \mathbf{m}_j\|\}$;
3. Update prototype vectors $\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{ci}(t)[\mathbf{x} - \mathbf{m}_i(t)]$, where $h_{ci}(t)$ is the neighborhood function that is often defined as $h_{ci}(t) = \alpha(t) \exp(-\frac{\|\mathbf{r}_c - \mathbf{r}_i\|^2}{2\sigma^2(t)})$, where $\alpha(t)$ is the monotonically decreasing learning rate, \mathbf{r} represents the position of corresponding neuron, and $\sigma(t)$ is the monotonically decreasing kernel width function, or

$$h_{ci}(t) = \begin{cases} \alpha(t) & \text{if node } c \text{ belongs to neighborhood of winning node } J \\ 0 & \text{otherwise} \end{cases}$$

4. Repeat steps 2 and 3 until no change of neuron position that is more than a small positive number is observed.

Adaptive resonance theory (ART) was developed, by Carpenter and Grossberg, as a solution to the plasticity and stability dilemma [11, 13]. ART can learn arbitrary input patterns in a stable, fast and self-organizing way, thus overcoming the effect of learning instability that plagues many other competitive networks. ART is not, as is popularly imagined, a neural network architecture. It is a learning theory, that resonance in neural circuits can trigger fast learning. As such, it subsumes a large family of current and future neural networks architectures, with many variants. ART1 is the first member, which only deals with binary input patterns [11], although it can be extended to arbitrary input patterns by a variety of coding mechanisms. ART2 extends the applications to analog input patterns [12] and ART3 introduces a new mechanism originating from elaborate biological processes to achieve more efficient parallel search in hierarchical structures [14]. By incorporating two ART modules, which receive input patterns (ART_a) and corresponding labels (ART_b) respectively, with an inter-ART module, the resulting ARTMAP system can be used for supervised classifications [15]. The match tracking strategy ensures the consistency of category prediction between two ART modules by dynamically adjusting the vigilance parameter of ART_a . A similar idea, omitting the inter-ART module, is known as LAPART [42].

The basic ART1 architecture consists of two-layer nodes (see Figure 1), the feature representation field F_1 and the category representation field F_2 . They are connected by adaptive weights, bottom-up weight matrix \mathbf{W}^{12} and top-down weight matrix \mathbf{W}^{21} . The prototypes of clusters are stored in layer F_2 . After it is activated according to the winner-takes-all competition, an expectation is reflected in layer F_1 , and compared with the input pattern. The orienting subsystem with the specified vigilance parameter ρ ($0 \leq \rho \leq 1$) determines whether the expectation and the input are closely matched, and therefore controls the generation of new clusters. It is clear that the larger ρ is, the more clusters are generated. Once weight adaptation occurs, both bottom-up and top-down weights are updated simultaneously. This is called

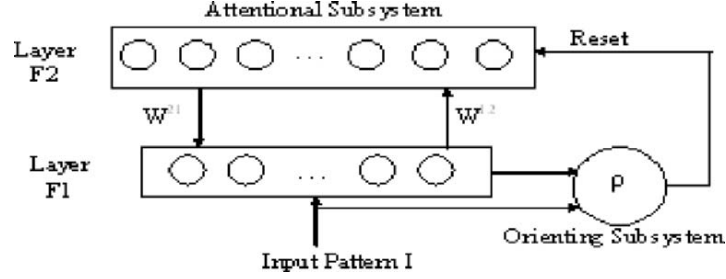


Fig. 1. ART1 Architecture.

resonance, from which the name comes. The ART1 algorithm can be described as follows:

1. Initialize weight matrices \mathbf{W}^{12} and \mathbf{W}^{21} as $W_{ij}^{12} = \alpha_j$, where α_j are sorted in a descending order and satisfies $0 < \alpha_j < 1/(\beta + |\mathbf{x}|)$ for $\beta > 0$ and any binary input pattern \mathbf{x} , and $W_{ji}^{21} = 1$;
2. For a new pattern \mathbf{x} , calculate the input from layer F_1 to layer F_2 as

$$T_j = \sum_{i=1}^d W_{ij}^{12} x_i = \begin{cases} |\mathbf{x}| \alpha_j & \text{if } j \text{ is uncommitted (first activated),} \\ \frac{|\mathbf{x} \cap \mathbf{W}_j^{21}|}{\beta + |\mathbf{W}_j^{21}|} & \text{if } j \text{ is committed,} \end{cases}$$

where \cap represents the logic AND operation.

3. Activate layer F_2 by choosing node J with the winner-takes-all rule $T_J = \max_j \{T_j\}$;
4. Compare the expectation from layer F_2 with the input pattern.
If $\rho \leq |\mathbf{x} \cap \mathbf{W}_J^{21}| / |\mathbf{x}|$, then go to step 5a, otherwise go to step 5b.
5. a Update the corresponding weights for the active node as $\mathbf{W}_J^{12}(\text{new}) = \frac{\mathbf{x} \cap \mathbf{W}_J^{21}(\text{old})}{\beta + |\mathbf{x} \cap \mathbf{W}_J^{21}(\text{old})|}$ and $\mathbf{W}_J^{21}(\text{new}) = \mathbf{x} \cap \mathbf{W}_J^{21}(\text{old})$;
- b Send a reset signal to disable the current active node by the orienting subsystem and return to step 3;
6. Present another input pattern, return to step 2 until all patterns are processed.

Note the relation between ART network and other clustering algorithms described in traditional and statistical language. Moore used several clustering algorithms to explain the clustering behaviors of ART1 and therefore induced and proved a number of important properties of ART1, notably its equivalence to varying K -means clustering [57]. She also showed how to adapt these algorithms under the ART1 framework. In [83] and [84], the ease with which ART may be used for hierarchical clustering is also discussed.

Fuzzy ART (FA) benefits the incorporation of fuzzy set theory and ART [16]. FA maintains similar operations to ART1 and uses the fuzzy set operators to replace the binary operators, so that it can work for all real data sets.

FA exhibits many desirable characteristics such as fast and stable learning and atypical pattern detection. The criticisms for FA are mostly focused on its inefficiency in handling noise and the deficiency of hyperrectangular representation for clusters [4, 5, 81]. Williamson described Gaussian ART (GA) to overcome these shortcomings, in which each cluster is modeled with Gaussian distribution and represented as a hyperellipsoid geometrically [81]. GA does not inherit the offline fast learning property of FA, as indicated by Anagnostopoulos et al. [3], who proposed Ellipsoid ART (EA) for hyperellipsoidal clusters to explore a more efficient representation of clusters, while keeping important properties of FA [3]. Baraldi and Alpaydin proposed Simplified ART (SART) following their general ART clustering networks frame, which is described through a feed-forward architecture combined with a match comparison mechanism [4]. As specific examples, they illustrated Symmetric Fuzzy ART (SFART) and Fully Self-Organizing SART (FOSART) networks. These networks outperform ART1 and FA according to their empirical studies [4].

Like ART family, there are other neural network-based constructive clustering algorithms that can adaptively and dynamically adjust the number of clusters rather than use a pre-specified and fixed number, as K -means and SOFM require [26, 62, 65, 90].

4 Kernel-Based Clustering

Kernel-based learning algorithms [60, 71, 80] are based on Cover's theorem. By nonlinearly transforming a set of complex and nonlinearly separable patterns into a higher-dimensional feature space, we can obtain the possibility to separate these patterns linearly [41]. The difficulty of curse of dimensionality can be overcome by the kernel trick, arising from Mercer's theorem [41]. By designing and calculating an inner-product kernel, we can avoid the time-consuming, sometimes even infeasible process, to explicitly describe the nonlinear mapping and compute the corresponding points in the transformed space.

In [72], Schölkopf, Smola and Müller depicted a kernel- K -means algorithm in the online mode. Suppose we have a set of patterns $\mathbf{x}_j \in \mathbb{R}^d, j = 1, \dots, N$, and a nonlinear map $\Phi : \mathbb{R}^d \rightarrow F$. Here, F represents a feature space with arbitrarily high dimensionality. The object of the algorithm is to find K centers so that we can minimize the distance between the mapped patterns and their closest center $\|\Phi(\mathbf{x}) - \mathbf{m}_l\|^2 = \|\Phi(\mathbf{x}) - \sum_{j=1}^N \tau_{lj}\Phi(\mathbf{x}_j)\|^2 = k(\mathbf{x}, \mathbf{x}) - 2 \sum_{j=1}^N \tau_{lj}k(\mathbf{x}, \mathbf{x}_j) + \sum_{i,j=1}^N \tau_{li}\tau_{lj}k(\mathbf{x}_i, \mathbf{x}_j)$, where \mathbf{m}_l is the center for the l^{th} cluster and lies in a span of $\Phi(\mathbf{x}_1), \dots, \Phi(\mathbf{x}_N)$, and $k(\mathbf{x}, \mathbf{x}_j) = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}_j)$ is the inner-product kernel.

Define the cluster assignment variable

$$C_{jl} = \begin{cases} 1 & \text{if } \mathbf{x}_j \text{ belongs to cluster } l, \\ 0 & \text{otherwise,} \end{cases}$$

then the kernel- K -means algorithm can be formulated as below:

1. Initialize the centers \mathbf{m}_l with the first i , ($i \geq K$), observation patterns;
2. Take a new pattern \mathbf{x}_{i+1} and calculate $C_{(i+1)h}$ as

$$C_{(i+1)h} = \begin{cases} 1 & \text{if } \|\Phi(\mathbf{x}_{i+1}) - \mathbf{m}_h\|^2 < \|\Phi(\mathbf{x}_{i+1}) - \mathbf{m}_j\|^2, \forall j \neq h; \\ 0 & \text{otherwise} \end{cases};$$

3. Update the mean vector \mathbf{m}_h whose corresponding $C_{(i+1)h}$ is 1,

$$\mathbf{m}_h^{new} = \mathbf{m}_h^{old} + \xi(\Phi(\mathbf{x}_{i+1}) - \mathbf{m}_h^{old}),$$

where $\xi = C_{(i+1)h} / \sum_{j=1}^{i+1} C_{jh}$;

4. Adapt the coefficients τ_{hj} for each $\Phi(\mathbf{x}_j)$ as

$$\tau_{hj}^{new} = \begin{cases} \tau_{hj}^{old}(1 - \xi) & \text{for } j \neq i + 1; \\ \xi & \text{for } j = i + 1; \end{cases}$$

5. Repeat the steps 2-4 until convergence is achieved.

Two variants of kernel- K -means were introduced in [20], motivated by SOFM and ART networks.

An alternative kernel-based clustering approach is in [30]. The problem was formulated to determine an optimal partition $\mathbf{\Gamma}$ to minimize the trace of within-group scatter matrix in the feature space,

$$\begin{aligned} \mathbf{\Gamma} &= \arg \min_{\mathbf{\Gamma}} Tr(S_W^{\Phi}) \\ &= \arg \min_{\mathbf{\Gamma}} Tr \left\{ \frac{1}{N} \sum_{i=1}^K \sum_{j=1}^N \gamma_{ij} (\Phi(\mathbf{x}_j) - \mathbf{m}_i) (\Phi(\mathbf{x}_j) - \mathbf{m}_i)^T \right\} \\ &= \arg \min_{\mathbf{\Gamma}} \sum_{i=1}^K \xi_i R(\mathbf{x} | C_i) \end{aligned}$$

where $\xi_i = N_i/N$, $R(\mathbf{x} | C_i) = \frac{1}{N_i^2} \sum_{l=1}^N \sum_{j=1}^N \gamma_{il} \gamma_{ij} k(\mathbf{x}_l, \mathbf{x}_j)$, and N_i is the total number of patterns in the i^{th} cluster. The kernel function utilized in this case is the radial basis function.

Ben-Hur et al. presented a new clustering algorithm, Support Vector Clustering (SVC), in order to find a set of contours used as the cluster boundaries in the original data space [6]. These contours can be formed by mapping back the smallest enclosing sphere, which contains all the data points in the transformed feature space. Chiang and Hao extended the idea by considering each cluster corresponding to a sphere, instead of just one sphere in SVC [19]. They adopted a mechanism similar to ART to dynamically generate clusters. When an input is presented, clusters compete based on some pre-specified distance function. A validation test is performed to ensure the eligibility of the cluster to represent the input pattern. A new cluster is created as a result of the failure of all clusters available to the vigilance test. Furthermore, the distance

between the input pattern and the cluster center and the radius of the sphere provide a way to calculate the fuzzy membership function.

Kernel-based clustering algorithms have many advantages:

1. It is more possible to obtain a linearly separable hyperplane in the high-dimensional, or even infinite feature space;
2. They can form arbitrary clustering shapes other than hyperellipsoid and hypersphere;
3. Kernel-based clustering algorithms, like SVC, have the capability of dealing with noise and outliers;
4. For SVC, there is no requirement for prior knowledge to determine the system topological structure. In [30], Girolami performed eigenvalue decomposition on the kernel matrix in the high-dimensional feature space and used the dominant K components in the decomposition summation as an indication of the possible existence of K clusters.

5 Applications

Clustering has been applied in a wide variety of fields [28, 46]. We illustrate the applications of clustering algorithms in five interesting and important aspects, as described through Subsection 5.1 to 5.5.

5.1 Traveling Salesman Problem

The Traveling Salesman Problem (TSP) is one of the most studied examples in NP-complete problems. Given a complete undirected graph $G = (V, E)$, where V is a set of vertices and E is a set of edges with an associated non-negative integer cost, the most general form of the TSP is equivalent to finding any Hamiltonian cycle, which is a tour over G that begins and ends at the same vertex and visits other vertices exactly once. The more common form of the problem is the optimization problem of trying to find the shortest Hamiltonian cycle, and in particular, the most common is the Euclidean version, where the vertices and edges all lie in the plane. Mulder and Wunsch applied a divide-and-conquer clustering technique, with ART networks, to scale the problem to a million cities [59], and later, to 25 million cities [85]. The divide and conquer paradigm gives the flexibility to hierarchically break large problems into arbitrarily small clusters depending on what trade-off between accuracy and speed is desired. In addition, the sub-problems provide an excellent opportunity to take advantage of parallel systems for further optimization. As the first stage of the algorithm, ART is used to cluster the cities. The clusters were then each passed to a version of the Lin-Kernighan algorithm. The last step combines the subtours back into one complete tour. Tours with good quality for up to 25 million cities were obtained within 13,500 seconds on a 2GHz AMD Athlon MP processor with 512M of DDR RAM.

5.2 Bioinformatics - Gene Expression Data Analysis

Genome sequencing projects have achieved great advance in recent years. However, these successes can only be seen as the first step towards understanding the functions of genes and proteins and the interactions among cellular molecules. DNA microarray technologies provide an effective way to measure expression levels of tens of thousands of genes simultaneously under different conditions, which makes it possible to investigate gene activities of the whole genome [24, 53]. We demonstrate the applications of clustering algorithms in analyzing the explosively increasing gene expression data through both genes and tissues clustering.

Cluster analysis, for grouping functionally similar genes, gradually became popular after the successful application of the average linkage hierarchical clustering algorithm for the expression data of budding yeast *Saccharomyces cerevisiae* and reaction of human fibroblasts to serum by Eisen et al. [25]. They used the Pearson correlation coefficient to measure the similarity between two genes, and provided a very informative visualization of the clustering results. Their results demonstrate that functionally similar genes tend to reside in the same clusters formed by their expression pattern. Tomayo et al. made use of SOFM to cluster gene expression data and its application in hematopoietic differentiation provided new insight for further research [77]. Since many genes usually display more than one function, fuzzy clustering may be more effective in exposing these relations [21]. Gene expression data is also important to elucidate the genetic regulation mechanism in a cell. Spellman et al. clustered 800 genes according to their expression during the yeast cell cycle [75]. Analyses of 8 major gene clusters unravel the connection between co-expression and co-regulation. Tavazoie et al. partitioned 3,000 genes into 30 clusters with the *K*-means algorithm [78]. For each cluster, 600 base pairs upstream sequences of the genes were searched for potential motifs. 18 motifs were found from 12 clusters in their experiments and 7 of them can be verified according to previous empirical results. Figure 2 (a) and (b) illustrate the application of hierarchical clustering and SOFM for the small round blue-cell tumors (SR-BCTs) data set, which consists of the measurement of the expression levels of 2,308 genes across 83 samples [47]. Hierarchical clustering was performed by the program CLUSTER and the results were visualized by the program TreeView, developed by Eisen in Stanford University. The software package GeneCluster, developed by Whitehead Institute/MIT Center for Genome Research, was used for SOFM analysis.

In addition to genes clustering, tissues clustering are valuable in identifying samples that are in the different disease states, discovering or predicting different cancer types, and evaluating the effects of novel drugs and therapies [1, 31, 70]. Golub et al. described the restriction of traditional cancer classification methods and divided cancer classification as class discovery and class prediction. They utilized SOFM to discriminate two types of human acute leukemias: acute myeloid leukemia (AML) and acute lymphoblastic leukemia

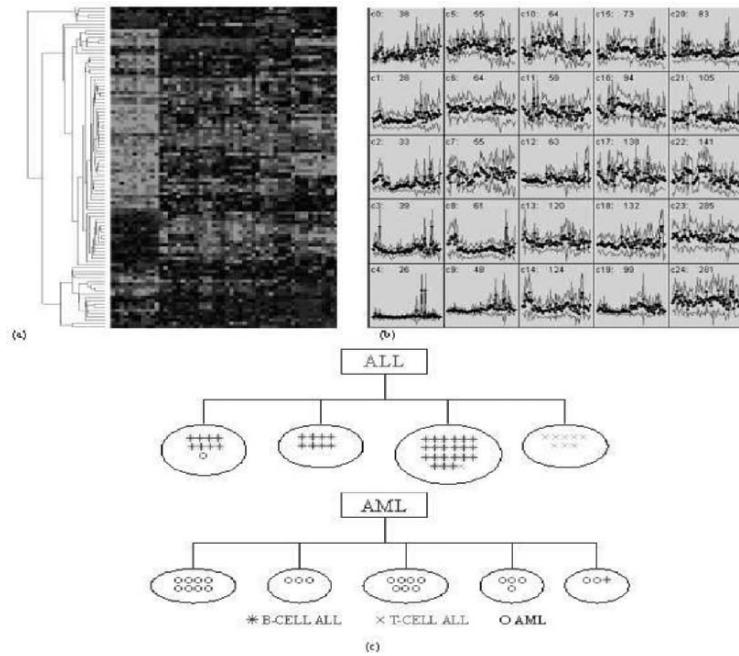


Fig. 2. Clustering for Gene Expression Data. (a) Hierarchical clustering result for the 100 selected genes from the SRBCT data set. The gene expression matrix is visualized through a color scale; (b) SOFM clustering result for all the 2308 genes of SRBCT data set. A 5x5 SOFM is used and 25 clusters are formed. Each cluster is represented by the average values; (c) EA clustering result for ALL/AML data set. EA effectively separates the two ALL subsets.

(ALL) [31]. Two subsets of ALL, with quite different origin of lineage, can be well separated. This result is also confirmed by the analysis with Ellipsoidal ART network, as illustrated in Figure 2 (c) [86]. Alizadeh et al. successfully distinguished two molecularly distinct subtypes of diffuse large B-cell lymphoma, which cause high percentage failure in clinical treatment, based on their gene expression profiles [1]. Scherf et al. constructed a gene expression database to study the relationship between genes and drugs for 60 human cancer cell lines, which provides an important criterion for therapy selection and drug discovery [70]. Moreover, gene expression profiles are extended for patient survival analysis. Rosenwald et al. used hierarchical clustering to divide diffuse large-B-cell lymphoma, and the Kaplan-Meier estimates of the survival probabilities for each group show significant difference [66].

Furthermore, bi-clustering concept has been raised, referring to the clustering of both the genes (rows) and samples or conditions (columns) simultaneously [17]. Therefore, it is more effective in specifying a set of genes related

to some certain experimental conditions or cellular processes. A good survey paper on bi-clustering can be found in [55].

5.3 Bioinformatics - DNA or Protein Sequences Clustering

In recent decades, DNA and protein sequences grew explosively [23, 37]. For example, the recent statistics released on June 15, 2005 (Release 148.0) shows that there are 49,398,852,122 bases from 45,236,251 reported sequences in GenBank database [29]. The information hidden in the sequences offers a cue to identify functions of genes and proteins. In contrast to sequence comparison and search, cluster analysis provides a more effective way to discover complicated relations among these sequences. We summarize the following clustering applications for DNA and protein sequences:

1. Function recognition of uncharacterized genes or proteins [36];
2. Structure identification of large-scale DNA or protein databases [69, 74];
3. Redundancy decrease of large-scale DNA or protein databases [52];
4. Domain identification [27, 35];
5. EST (Expressed Sequence Tag) clustering [10].

Since biology sequential data are expressed in an alphabetic form, conventional measure methods are not appropriate. If a sequence comparison is regarded as a process of transforming a given sequence to another with a series of substitution, insertion, and deletion operations, the distance between the two sequences can be defined by virtue of the minimum number of required operations, known as edit distance [37, 68]. These edit operations are weighted according to some prior domain knowledge and the distance herein is equivalent to the minimum cost to complete the transformation. In this sense, the similarity or distance between two sequences can be reformulated as an optimal alignment problem, which fits well in the framework of dynamic programming [23]. However, for the basic alignment algorithms, the computation complexity is $O(NM)$, which is incapable of dealing with tons of nucleic acids and amino acids in the current DNA or protein databases [23]. In practice, sequence comparison or proximity measure is achieved via some heuristics, such as BLAST and FASTA with their variants [2, 63]. The key idea of these methods is to identify regions that may have potentially high matches, with a list of pre-specified high-scoring words, at an early stage. Therefore, further search only needs to focus on these regions with expensive but accurate algorithms.

Generally, there are three strategies for clustering DNA or protein sequence data. Clustering algorithms can either directly operate on a proximity measure or are based on feature extraction. They also can be constructed according to the statistical models to describe the dynamics of each group of sequences. Somervuo and Kohonen illustrated an application of SOFM to cluster protein sequences in SWISSPROT database [74]. FASTA was used to calculate the sequence similarity. Based on the similarity measure of gapped BLAST, Sasson

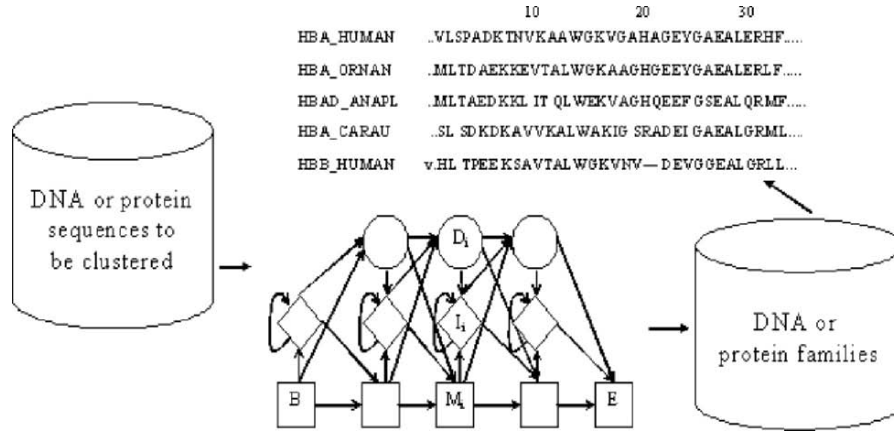


Fig. 3. DNA or Protein Clustering with HMMs. The result shown here is the part of the alignment of 9 globin sequences obtained from SWISS-PROT protein sequences databank.

et al. utilized an agglomerative hierarchical clustering paradigm to cluster all protein sequences in SWISSPROT [69]. In contrast with the proximity-based methods, Guralnik and Karypis transformed protein or DNA sequences into a new feature space, based on the detected sub-patterns working as the sequence features, and clustered with the *K*-means algorithm [36]. The method is immune from all-against-all expensive sequence comparison. However, it is largely dependent on the feature selection process, which may mislead the analysis. Krogh demonstrated the power of hidden Markov models (HMMs) [64] in biological sequences modeling and clustering of protein families [51]. Figure 3 depicts a typical clustering analysis of protein or DNA sequences with HMMs, in which match states (*M*), insert states (*I*), and delete states (*D*) are represented as rectangles, diamonds, and circles, respectively [23, 51]. These states correspond to substitution, insertion, and deletion in edit operations. For convenience, a begin state (*B*) and an end (*E*) state are added to the model. Either 4-letter nucleotide alphabets or 20-letter amino acid alphabets are generated from match and insert states according to some emission probability distributions. Delete states do not produce any symbols, and are used to skip the match states. *K* HMMs are required in order to describe *K* clusters, or families (subfamilies), which are regarded as a mixture model and proceeded with an EM learning algorithm. This paradigm models clusters directly from original data without additional process that may cause information loss. They provide more intuitive ways to capture the dynamics of data and more flexible means to deal with variable length sequences. However, determining the number of model components remains a complicated and uncertain process [73]. Also, the model selected is required to have sufficient complexity, in order to interpret the characteristics of data.

5.4 Dimensionality Reduction - Human Face Expression Recognition

Nowadays, it is more common to analyze data with very high dimensionality, which causes the problem curse of dimensionality [7, 41]. Fortunately, in practice, many high-dimensional data usually have an intrinsic dimensionality that is much lower than the original dimension [18]. Although strictly speaking, dimension reduction methods do not belong to clustering algorithms, they are still very important in cluster analysis. Dimensionality reduction not only reduces the computational cost and makes the high-dimensional data processible, but provides users with a clear picture and good visual examination of the data of interest. However, dimensionality reduction methods inevitably cause some information loss, and may damage the interpretability of the results, even distort the real clusters.

Unlike the typical linear components extraction techniques, like principle component analysis [22] and independent component analysis [44], Locally Linear Embedding (LLE) algorithm focuses on nonlinear dimensionality reduction [67]. LLE emphasizes the local linearity of the manifold and assumes that the local relations in the original data space (D -dimensional) are also preserved in the projected low-dimensional space (L -dimensional). This is represented through a weight matrix, describing how each point is related to the reconstruction of another data point. Therefore, the procedure for dimensional reduction can be constructed as the problem that finding L -dimensional vectors \mathbf{y}_i so that the criterion function $\sum_i |\mathbf{y}_i - \sum_j w_{ij} \mathbf{y}_j|$ is minimized. This process makes LLE different from other nonlinear projection techniques, such as Multidimensional Scaling (MDS) [88] and the isometric feature mapping algorithm (ISOMAP), which extends MDS and aims to estimate the shortest path between a pair of points on a manifold, by virtue of the measured input-space distances [79]. It is worth mentioning another method, elastic maps, which seek an optimal configuration of nodes, in a sense of minimum energy, to approximate the data points [32, 33].

An application for human face expression recognition by LLE is illustrated in [67]. The data set includes 2,000 face images from the same individual with different expressions. Each input pattern is a 560-dimensional vector, corresponding to the 20x28 grayscale of the images. The faces are mapped into a two-dimensional space, consisting of the first two constructed coordinates of LLE. The result shows that LLE can effectively find and capture the data structure.

5.5 Document Clustering

Document clustering, particularly web document clustering over Internet, has become more and more important as a result of the requirement for automatic creation of documents hierarchy, information retrieval from documents collections, and search engine results analysis. Steinbach et al. compared the

performance of agglomerative hierarchical clustering and K -means clustering (with one of its variants) on 8 document data sets [76]. Kohonen et al. demonstrated the effectiveness of SOFM for clustering of a large set of documental data, in which 6,840,568 patent abstracts were projected onto a SOFM with 1,002,240 nodes [50].

Different from methods based on individual words analysis, Hammouda and Kamel proposed a phase-based incremental web document clustering system [39]. Each document consists of a set of sentences, each of which includes a sequence of words and is weighted based on the occurrence in the documents, i.e., title, keywords, figure caption, etc., and is indexed through a Document Index Graph (DIG) model. Each node in DIG corresponds to a unique word and each directed edge between a pair of words indicates the order of their occurrence in the document. The similarity measure considers four components, i.e., the number, length, frequencies, and weights of the matching phrases in two documents. The online similarity histogram-based clustering algorithm aims to maintain a high coherency in each cluster, based on the histogram of the cluster's document similarities. A new document is added into a cluster only if it increases the calculated histogram ratio or does not cause a significant decrease of the ratio while still above some minimum threshold.

6 Conclusions

As an important tool for data exploration, cluster analysis examines unlabeled data and includes a series of steps. Clustering algorithms evolve from different research communities, attempt to solve different problems, and have their own pros and cons. Particularly, clustering algorithms, based on computational intelligence technologies, play an important role and attract more intensive efforts. However, there is no universal clustering algorithm that can be applied to solve all problems. In this sense, it is not accurate to say 'best' in the context of clustering algorithms and it is important to select the appropriate methods based on the specific applications. Though we have already seen many examples of successful applications of cluster analysis, there still remain many open problems due to the existence of many inherent uncertain factors. As a conclusion, we summarize the paper with a list of some important issues and research trends for clustering algorithms, however, some more detailed requirements for specific applications will affect these properties.

1. Generate arbitrary shapes of clusters rather than be confined to some particular shape;
2. Handle large volume of data as well as high-dimensional features with acceptable time and storage complexities;
3. Detect and remove possible outliers and noise;
4. Decrease the reliance of algorithms on users-dependent parameters;
5. Have the capability of dealing with newly occurring data without re-learning from the scratch;

6. Be immune to the effects of order of input patterns;
7. Provide some insight for the number of potential clusters without prior knowledge;
8. Show good data visualization and provide users with results that can simplify further analysis;
9. Be capable of handling both numerical and categorical data or be easily adaptable to some other data type.

Acknowledgement

We would like to thank the Eisen Laboratory in Stanford University for use of their CLUSTER and TreeView software and Whitehead Institute/MIT Center for Genome Research for use of their GeneCluster software. We thank S. Mulder for the part on the traveling salesman problem. Partial support for this research from the National Science Foundation, and from the M.K. Finley Missouri endowment, is gratefully acknowledged.

References

1. A. Alizadeh, M. Eisen, R. Davis, C. Ma, I. Lossos, A. Rosenwald, J. Boldrick, H. Sabet, T. Tran, X. Yu, J. Powell, L. Yang, G. Marti, T. Moore, J. Hudson, L. Lu, D. Lewis, R. Tibshirani, G. Sherlock, W. Chan, T. Greiner, D. Weisenburger, J. Armitage, R. Warnke, R. Levy, W. Wilson, M. Grever, J. Byrd, D. Botstein, P. Brown, and L. Staudt: Distinct types of diffuse large B-cell Lymphoma identified by gene expression profiling. *Nature*, 2000, 503–511.
2. S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman: Basic local alignment search tool. *Journal of Molecular Biology*, 1990, 403–410.
3. G. Anagnostopoulos and M. Georgiopoulos: Ellipsoid ART and ARTMAP for incremental unsupervised and supervised learning. In: *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'01)*, 2001, 1221–1226.
4. A. Baraldi and E. Alpaydin: Constructive feedforward ART clustering networks - Part I and II. *IEEE Transactions on Neural Networks*, 2002, 645–677.
5. A. Baraldi and P. Blonda: A survey of fuzzy clustering algorithms for pattern recognition - Part I and II. *IEEE Transactions on Systems, Man, And Cybernetics - Part B: Cybernetics*, 1999, 778–801.
6. A. Ben-Hur, D. Horn, H. Siegelmann, and V. Vapnik: Support vector clustering. *Journal of Machine Learning Research*, 2001, 125–137.
7. K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft: When is nearest neighbor meaningful. In: *Proceedings of 7th International Conference on Database Theory*, 1999, 217–235.
8. J. Bezdek: *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.
9. C. Bishop: *Neural networks for pattern recognition*. Oxford University Press, 1995.

10. J. Burke, D. Davison, and W. Hide: d2_Cluster: a validated method for clustering EST and full-length cDNA sequences. *Genome Research*, 1999, 1135–1142.
11. G. Carpenter and S. Grossberg: A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, 1987, 54–115.
12. G. Carpenter and S. Grossberg: ART2: Self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, 1987, 4919–4930.
13. G. Carpenter and S. Grossberg: The ART of adaptive pattern recognition by a self-organizing neural network. *IEEE Computer*, 1988, 77–88.
14. G. Carpenter and S. Grossberg: ART3: Hierarchical search using chemical transmitters in self-organizing pattern recognition Architectures. *Neural Networks*, 1990, 129–152.
15. G. Carpenter, S. Grossberg, and J. Reynolds: ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks*, 1991, 169–181.
16. G. Carpenter, S. Grossberg, and D. Rosen: Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 1991, 759–771.
17. Y. Cheng and G. Church: Biclustering of expression data. In: *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB '00)*, 2000, 93–103.
18. V. Cherkassky, and F. Mulier: *Learning from Data: Concepts, Theory, and Methods*, John Wiley & Sons, Inc., 1998.
19. J. Chiang and P. Hao: A new kernel-based fuzzy clustering approach: support vector clustering with cell growing. *IEEE Transactions on Fuzzy Systems*, 2003, 518–527.
20. J. Corchado and C. Fyfe: A comparison of kernel methods for instantiating case based reasoning systems. *Computing and Information Systems*, 2000, 29–42.
21. D. Dembélé and P. Kastner: Fuzzy c-means method for clustering microarray data. *Bioinformatics*, 2003, 973–980.
22. R. Duda, P. Hart, and D. Stork: *Pattern Classification*. 2nd edition, John Wiley & Sons, Inc., 2001.
23. R. Durbin, S. Eddy, A. Krogh, and G. Mitchison: *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.
24. M. Eisen and P. Brown: DNA arrays for analysis of gene expression. *Methods Enzymol*, 1999, 179–205.
25. M. Eisen, P. Spellman, P. Brown, and D. Botstein: Cluster analysis and display of genome-wide expression patterns. In: *Proceedings of the National Academy of Science*, 1998, 14863–14868.
26. T. Eltoft and R. deFigueiredo: A new neural network for cluster-detection-and-labeling. *IEEE Transactions on Neural Networks*, 1998, 1021–1035.
27. A. Enright and C. Ouzounis: GeneRAGE: A robust algorithm for sequence clustering and domain detection. *Bioinformatics*, 2000, 451–457.
28. B. Everitt, S. Landau, and M. Leese: *Cluster Analysis*, Arnold, 2001.
29. GenBank Release Notes 148.0, 2005.
30. M. Girolami: Mercer kernel based clustering in feature space. *IEEE Transactions on Neural Networks*, 2002, 780–784.

31. T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and E. Lander: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 1999, 531–537.
32. A. Gorban, A. Pitenko, A. Zinovyev, and D. Wunsch II: Visualization of any data with elastic map method. In: *Proc. Artificial Neural Networks in Engineering*, 2001.
33. A. Gorban, A. Zinovyev, and D. Wunsch II: Application of the method of elastic maps in analysis of genetic texts. In: *Proc. International Joint Conference on Neural Networks (IJCNN)*, 2003.
34. A. Gordon: *Classification*. 2nd edition, Chapman and Hall/CRC Press, 1999.
35. X. Guan and L. Du: Domain identification by clustering sequence alignments. *Bioinformatics*, 1998, 783–788.
36. V. Guralnik and G. Karypis: A scalable algorithm for clustering sequential data. In: *Proceedings of the 1st IEEE International Conference on Data Mining (ICDM 2001)*, 2001, 179–186.
37. D. Gusfield: *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997.
38. L. Hall, I. özyurt, and J. Bezdek: Clustering with a genetically optimized approach. *IEEE Transactions on Evolutionary Computation*, 1999, 103–112.
39. K. Hammouda and M. Kamel: Efficient phrase-based document indexing for web document clustering. *IEEE Transactions on Knowledge and Data Engineering*, 2004, 1279–1296.
40. P. Hansen and B. Jaumard: Cluster analysis and mathematical programming. *Mathematical Programming*, 1997, 191–215.
41. S. Haykin: *Neural Networks: A Comprehensive Foundation*. 2nd edition, Prentice Hall, 1999.
42. M. Healy, T. Caudell, and S. Smith: A neural architecture for pattern sequence verification through inferencing. *IEEE Transactions on Neural Networks*, 1993, 9–20.
43. F. Höppner, F. Klawonn, and R. Kruse: *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*. Wiley, New York, 1999.
44. A. Hyvärinen: Survey of independent component analysis. *Neural Computing Surveys*, 1999, 94–128.
45. A. Jain and R. Dubes: *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, 1988.
46. A. Jain, M. Murty, and P. Flynn: Data clustering: a review. *ACM Computing Surveys*, 1999, 264–323.
47. J. Khan, J. Wei, M. Ringnér, L. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. Antonescu, C. Peterson, and P. Meltzer: Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, 2001, 673–679.
48. T. Kohonen: The self-organizing map. *Proceedings of the IEEE*, 1990, 1464–1480.
49. T. Kohonen: *Self-Organizing Maps*. 3rd edition, Springer, 2001.
50. T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paatero, and A. Saarela: Self organization of a massive document collection. *IEEE Transactions on Neural Networks*, 2000, 574–585.

51. A. Krogh, M. Brown, I. Mian, K. Sjölander, and D. Haussler: Hidden Markov models in computational biology: applications to protein modeling. *Journal of Molecular Biology*, 1994, 1501–1531.
52. W. Li, L. Jaroszewski, and A. Godzik: Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics*, 2001, 282–283.
53. R. Lipshutz, S. Fodor, T. Gingeras, and D. Lockhart: High density synthetic oligonucleotide arrays. *Nature Genetics*, 1999, 20–24.
54. J. MacQueen: Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley Symposium*, 1967, 281–297.
55. S. Madeira and A. Oliveira: Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2004, 24–45.
56. G. McLachlan and D. Peel: *Finite Mixture Models*. John Wiley & Sons, New York, 2000.
57. B. Moore: ART1 and pattern clustering. In: *Proceedings of the 1988 Connectionist Models Summer School*, Morgan Kaufmann, 1989, 174–185.
58. Y. Moreau, F. Smet, G. Thijs, K. Marchal, and B. Moor: Functional bioinformatics of microarray data: from expression to regulation. *Proceedings of the IEEE*, 2002, 1722–1743.
59. S. Mulder and D. Wunsch: Million city traveling salesman problem solution by divide and conquer clustering with adaptive resonance neural networks. *Neural Networks*, 2003, 827–832.
60. K. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf: An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 2001, 181–201.
61. N. Pal, J. Bezdek, and E. Tsao: Generalized clustering networks and Kohonen’s self-organizing scheme. *IEEE Transactions on Neural Networks*, 1993, 549–557.
62. G. Patané and M. Russo: Fully automatic clustering system. *IEEE Transactions on Neural Networks*, 2002, 1285–1298.
63. W. Pearson: Improved tools for biological sequence comparison. *Proceedings of the National Academy of Science*, 1988, 2444–2448.
64. L. Rabiner: A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 1989, 257–286.
65. S. Ridella, S. Rovetta, and R. Zunino: Plastic algorithm for adaptive vector quantization. *Neural Computing and Applications*, 1998, 37–51.
66. A. Rosenwald, G. Wright, W. Chan, J. Connors, C. Campo, R. Fisher, R. Gascoyne, H. Muller-Hermelink, E. Smeland, and L. Staudt: The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma. *The New England Journal of Medicine*, 2002, 1937–1947.
67. S. Roweis and L. Saul: Nonlinear dimensionality reduction by locally linear embedding. *Science*, 2000, 2323–2326.
68. D. Sankoff and J. Kruskal: *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. CSLI publications, 1999.
69. O. Sasson, N. Linial, and M. Linial: The metric space of proteins - comparative study of clustering algorithms. *Bioinformatics*, 2002, s14–s21.
70. U. Scherf, D. Ross, M. Waltham, L. Smith, J. Lee, L. Tanabe, K. Kohn, W. Reinhold, T. Myers, D. Andrews, D. Scudiero, M. Eisen, E. Sausville, Y. Pommier, D. Botstein, P. Brown, and J. Weinstein: A gene expression database for the molecular pharmacology of cancer. *Nature Genetics*, 2000, 236–44.

71. B. Schölkopf and A. Smola: *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, 2002.
72. B. Schölkopf, A. Smola, and K. Müller: Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 1998, 1299–1319.
73. P. Smyth: Clustering sequences with hidden Markov models. In: *Advances in Neural Information Processing*, M. Mozer, M. Jordan and T. Petsche (eds.), MIT Press, 1997, 648–654.
74. P. Somervuo and T. Kohonen: Clustering and visualization of large protein sequence databases by means of an extension of the self-organizing map. *LNAI 1967*, 2000, 76–85.
75. P. Spellman, G. Sherlock, M. Ma, V. Iyer, K. Anders, M. Eisen, P. Brown, D. Botstein, and B. Futcher: Comprehensive identification of cell cycle-regulated genes of the Yeast *Saccharomyces Cerevisiae* by microarray hybridization. *Mol. Biol. Cell*, 1998, 3273–3297.
76. M. Steinbach, G. Karypis, and V. Kumar: A comparison of document clustering techniques. In: *Proceedings of KDD Workshop on Text Mining*, 2000.
77. P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. Lander, and T. Golub: Interpreting patterns of gene expression with self-organizing maps: Methods and application to Hematopoietic differentiation. In: *Proceedings of the National Academy of Science*, 1999, 2907–2912.
78. S. Tavazoie, J. Hughes, M. Campbell, R. Cho, and G. Church: Systematic determination of genetic network architecture. *Nature Genetics*, 1999, 281–285.
79. J. Tenenbaum, V. Silva, and J. Langford: A global geometric framework for nonlinear dimensionality reduction. *Science*, 2000, 2319–2323.
80. V. Vapnik: *Statistical Learning Theory*. John Wiley & Sons, New York, 1998.
81. J. Williamson: Gaussian ARTMAP: a neural network for fast incremental learning of noisy multidimensional maps. *Neural Networks*, 1996, 881–897.
82. S. Wu, A. Liew, H. Yan, and M. Yang: Cluster analysis of gene expression data based on self-splitting and merging competitive learning. *IEEE Transactions on Information Technology in Biomedicine*, 2004, 5–15.
83. D. Wunsch: *An Optoelectronic Learning Machine: Invention, Experimentation, Analysis of First Hardware Implementation of the ART1 Neural Network*. Ph.D. dissertation, University of Washington, 1991.
84. D. Wunsch, T. Caudell, C. Capps, R. Marks, and R. Falk: An optoelectronic implementation of the adaptive resonance neural network. *IEEE Transactions on Neural Networks*, 1993, 673–684.
85. D. Wunsch and S. Mulder: Evolutionary algorithms, Markov decision processes, adaptive critic designs, and clustering: commonalities, hybridization, and performance. In: *Proceedings of IEEE International Conference on Intelligent Sensing and Information Processing*, 2004.
86. R. Xu, G. Anagnostopoulos, and D. Wunsch: Tissue classification through analysis of gene expression data using a new family of ART architectures. In: *Proceedings of International Joint Conference on Neural Networks 02*, 2002, 300–304.
87. R. Xu and D. Wunsch: Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 2005, 645–678.
88. F. Young and R. Hamer: *Multidimensional Scaling: History, Theory, and Applications*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1987.
89. L. Zadeh: Fuzzy sets. *Information and Control*, 1965, 338–353.
90. Y. Zhang and Z. Liu: Self-splitting competitive learning: a new on-line clustering paradigm. *IEEE Transactions on Neural Networks*, 2002, 369–380.

Energy-Based Image Simplification with Nonlocal Data and Smoothness Terms

Stephan Didas¹, Pavel Mrázek², and Joachim Weickert¹

¹ Faculty of Mathematics and Computer Science, Saarland University, D-66041 Saarbrücken, Germany, {didas,weickert}@mia.uni-saarland.de

² Upek R&D s.r.o., Husinecka 7, 130 00 Prague 3, Czech Republic, pavel.mrazek@upek.com

Summary. Image simplification and smoothing is a very important basic ingredient of a lot of practical applications. In this paper we compare different numerical approaches to solve this image approximation task within a unifying variational approach presented in [8]. For methods based on fixed point iterations we show the existence of fixed points. To speed up the convergence we also use two approaches involving Newton's method which is only applicable for convex penalisers. The running time in practice is studied with numerical examples in 1-D and 2-D.

1 Introduction

The task of image smoothing, simplification and denoising is the subject of various approaches and applications. An initial image is approximated by filtered versions which are smoother or simpler in some sense. Statistical estimation, median or mode filters, nonlinear diffusion, bilateral filtering or regularisation methods are among the tools helpful to reach this aim. Most of these tools somehow incorporate a neighbourhood of the pixel under consideration and perform some kind of averaging on the grey values. One of the earliest examples for such filters has been presented by Lee [7], followed by a lot of successors like the *SUSAN* filter by Smith and Brady [14]. In the context of statistical methods, Polzehl and Spokoiny presented a technique called *adaptive weights smoothing* [11]. The *W-estimator* by Winkler et al. [17] can be related to a spatially weighted *M-smoother* [5]. A very similar evolution is the *bilateral filter* by Tomasi and Manduchi [16], another prominent example for a weighted averaging filter. In its original form it is interestingly not meant to be iterative. There are approaches to relate it to variational principles [4]. In general there are a lot of approaches to give relations between averaging methods and techniques based on minimisation of energy functionals or on partial differential equations [1, 13].

In [8], an energy-based approach has been proposed which allows to consider a whole spectrum of well-known methods as different facets of the same

model. This approach makes use of so-called *Nonlocal Data and Smoothness terms*; thus it will be called *NDS* here. These terms can consider not only information from a small region around a pixel but also make it possible to involve large neighbourhoods. The data term rewards similarity of our filtered image to the given one while the smoothness term penalises high deviations inside a neighbourhood of the evolving image.

The goal of the present paper is to analyse numerical methods for this approach. This paper is organised as follows: Section 2 gives a closer description of the energy functional we deal with and its relations to well-known filtering methods like M-smoothers and the bilateral filter. In Section 3 we discuss different approaches to minimise the NDS functional including a fixed point scheme and Newton's method. Numerical experiments in 1-D and 2-D in Section 4 compare the behaviour and running time of the presented approaches. A summary of the results and an outlook conclude the paper in Section 5.

2 The Filtering Framework

In this section we review the variational model presented in [8] and relate it to other filtering techniques. Let $f, u \in \mathbb{R}^n$ be discrete one- or two-dimensional images. We always denote the initial noisy image of the filtering process with f and the processed one with u . Let $\Omega = \{1, \dots, n\}$ be the index set of all pixels in the images. The pixel positions on the one- or two-dimensional grid will be denoted with $x_i (i \in \Omega)$. That means $|x_i - x_j|^2$ yields the square of the Euclidean distance between the two pixels x_i and x_j in the real line (1-D) or the plane (2-D). This will be referred to as *spatial distance*. The *tonal distance* then is the distance between grey values of two pixels, for example $|u_i - f_j|^2$.

We start with an energy functional involving the tonal distance between u and f :

$$E_D(u) = \sum_{i \in \Omega} \sum_{j \in \Omega} \Psi_D (|u_i - f_j|^2) w_D (|x_i - x_j|^2) \quad (1)$$

The iterative minimisation of such a scheme leads to the well-known W-estimator

$$u_i^0 := f_i, \quad u_i^{k+1} := \frac{\sum_{j \in \Omega} \Psi_D' (|u_i^k - f_j|^2) w_D (|x_i - x_j|^2) f_j}{\sum_{j \in \Omega} \Psi_D' (|u_i^k - f_j|^2) w_D (|x_i - x_j|^2)} \quad (2)$$

This scheme is very similar to another well-established filtering technique known in image processing: the bilateral filter presented by Tomasi and Manduchi [16]. The bilateral filter can be obtained by replacing f_j with u_j in (2). Similar to the above reasoning the bilateral filter can be thought of as minimisation scheme for a nonlocal smoothness term:

$$E_S(u) = \sum_{i \in \Omega} \sum_{j \in \Omega} \Psi_S (|u_i - u_j|^2) w_S (|x_i - x_j|^2) \quad (3)$$

We keep in mind that a minimisation of (3) would lead to a constant image with an arbitrary grey value, since the initial image f does not appear in E_S . Nevertheless, the bilateral filter can be seen as the first step of an iterative minimisation procedure for (3).

The functional E of the NDS filter presented in [8] is a linear combination of both data and smoothness terms:

$$E(u) = \alpha \sum_{i \in \Omega} \sum_{j \in \Omega} \Psi_D (|u_i - f_j|^2) w_D (|x_i - x_j|^2) + (1 - \alpha) \sum_{i \in \Omega} \sum_{j \in \Omega} \Psi_S (|u_i - u_j|^2) w_S (|x_i - x_j|^2) . \quad (4)$$



Here we have incorporated a similarity constraint which can lead to non-flat minimisers and a smoothness constraint. The spatial weights w_D and w_S incorporate the spatial distance between pixel positions x_i and x_j while the tonal weights Ψ_D and Ψ_S penalise high deviations between the corresponding grey values. Table 1 shows some possible choices Ψ for the tonal weights Ψ_D in the data term and Ψ_S in the smoothness term. The NDS functional (4) allows to express a lot of different models, so it is natural that the tonal weights are motivated from different contexts. The list in Table 1 is clearly not meant to be complete since there is a whole variety of possible penalisers at hand. The choice of a special one should be motivated from the type of noise and image, but this is not within the scope of this article.

Table 1. Possible choices for tonal weights Ψ .

$\Psi(s^2)$		$\Psi'(s^2)$	known in the context of
s^2		1	Tikhonov regularisation [15]
$2(\sqrt{s^2 + \varepsilon^2} - \varepsilon)$		$(s^2 + \varepsilon^2)^{-\frac{1}{2}}$	regularised total variation [12]
$2\lambda^2 \left(\sqrt{1 + \frac{s^2}{\lambda^2}} - 1 \right)$		$\left(1 + \frac{s^2}{\lambda^2}\right)^{-\frac{1}{2}}$	nonlinear regularisation, Charbonnier et al. [2]
$\lambda^2 \log \left(1 + \frac{s^2}{\lambda^2} \right)$		$\left(1 + \frac{s^2}{\lambda^2}\right)^{-1}$	nonlinear diffusion, Perona and Malik [10]
$\lambda^2 \left(1 - \exp \left(-\frac{s^2}{\lambda^2} \right) \right)$		$\exp \left(-\frac{s^2}{\lambda^2} \right)$	nonlinear diffusion, Perona and Malik [10]
$\min(s^2, \lambda^2)$		$\begin{cases} 1 & s < \lambda \\ 0 & \text{else} \end{cases}$	segmentation, Mumford and Shah [9]

Two simple examples of functions which can lead as spatial weights are displayed in Table 2. They both have in common that they are symmetric.

Table 2. Possible choices for spatial weights w .

$w(s^2)$		known in the context of
$\begin{cases} 1 & s < \lambda \\ 0 & \text{else} \end{cases}$		hard window
$\exp\left(-\frac{s^2}{\lambda^2}\right)$		soft window

Since in our model (4) we only use $w(s^2)$ we only plug in nonnegative values and this symmetry is obtained automatically. Essentially the same model allows to use nonsymmetric spatial weights, too. We also have chosen spatial weights which are between 0 and 1 and have their maximum in the point 0. This makes sure that the pixel itself is taken into consideration with the highest weight. Centering the spatial weight in the data term around a number different from 0 would perform a shift of the whole image during filtering.

3 Minimisation Methods

After discussing the derivation and the meaning of the NDS functional we now study different methods to minimise it. All numerical minimisation methods are based on conditions on the derivatives of E so we now calculate the first and second partial derivatives of E .

Taking the partial derivatives of the data term (1) yields

$$\begin{aligned} \frac{\partial E_D}{\partial u_k} &= 2 \sum_{j \in \Omega} \Psi'_D(|u_k - f_j|^2) (u_k - f_j) w_D(|x_k - x_j|^2) \\ \frac{\partial^2 E_D}{\partial u_k \partial u_l} &= \begin{cases} 2 \sum_{j \in \Omega} [2\Psi''_D(|u_l - f_j|^2) (u_l - f_j)^2 \\ \quad + \Psi'_D(|u_l - f_j|^2)] w_D(|x_l - x_j|^2) & l = k \\ 0 & l \neq k \end{cases} \end{aligned} \quad (5)$$

In a similar way we calculate the derivatives of the smoothness term (3) which leads to

$$\begin{aligned} \frac{\partial E_S}{\partial u_k} &= 4 \sum_{j \in \Omega} \Psi'_S(|u_k - u_j|^2) (u_k - u_j) w_S(|x_k - x_j|^2) \\ \frac{\partial^2 E_S}{\partial u_k \partial u_l} &= \begin{cases} 4 \sum_{j \in \Omega} [2\Psi''_S(|u_l - u_j|^2) (u_l - u_j)^2 \\ \quad + (1 - \delta_{lj}) \Psi'_S(|u_l - u_j|^2)] w_S(|x_l - x_j|^2) & l = k \\ -4 [2\Psi''_S(|u_k - u_l|^2) (u_k - u_l)^2 \\ \quad + \Psi'_S(|u_k - u_l|^2)] w_S(|x_k - x_l|^2) & l \neq k \end{cases} \end{aligned} \quad (6)$$

In the second derivatives δ_{lj} denotes the Kronecker symbol $\delta_{lj} = \begin{cases} 1 & l = j \\ 0 & \text{else} \end{cases}$.

It is clear that the complete derivatives then have the form

$$\frac{\partial E}{\partial u_i} = \alpha \frac{\partial E_D}{\partial u_i} + (1 - \alpha) \frac{\partial E_S}{\partial u_i} ,$$

and the corresponding sum for the second derivatives. Having these derivatives at hand we can now study the concrete minimisation algorithms.

3.1 Jacobi Method – Fixed-Point Iteration

For a critical point u of the energy functional E we have

$$\nabla E(u) = 0 \iff \frac{\partial E}{\partial u_i} = 0 \quad \text{for all } i \in \{1, \dots, n\} . \quad (7)$$

We define the abbreviations

$$\begin{aligned} d_{i,j} &:= \Psi'_D (|u_i - f_j|^2) w_D (|x_i - x_j|^2) , \\ s_{i,j} &:= \Psi'_S (|u_i - u_j|^2) w_S (|x_i - x_j|^2) \end{aligned}$$

which help us to rewrite (7) as

$$0 = \alpha \sum_{j \in \Omega} d_{i,j} (u_i - f_j) + 2(1 - \alpha) \sum_{j \in \Omega} s_{i,j} (u_i - u_j)$$

where we use the partial derivatives shown in (5) and (6). This can be transformed into fixed point form

$$u_i = \frac{\alpha \sum_{j \in \Omega} d_{i,j} f_j + 2(1 - \alpha) \sum_{j \in \Omega} s_{i,j} u_j}{\alpha \sum_{j \in \Omega} d_{i,j} + 2(1 - \alpha) \sum_{j \in \Omega} s_{i,j}} .$$

To have a positive denominator we assume that $\Psi'_{\{S,D\}}(s^2) > 0$, i. e., the penalisers are monotonically increasing. Furthermore we assume that $w_{\{S,D\}}(s^2) \geq 0$ and $w_{\{S,D\}}(0) > 0$ for the spatial weights. We use this equation to build up a first iterative method to minimise the value of E where an additional index k denotes the iteration number. Note that $d_{i,j}$ and $s_{i,j}$ also depend on the evolving image u^k and thus also get a superscript to denote the iteration level involved. The corresponding fixed point iteration then reads as

$$\begin{aligned} u_i^0 &:= f_i , \\ u_i^{k+1} &:= \frac{\alpha \sum_{j \in \Omega} d_{i,j}^k f_j + 2(1 - \alpha) \sum_{j \in \Omega} s_{i,j}^k u_j^k}{\alpha \sum_{j \in \Omega} d_{i,j}^k + 2(1 - \alpha) \sum_{j \in \Omega} s_{i,j}^k} . \end{aligned} \quad (8)$$

With our assumptions on $\Psi_{\{D,S\}}$ and $w_{\{D,S\}}$ from above we know that $d_{i,j}^k \geq 0$ and $s_{i,j}^k \geq 0$ for all i, j, k . That means in (8), u_i^{k+1} is calculated as

a convex combination of grey values of the initial image f_j and of the last iteration step u_j^k . Thus we have

$$\min_{j \in \Omega} \{u_j^k, f_j\} \leq u_i^{k+1} \leq \max_{j \in \Omega} \{u_j^k, f_j\} \quad \text{for all } i \in \Omega, k \in \mathbb{N} .$$

Induction shows that the fixed point scheme (8) satisfies a maximum-minimum principle, i.e.,

$$\min_{j \in \Omega} \{f_j\} \leq u_i^k \leq \max_{j \in \Omega} \{f_j\} \quad \text{for all } i \in \Omega, k \in \mathbb{N} .$$

Let us now consider the set $M := \{u \in \mathbb{R}^n \mid \|u\|_\infty \leq \|f\|_\infty\}$ with the norm $\|u\|_\infty := \max_{j \in \Omega} |u_j|$. $M \neq \emptyset$ is compact and convex. Writing our scheme (8) in the form $u^{k+1} = F(u^k)$ with $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$, the maximum-minimum stability implies that $F(M) \subseteq M$. With our requirements on $\Psi_{\{D,S\}}$ and $w_{\{D,S\}}$, the denominator in (8) is always larger than zero. This means that each component $F_i: \mathbb{R}^n \rightarrow \mathbb{R}$ is continuous with respect to the norm $\|\cdot\|_\infty$. Since this holds for all i , we know that $F: (\mathbb{R}^n, \|\cdot\|_\infty) \rightarrow (\mathbb{R}^n, \|\cdot\|_\infty)$ is continuous. Then Brouwer's fixed point theorem (see for example [18, page 51]) shows that F has a fixed point in M .

In the fixed point iteration scheme (8) we calculate u^{k+1} using only components of the vector u^k of the old iteration level:

$$u_i^{k+1} := F_i(u^k) \quad \text{for all } i \in \Omega, k \in \mathbb{N} . \quad (9)$$

Such a method can also be called a nonlinear Jacobi method.

3.2 Newton's Method

We search a zero of the gradient $\nabla E(u) = 0$. To this end we use Newton's method for the function ∇E :

$$u^{k+1} = u^k - H(E, u^k)^{-1} \nabla E(u^k) , \quad (10)$$

where $H(E, u^k)$ is the Hessian matrix of E at the point u^k . In each step of (10) we have to solve a linear system of equations. This system of equations can only be solved if the Hessian matrix is invertible which is the case for a convex functional E . That means we cannot use Newton's method for all penalisers shown in the last section. If both $\Psi_D(s^2)$ and $\Psi_S(s^2)$ are convex in s , i. e. $2\Psi''(s^2)s^2 + \Psi'(s^2) > 0$, the Hessian matrix $H(E, u^k)$ has positive diagonal entries and is strictly diagonally dominant. This does not only allow us to solve the linear system of equations, but it also gives us the possibility to use a whole variety of iterative solution algorithms like the Gauß-Seidel, successive overrelaxation, or conjugate gradient method. We have chosen to use the Gauß-Seidel method here to solve the linear system of equations.

A practical observation shows that the steps of Newton's method are often too long. Thus we have used a simple line-search strategy:

$$u^{k+1} = u^k - \sigma_k H(E, u^k)^{-1} \nabla E(u^k)$$

with $\sigma_k \in (0, 1]$. We try $\sigma_k = 1, \frac{1}{2}, \frac{1}{4}, \dots$ until the energy is decreasing in the step: $E(u^{k+1}) < E(u^k)$.

It is clear that one step of Newton's method is much more expensive than one fixed point iteration step. Nevertheless, numerical examples will show that the whole process can still converge faster.

3.3 Gauß-Seidel Method

Instead of the nonlinear Jacobi method (9) one can also use a nonlinear Gauß-Seidel method which involves pixels of the old and the new iteration level. For each pixel $u_i =: x^0$, we perform m steps of a local fixed point iteration

$$x^{l+1} := F_i(u_1^{k+1}, \dots, u_{i-1}^{k+1}, x^l, u_{i+1}^k, \dots, u_n^k) \quad l = 1, 2, 3, \dots$$

and set $u_i^{k+1} := x^m$ afterwards. Since these inner steps satisfy a maximum-minimum principle, the whole Gauß-Seidel method does. Thus one can apply the same reasoning as above and gets the existence of fixed points for the equation.

3.4 Gauß-Seidel Newton Method

Here we solve the single component equations with Newton's method. We start with the pixel value $x^0 = u_i^k$ of the last iteration level and set

$$x^{l+1} = x^l - \sigma_l \left(\frac{\partial^2 E}{\partial u_i^2}(\tilde{u}) \right)^{-1} \frac{\partial E}{\partial u_i}(\tilde{u})$$

with $\tilde{u} = (u_1^{k+1}, \dots, u_{i-1}^{k+1}, x^l, u_{i+1}^k, \dots, u_n^k)$. After m steps of this method we set $u_i^{k+1} = x^m$ and proceed with the next pixel. The only difference is that we use the criterion $E_{loc}(x^{l+1}) < E_{loc}(x^l)$ for the choice of the step size σ_l where the local energy is defined as

$$\begin{aligned} E_{loc}(u) = & \alpha \sum_{j \in \Omega} \Psi_D (|x^l - f_j|^2) w_D (|x_i - x_j|^2) \\ & + (1 - \alpha) \sum_{j \in \Omega} \Psi_S (|x^l - \tilde{u}_j|^2) w_S (|x_i - x_j|^2) . \end{aligned}$$

We should note that besides the number of (outer) iterations, all methods except of the Jacobi method have the number of inner iterations as an additional parameter for the numerics.

4 Numerical Experiments

Now we investigate the practical behaviour of the methods presented in the last section. We use the two stopping criteria $\|u^{k+1} - u^k\|_2 < a$ and $|E(u^{k+1}) - E(u^k)| < b$. That means we stop the algorithm if the changes of both the evolving image (in terms of the Euclidean norm) and the energy value are smaller than prescribed limits a and b . As quality measure we use the signal-to-noise-ratio $\text{SNR}(f, g) = 10 \log_{10} \left(\frac{\|g - \mu\|_2^2}{\|f - g\|_2^2} \right)$ where μ stands for the mean value of the original image g , and f is the noisy image. The results of the 1-D example are displayed in Figure 1 and Table 3. Here we have Gaussian noise, and we have chosen $\Psi_D(s^2) = s^2$, $\Psi_S(s^2) = 2(\sqrt{s^2 + \varepsilon^2} - \varepsilon)$ with $\varepsilon = 0.01$, and $w_D(s^2) = w_S(s^2) = 1.0$ inside a data term window of size 7 and a smoothness term window of size 11 with $\alpha = 0.5$. The number of inner iterations was optimised to yield a fast convergence for each method. We see that Newton's method is the fastest one in this case while all of the methods yield almost equal SNR values.

Figure 2 and Table 4 contain the results of the 2-D experiments. For the removal of salt-and-pepper noise we chose $\Psi_D(s^2) = 2(\sqrt{s^2 + \varepsilon^2} - \varepsilon)$ with $\varepsilon = 0.01$, $\Psi_S(s^2) = 2\lambda^2 \left(1 + \frac{s^2}{\lambda^2}\right)^{\frac{1}{2}}$ with $\lambda = 0.1$. We set $w_D(s^2) = w_S(s^2) = 1.0$ with both windows of size 3 and $\alpha = 0.95$. Here we have the opposite case, and the simple fixed point scheme is faster than Newton's method. We have performed some more experiments indicating that this does not depend on the dimension of the problem but on the choice of penalisers. That the convergence is much slower for Newton's method is also shown by the smaller SNR value in this example.

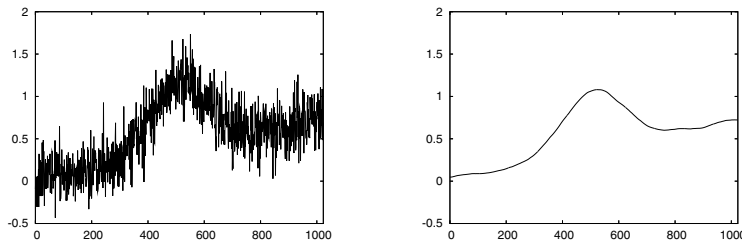


Fig. 1. Denoising experiment in 1-D. Left: Test signal with additive Gaussian noise with zero mean, size 1024 pixels, SNR 4.44. Right: Denoised version of the signal.

Table 3. Denoising experiment in 1-D with $a = 10^{-2}$ and $b = 10^{-6}$.

method	iterations	inner it.	energy	SNR	time [sec]
Fixed point	1309	–	165.70820	21.90	3.332
Newton	25	60	165.70807	21.87	0.515
Gauß-Seidel	842	1	165.70815	21.89	2.193
G.-S. Newton	683	1	165.70813	21.89	5.739

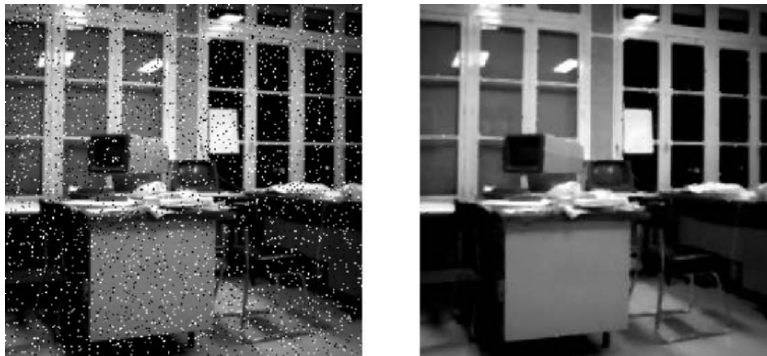


Fig. 2. Denoising experiment in 2-D. Left: Test image with salt-and-pepper noise (256×256 pixels, SNR 11.50). Right: Denoised version of the image.

Table 4. Denoising experiment in 2-D with $a = b = 10^3$.

method	iterations	inner it.	energy	SNR	time [sec]
Fixed point	38	–	$1.86 \cdot 10^7$	19.05	8.175
Newton	25	5	$2.07 \cdot 10^7$	16.18	89.239
Gauß-Seidel	3	25	$1.86 \cdot 10^7$	19.15	8.502
G.-S. Newton	6	2	$1.86 \cdot 10^7$	19.14	23.317

5 Conclusions

We have investigated four different algorithmic approaches for the variational image simplification NDS-model presented in [8]. For schemes based on fixed point iterations we have shown the existence of fixed points. Newton’s method is only applicable for a certain class of convex penalisers. We have seen with practical examples that in terms of running time we cannot prefer one single method in general. Currently we are considering the question if other numerical approaches based on multigrid ideas could help to reduce the running time especially of the fixed point approaches applicable for all weighting types.

Acknowledgement

We gratefully acknowledge partly funding within the priority programme SPP1114 of the *Deutsche Forschungsgemeinschaft (DFG)*, proj. WE 2602/2-2.

References

1. D. Barash: A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(6), 2002, 844–847.

2. P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud: Two deterministic half-quadratic regularization algorithms for computed imaging. Proc. IEEE International Conference on Image Processing **2**, (ICIP-94, Austin, Nov. 13-16, 1994), 168–172.
3. C.K. Chu, I.K. Glad, F. Godtliebsen, and J.S. Marron: Edge-preserving smoothers for image processing. Journal of the American Statistical Association **93**(442), 1998, 526–541.
4. M. Elad: On the origin of the bilateral filter and ways to improve it. IEEE Transactions on Image Processing **11**(10), 2002, 1141–1151.
5. L.D. Griffin: Mean, median and mode filtering of images. Proceedings Royal Society of London A **456**, 2000, 2995–3004.
6. J.J. Koenderink and A.L. Van Doorn: The structure of locally orderless images. International Journal of Computer Vision **31**(2/3), 1999, 159–168.
7. J.-S. Lee: Digital image smoothing and the sigma filter. Computer Vision, Graphics, and Image Processing **24**, 1983, 255–269.
8. P. Mrázek, J. Weickert, and A. Bruhn: On robust estimation and smoothing with spatial and tonal kernels. In: *Geometric Properties for Incomplete Data*, R. Klette, R. Kozera, L. Noakes, and J. Weickert (eds.), vol. 31 of *Computational Imaging and Vision*, Springer, 2005.
9. D. Mumford and J. Shah: Optimal approximation of piecewise smooth functions and associated variational problems. Communications on Pure and Applied Mathematics **42**, 1989, 577–685.
10. P. Perona and J. Malik: Scale space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. **12**, 1990, 629–639.
11. J. Polzehl and V. Spokoiny: Adaptive weights smoothing with applications to image restoration. Journal of the Royal Statistical Society, Series B **62**(2), 2000, 335–354.
12. L.I. Rudin, S. Osher, and E. Fatemi: Nonlinear total variation based noise removal algorithms. Physica D **60**, 1992, 259–268.
13. P. Saint-Marc, J.-S. Chen, and G. Medioni: Adaptive smoothing: a general tool for early vision. IEEE Trans. Pattern Anal. Mach. Intell. **13**(6), 1991, 514–529.
14. S.M. Smith and J.M. Brady: SUSAN – a new approach to low level image processing. International Journal of Computer Vision **23**(1), 1997, 43–78.
15. A.N. Tikhonov: Solution of incorrectly formulated problems and the regularization method. Soviet Mathematics Doklady **4**(2), 1963, 1035–1038.
16. C. Tomasi and R. Manduchi: Bilateral filtering for gray and colour images. In: *Proc. of the 1998 IEEE International Conference on Computer Vision*, Bombay, India, January 1998. Narosa Publishing House, 839–846.
17. G. Winkler, V. Aurich, K. Hahn, and A. Martin: Noise reduction in images: some recent edge-preserving methods. Pattern Recognition and Image Analysis **9**(4), 1999, 749–766.
18. E. Zeidler: *Nonlinear Functional Analysis and Applications I: Fixed-Point Theorems*. Springer, New York, 1986.

Multiscale Voice Morphing Using Radial Basis Function Analysis

Christina Orphanidou^{1,2}, Irene M. Moroz¹, and Stephen J. Roberts²

¹ Oxford Centre for Industrial and Applied Mathematics, University of Oxford,
Oxford OX1 3LB, UK, {orphanid,moroz}@maths.ox.ac.uk

² Pattern Analysis and Machine Learning Research Group, University of Oxford,
Oxford OX1 3PJ, UK, sjrob@robots.ox.ac.uk

Summary. A new multiscale voice morphing algorithm using radial basis function (RBF) analysis is presented in this paper. The approach copes well with small training sets of high dimension, which is a problem often encountered in voice morphing. The aim of this algorithm is to transform one person's speech pattern so that it is perceived as if it was spoken by another speaker. The voice morphing system we propose assumes parallel training data from source and target speakers and uses the theory of wavelets in order to extract speaker feature information. The spectral conversion is modelled using RBF analysis. Independent listener tests demonstrate effective transformation of the perceived speaker identity.

1 Introduction

Voice morphing technology enables a user to transform one person's speech pattern into another person's speech pattern with distinct characteristics, giving it a new identity, while preserving the original content. It transforms *how* something is said without changing *what* is said. The applications of such a technology are numerous such as text-to-speech adaptation where the voice morphing system can be trained on relatively small amounts of data and allows new voices to be created at a much lower cost than the currently existing systems. The voice morphing system can also be used in situations when the speaker is not available and previous recordings have to be used. Other applications can be found in broadcasting, voice editing, karaoke applications, internet voice applications as well as computer and video games. Voice morphing is performed in two steps. In the training stage, acoustic parameters of the speech signals uttered by both the source and target speakers are computed and appropriate rules mapping the acoustic space of the source speaker into that of the target speaker are obtained. In the transformation stage, the acoustic features of the source signal are transformed using the mapping rules such that the synthesized speech sounds like the target speaker. In order to

build a successful voice morphing system two issues need to be addressed. Firstly, a successful mathematical representation of the speech signal must be obtained that represents the speech signal so that the synthetic speech can be regenerated and the accents and pauses can be manipulated without artifacts. In this representation factors such as identifying and extracting the key features of speaker identity are of primary importance. Voice morphing can then be achieved by modifying these features. Secondly, the type of conversion function and the method of training and application must be decided.

2 Description of the System

2.1 Overview of existing methods

There has been a considerable amount of research directed at the problem of voice transformation [2, 3, 6, 10, 11, 13, 17, 20], using the general approach described above.

The first approaches were based around linear predictive coding (LPC) [14]. This approach was improved up by using residual-excited LPC (RELPC), where the residual error was measured and used to produce the excitation signal [2, 3, 17]. Most authors developed methods based on either the interpolation of speech parameters and modelling the speech signals using formant frequencies [1], Linear Prediction Coding (LPC) cepstrum coefficients [8], Line Spectral Frequencies (LSFs) [12], and harmonic-plus-noise model parameters [20] or based on mixed time- and frequency- domain methods to alter the pitch, duration, and spectral features. These methods are forms of single-scale morphing.

Although the above methods provide good approximation to the source-filter model of the human vocal tract and they encode good quality speech at a low bit rate they face two problems: artifacts are introduced at boundaries between successive speech frames and there is absence of the detailed information during the extraction of formant coefficients and the excitation signal. These result in the limitation on accurate estimation of parameters and distortion caused during synthesis of target speech. In addition to this, previously, the unvoiced phonemes were often left untouched and directly passed to the output thereby keeping the source speaker's consonants. In other studies, the voiced/unvoiced phonemes were not separated thus causing some audible artifacts. One of the main reasons is that it is difficult for single-scale methods like LPC to extract the voice characteristics from a complex speech signal which mixes many different high-frequency components.

There have been a number of different approaches to the problem of determining the mapping of parameters from the source speech to the target speech. Arslan and Talkin [2, 3] proposed a system in which the speech of both speakers is marked up automatically into phonemes. Then, the Line

Spectral Frequencies for each frame of each utterance are calculated and labeled with the relevant phoneme. Following this, the centroid vector for each phoneme is calculated, and a one-to-one mapping from source to target codebooks is established. This process is also performed on the residual signal. The transformation may then be carried out by the use of *codebook mapping*. However, the quality suffered due to the fact that the converted signal was limited to a discrete set of phonemes.

Stylianou et al. [17] suggested improvements to the method of Arslan and Talkin through the use of Gaussian mixture models of the speaker's spectral parameters. They time-aligned the source and target speech, performed initial clustering (grouping according to a specific attribute) of the speech, followed by the use of Gaussian mixtures to learn the mapping for each class of speech segments. Each class is characterized by its mean together with the characteristic spread around the center of the class. In order to establish the parameters of the mixture model, they used the expectation-maximisation (EM) algorithm. This method led to less unnatural discontinuities within the synthesized speech. Kain [11] proposed a solution where he mapped the spectral envelope in the same manner to [17], but then predicted the residual from the predicted spectral envelope. This resulted in fewer artifacts than existing systems, but was restricted to speech where the speakers were speaking in a monotone, and where the speakers were asked to mimic the timing of another speaker [10]. Orphanidou et al. [16] proposed using the Generative Topographic Mapping, a non-linear, parametric, latent variable Gaussian mixtures model in order to transform the speaker's spectral parameters as modelled by the LPC coefficients. Although the non-linear model proved successful in learning and mapping the speech characteristics by generating speech recognized as the target speaker's, it suffered by losing some high-frequency components as well as distortion during speech synthesis.

2.2 Proposed Model

The lack of detail in the morphed speech produced by the existing methods leads to the conclusion that a multi-scale voice morphing method should be tested that performs the conversion in different levels of analysis (subbands) and captures in more detail the range of frequencies of the speech signals. Our proposed model uses the theory of Wavelets as a means of extracting the speech features followed by the Radial Basis Function Neural Networks (RBFNN) for modelling the spectral conversion. The identification of such conversion functions is based upon a procedure which *learns* the shape of the conversion from a few target spectra from a data set [6].

The theory of wavelets has developed rapidly over the past few years and has been successfully applied in many areas of physics, engineering, sciences, statistics and applied mathematics, forming a versatile tool for representing general functions and data sets. Wavelets have been used in speech analysis [4, 7] and image morphing but applications to voice morphing are almost

untouched. Only [19] is found, which introduced the Discrete Wavelet Transform and got some encouraging results.

The radial basis conversion functions introduced here are characterized by a perceptually-based fast training procedure, desirable interpolation properties and computational efficiency.

Figure 1 depicts a diagrammatic representation of our proposed model. Source and target training data is time-aligned, normalized and then analyzed as follows by wavelets: The wavelet coefficients are calculated in several levels of detail. At each level, the wavelet coefficients are normalized and a mapping is learned using the RBFNN model. Test data from the source speaker are normalized and decomposed to the same number of levels as the training data and the wavelet coefficients are projected through the calculated network in order to produce the morphed coefficients. The morphed coefficients are then used in order to reconstruct the target speaker's speech signal.

3 Wavelet Analysis

Wavelet decomposition is done using the Wavelet Toolbox in MATLAB [15]. In order to reduce the dimension of the problem the wavelet coefficients at the two highest frequency levels are set to zero. The best basis is chosen by minimizing the normalized mean-square error, or *reconstruction* error, given by:

$$\text{NMSE} = \text{E}^{\text{REC}} = \sqrt{\frac{\sum_{x=1}^N (y(x) - y^*(x))^2}{\sum_{n=1}^N (y(x)^2)}}$$

after the two sets of wavelet coefficients are set to zero. Here N is the number of points in the sample, $x = 1, \dots, N$ is the index of each point, $y(x)$ is the original signal and $y^*(x)$ is the reconstructed signal. The mean-square error of the reconstructed signal and the original one is divided by the norm of the original signal so that a more objective indication of the error can be obtained. The Coiflet 5 and Biorthogonal 6.8 basis minimized the reconstruction error for the male and female speakers, respectively, and were therefore used.

The wavelet coefficients calculated at each level of decomposition, thus, form the *feature vectors*, \mathbf{x} , to be used as input data in the network training process.

4 Radial Basis Functions and Network Training

The basic form of the RBFNN mapping is

$$y_k(\mathbf{x}) = \sum_{j=1}^M w_{kj} \phi_j(\mathbf{x}) + w_{k0},$$

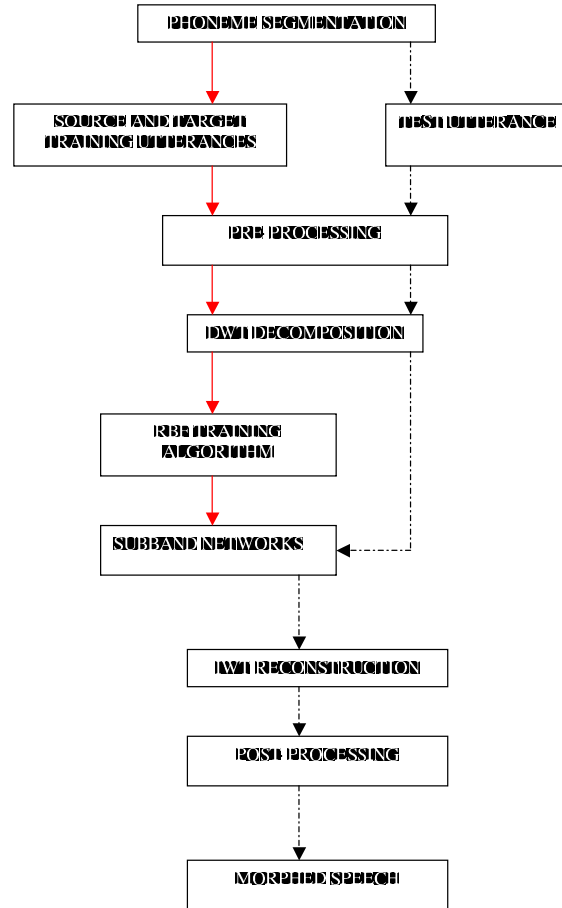


Fig. 1. Proposed Model

where w_{k0} is the bias term which can be absorbed into the summation by including an extra basis function ϕ_0 whose activation is set to 1. For the case of Gaussian basis functions we have:

$$\phi_j(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mu_j\|^2}{2\sigma_j^2}\right).$$

Here \mathbf{x} is the d -dimensional input vector with elements x_i and μ_j is the vector determining the centre of basis function ϕ_j and has elements μ_{ji} . This Gaussian radial basis functions can be generalized to allow for arbitrary covariance matrices Σ_j ³. The basis function is, therefore, taken to have the form

³ Given n sets of variates denoted $\{X_1\}, \dots, \{X_n\}$, the first order covariance matrix is defined by $V_{ij} = \text{cov}(x_i, x_j) \equiv \langle (x_i - \mu_i)(x_j - \mu_j) \rangle$, where μ_i is the mean.

$$\phi_j(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_j)^T \Sigma_j^{-1}(\mathbf{x} - \mu_j)\right).$$

4.1 Learning a Radial Basis Function Network

The RBFNN is considered a *2-layered* network, because the learning process is done in two different stages, referred to as *layers* [5]. A key aspect is the distinction between the first and second layers of weights. In the first stage, the input data set \mathbf{x}^n alone is used to determine the parameters of the basis functions, the first-layer weights. As only the input data is used, the training method is called *unsupervised*. The first layer weights are then kept fixed while the second layer weights are found in the second phase. The second stage is supervised as both input and target data is required. Optimization is done by a classic least squares approach. Considering the RBFNN mapping we defined in Subsection 2.2 (and absorbing the bias parameter into the weights) we now have

$$y_k(\mathbf{x}) = \sum_{j=0}^M w_{kj} \phi_j(\mathbf{x})$$

where ϕ_0 is an extra “basis function” with activation value fixed at 1. Writing this in matrix notation

$$\mathbf{y}(\mathbf{x}) = \mathbf{W}\phi,$$

where $\mathbf{W} = (w_{kj})$ and $\phi = (\phi_j)$. The weights can now be optimized by minimization of a suitable error function, e.g. the sum-of-squares error function

$$E = \frac{1}{2} \sum_n \sum_k \{y_k(\mathbf{x}^n) - t_k^n\}^2$$

where t_k^n is the target value for output unit k when the network is presented with the input vector \mathbf{x}^n . The weights are then determined by the linear equations [5]

$$\Phi^T \Phi \mathbf{W}^T = \Phi^T \mathbf{T},$$

where $(\mathbf{T})_{nk} = t_k^n$ and $(\Phi)_{nj} = \phi_j(\mathbf{x}^n)$. This can be solved by

$$\mathbf{W}^T = \Phi^\dagger \mathbf{T}$$

where the notation Φ^\dagger denotes the *pseudo-inverse* of Φ . Thus, the second-layer weights can be found by fast, linear matrix inversion techniques [5].

5 Voice Conversion

Our voice morphing algorithm is implemented using the following steps:

1. Source and target speech signals are chosen for two people uttering the same sentence/word/phoneme. The signals are split into the *training*, *validation* and *test* data sets.
2. The raw source and target training samples are time-aligned i.e. resampled so that they have the same length.
3. The training and test samples are normalized in order to have 0 mean and 1 standard deviation. As a result, both samples now have the same length and statistics.
4. The training and test samples are divided into frames and 5-level wavelet decomposition is performed to each frame. 6 sets of wavelet coefficients (approximation at level 5 and detail at levels 5,4,3,2 and 1) are obtained for each frame of each sample.
5. The level 1 and level 2 detail coefficients are set to zero.
6. The wavelet coefficients at the four remaining levels are normalized.
7. For each level of decomposition, a radial basis function network is initialised and trained using the source and target training sample wavelet coefficients. 3-fold cross-validation is used (using the training and validation samples) and the best network is obtained (i.e. the one that gives the smallest validation error).
8. At each level, the source speaker's test samples' coefficients are projected through the corresponding network and the transformed coefficients are obtained.
9. The transformed coefficients are un-normalized with respect to the target speaker coefficients' original statistics so that it has the mean and standard deviation of the target speaker's speech samples.
10. The transformed coefficients are used in order to reconstruct the signal.
11. The reconstructed signal is un-normalised with respect to the target speaker training sample's statistics.
12. The transformed signal is tested and compared to the target signal to assess the transformation.

6 Results and Evaluation

The system was tested using data from the TIMIT database [9]. In order to evaluate the performance of our system in terms of its perceptual effects an *ABX-style* preference test was performed, which is common practice for voice morphing evaluation tests [2, 12, 18]. Independent listeners were asked to judge whether an utterance X sounded closer to utterance A or B in terms of speaker identity, where X was the converted speech and A and B were the source and target speech, respectively. Note that the *ABX-style* test we perform here is a variation of the standard ABX test as the sound X is not actually spoken by either speaker A or B, it is a new sound and the listeners need to identify which of the two sounds it sounds *like*. Also, utterances A and B were presented to the listeners in random order. In total, 12 utterances were

tested which consisted of 3 male-to-male, 3 female-to-female, 3 female-to-male and 3 male-to-female source-target combinations. All utterances were taken from the TIMIT database. 13 independent listeners took part in the testing. Each listener was presented with the 12 different triads of sounds (source, target and converted speech, the first two in random order) and had only one chance of deciding whether sound X sounds like A or B. Table 1 shows the % success of the test.

Table 1. Results of listener tests

Source-Target	%
Male-to-Male	84.6
Female-to-Female	79.5
Male-to-Female	89.7
Female-to-Male	92.3

7 Conclusion

In this study, we have proposed a new multi-scale method for voice morphing which uses the theory of wavelets and radial basis function neural networks. Listening tests were performed to demonstrate the performance of the system. The obtained conversion effect is satisfying as transformed signals can be recognized as of the target speaker although a muffling effect is observed. Future developments of the voice morphing method introduced in this paper will include its evaluation with other wavelet bases, examining thresholding methods in order to decrease the number of coefficients required as well as training the conversion network with larger databases.

References

1. M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara: Voice conversion through vector quantization. *IEEE Proceedings of the IEEE ICASSP*, 1998, 565–568.
2. L. Arslan: Speaker transformation algorithm using segmental codebooks (stasc). *Speech Communication* **28**, 1999, 211–226.
3. L. Arslan and D. Talkin: Voice conversion by codebook mapping of line spectral frequencies and excitation spectrum. *Proc. EUROSPEECH*, 1997, 1347–1350.
4. A.K. Barros, T. Rutkowski, F. Itakura, and N. Ohnishi: Estimation of speech embedded in a reverberant and noisy environment by independent component analysis and wavelets. *IEEE Transactions on Neural Networks* **13**(4), 2002, 808–893.
5. C.M. Bishop: *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1997.

6. C. Drioli: Radial basis function networks for conversion of sound speech spectra. Proceedings of the 2nd COST G-6 Workshop on Digital Audio Effects, 1999.
7. E. Ercelesi: Second generation wavelet transform-based pitch period estimation and voiced/unvoiced decision for speech signals. *Applied Acoustics* **64**, 2003, 25–41.
8. S. Furui: Research on individuality features in speech waves and automatic speaker recognition techniques. *Speech Communication* **5**, 1986, 183–197.
9. J.S. Garofolo, L.F. Lamel, W.M. Fisher, J.G. Fiscus, D.S. Pallet, and N.L. Dahlgren: *DARPA TIMIT-Acoustic Phonetic Continuous Speech Corpus*. US Department of Commerce, 1993.
10. B. Gillett: Voice transformation: making new voices for speech synthesis. Proc. Postgraduate Conference 2002, Theoretical and Applied Linguistics, University of Edinburgh, 2002.
11. K.A. Gillow: *High Resolution Voice Transformation*. Ph.D. thesis, Oregon Health and Science University, Portland, Oregon, 2001.
12. A. Kain and M.W. Macon: Spectral voice conversion for text-to-speech synthesis. Proc. ICASSP'98 **1**, 1998, 285–288.
13. M.W. Macon: *Speech Synthesis Based on Sinusoidal Modeling*. Ph.D. thesis, Oregon Graduate Institute Center for Spoken Language Understanding, 1996.
14. J. Makhoul: Linear prediction – a tutorial review. Proc. IEEE **63**, 1975, 561–580.
15. M. Misiti, G. Oppenheim, and J.M. Poggi: *Wavelet toolbox 2.2*. The Mathworks, 2002.
16. C. Orphanidou, I.M. Moroz, and S.J. Roberts: Voice morphing using the generative topographic mapping. Proceedings of the International Conference on Computer, Communication and Control Technologies **1**, 2003, 222–225.
17. Y. Stylianou, O. Cappe, and E. Moulines: Statistical methods for voice quality transformation. Proc. EUROSPEECH, 1995, 447–450.
18. Y. Stylianou, O. Cappe, and E. Moulines: Continuous probabilistic transform for voice conversion. IEEE Trans. on Speech and Audio Processing **6**(2), 1998, 131–142.
19. O. Turk and L.M. Arslan: Subband based voice conversion Proceedings ICSLP, 2002.
20. H. Valbret, E. Moulines, and J.P. Tubach: Voice transformation using psola technique. *Speech Communication* **11**, 1992, 175–187.

Associating Families of Curves Using Feature Extraction and Cluster Analysis

Jane L. Terry, Andrew Crampton, and Chris J. Talbot

School of Computing and Engineering, University of Huddersfield, Huddersfield
HD1 3DH, UK, {j.l.terry,a.crampton,c.j.talbot}@hud.ac.uk

Summary. The focus of this paper is to provide a reliable approach for associating families of curves from within a large number of curves. The method developed assumes that it is not known how many families are present, or how many curves are held within a family. The algorithm described has been developed for use on acoustical data, where there is a strong physical relationship between related curves. In the solution to the problem each of the curves have several key features which are measured and parametrised. This results in the characteristics of each curve being described by a small number of directly comparable parameters. Using these parameters it is then possible to find the related curves by applying cluster analysis to the feature space.

1 Introduction

This paper introduces a new method of associating families of curves that share a strong physical relationship. In the method described there are a large number of data sets, each of which can be represented by a curve. Within these data sets there is an unknown number of families present, each with an unknown number of curves.

Acoustical data, specifically data recorded using a single omni-directional passive sonar sensor, was used in the development of the algorithm. The recorded sound wave is separated into its frequency components using Fast Fourier Transforms over a series of short time intervals, so that the data is now represented in the time-frequency domain. Each of the time values in this space represents the output of a single FFT, where the amplitude of each of the frequencies is also present. The curves analysed in this paper are the paths of high amplitude frequencies over time.

In this application a family of curves represents all the sound waves emitted from a single noise source, collectively these waves form a harmonic set; a single noise source will emit a sound wave at a fundamental frequency, whilst also at integer multiples of this frequency, this is what is known as a harmonic set.

In the solution to the problem outlined, each of the curves have several key features that are measured and parametrised. This results in the characteristics of each curve being described by a small number of directly comparable parameters. It is shown that the results of the association can be greatly improved, with the different families being more distinct, if the data is pre-processed before the features are measured.

Having created a feature set to describe the curves, cluster analysis is used to separate the different curves into groups. Whilst cluster analysis is able to distribute the curves into groups, it is not an unsupervised method of finding the optimal distribution of the data. To achieve this a ratio is applied to each of the possible distributions found during the clustering to find the optimal grouping. The ratio applied here is one developed by Calinski and Harabasz [1].

An overview of the algorithmic procedure is shown in Figure 1. Each of the individual stages of the processing chain are described in the following sections.



Fig. 1. Diagrammatic overview of algorithmic procedure.

2 Feature Extraction

Any number of features can be measured for each of the curves, however in the examples shown in this paper only three are used in order to enable the visualisation of the results.

The choice of features that can be used in this application is limited only by the necessity of the features being represented by a small number of parameters. For example, whilst the derivative of a curve may yield some useful properties, the fact that it produces a time series makes it unusable in this algorithm. Other features, such as the mean of a distribution, which are defined by a single parameter, are acceptable for inclusion in the analysis. Currently no feature selection algorithm has been employed, so the choice of features is made manually.

Many of the features that have been considered for application in this problem originate from surface texture analysis [4], where the surface profiles measured by a stylus are analysed. Other features that have been implemented are standard statistical parameters, such as the variance of a distribution.

Once the parameters have been measured they are stored in an $(n \times p)$ feature matrix, where there are n data sets representing curves in the data and p measured parameters. From this point the algorithm is now operating

in feature space, i.e., each of the curves being analysed is now represented by a single point in p -dimensional space.

An example of the transformation from time-frequency space to feature space is shown in Figure 2. In this example feature space is three dimensional, with the dimensions representing quadrature, average roughness and frequency range. The definitions of these features are given in (1). $\mathbf{t} = [t_1, t_2, \dots, t_N]^T$ represents the time updates of the data, where each value of t represents a row from the Lofargram and f is the corresponding frequency value. The parameters used here are

$$\begin{aligned} \text{quadrature: } q &= \sum_{i=1}^{N-1} \frac{|f(\mathbf{t}_{i+1})| - |f(\mathbf{t}_i)|}{\mathbf{t}_{i+1} - \mathbf{t}_i}, \\ \text{average roughness: } r_a &= \frac{1}{N} \sum_{i=1}^N |f_i|, \\ \text{frequency range: } f_r &= \max(f(\mathbf{t})) - \min(f(\mathbf{t})). \end{aligned} \tag{1}$$

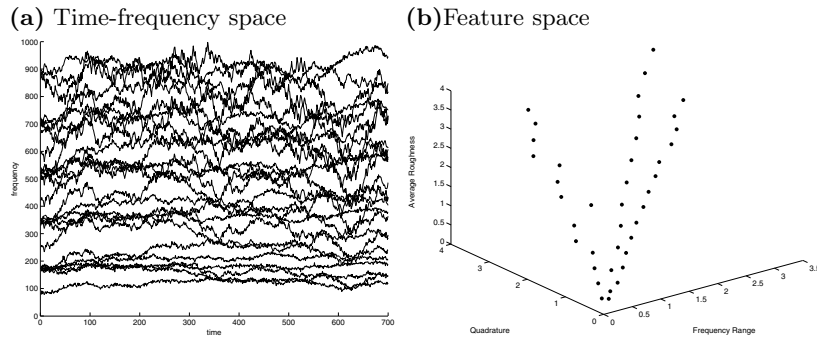


Fig. 2. Example data set shown in both time-frequency and feature space.

3 Normalisation

Before the feature extraction occurs, the data needs to be pre-processed to get all of the curves into the same frame of reference where they are directly comparable. It can be seen from Figure 2b that the data points are not naturally clustered and that the range of each of the different features, or dimensions, are not the same. These problems can be rectified by pre-processing the data, before measuring the features. The pre-processing that occurs is to apply a standard normalisation technique to the data. The normalisation technique applied to the k th curve is defined as

$$\hat{f}_k(\mathbf{t}) = \frac{f_k(\mathbf{t}) - \mu_k}{\sigma_k},$$

where \mathbf{t} is a vector of the time updates and $k = 1, 2, \dots, n$ where there are n curves. The normalised frequency $\hat{f}_k(\mathbf{t})$ is found by evaluating the mean and standard deviation (μ_k and σ_k respectively) of the frequency distributions.

In terms of measuring features and being able to cluster the curves into their respective families, this normalisation has an additional advantage. The curves being analysed in this paper represent acoustical data, with each family of curves denoting all sound emanating from a single noise source, meaning that they are harmonically related. Consequently, there is a strong physical relationship between associated curves. This relationship is described as

$$\frac{a}{b}f_b(\mathbf{t}) = f_a(\mathbf{t}), \quad (2)$$

where a and b are integer scaling parameters (representing the harmonic number of the curve, where the fundamental frequency of a harmonic set is represented by 1) and f is the frequency of the curve, this result is valid over all time, t .

Using the result in (2) and assuming zero error in the data, the mean value of curve a can be represented as

$$\mu_a = \frac{a}{b}\mu_b, \quad (3)$$

and the standard deviation can be represented as

$$\sigma_a = \frac{a}{b}\sigma_b. \quad (4)$$

Using results (3) and (4) it is clear to see that the normalised curve a , \hat{f}_a is

$$\hat{f}_a = \frac{\frac{a}{b}f_b - \frac{a}{b}\mu_b}{\frac{a}{b}\sigma_b} = \hat{f}_b,$$

which means that the normalised curves that are related will now be approximately identical. The feature space for the normalised curves can be seen in Figure 3 to clearly cluster the curves into distinct clusters.

4 Standardisation

An essential part of the operation of the clustering algorithm, which analyses the feature matrix, is to measure the distance between pairs of points. The decision of which points to associate is made from the magnitude of this distance.

It is visually clear that the feature parameters that have been measured can separate into distinct clusters. However, for this separation to also be detected by the clustering algorithm the scale of the parameters is important.

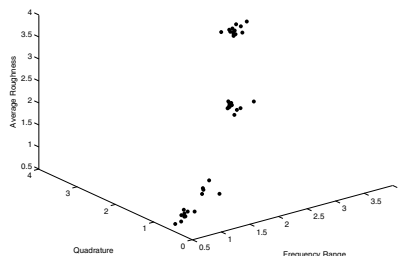


Fig. 3. Normalised frequency space

If one feature has a range of 1000 across all points and another has a range of 0.001 then it is clear that when measuring the proximity between points the result will be heavily dominated by one of the features, effectively reducing the dimensionality of the problem. The purpose of standardisation is to transform the data in such a way so that the relative distance between points is unaffected in a single dimension, but the scales across the dimensions are comparable in magnitude.

Many different standardisation techniques have been suggested in previous works [6], [3], [2]. Whilst the most used method of standardisation is to reduce the distribution of the data to unit variance, this method is often called autoscaling. It has been shown in [6], [3] that there are other standardisation methods that are more effective in most clustering applications. In particular in [6] it is shown that dividing the distribution by the sample range outperforms other standardisation techniques. Consequently, it is this method of standardisation that has been chosen for use in this application.

The measured feature parameters are held in a $(n \times p)$ feature matrix \mathbf{S} , with p features for n observations, or curves. So that

$$\mathbf{S} = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1p} \\ s_{21} & s_{22} & \cdots & s_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{np} \end{bmatrix}.$$

The standardised result for the feature matrix is found using

$$\hat{\mathbf{s}}_i = \left\{ \frac{\mathbf{s}_i}{\max(\mathbf{s}_i) - \min(\mathbf{s}_i)} \right\}_{i=1}^p,$$

where \mathbf{s}_i is obtained from the feature matrix \mathbf{S} , so that $\mathbf{s}_i = [s_{1i}, s_{2i}, \dots, s_{ni}]^T$.

This technique is applied to each of the columns of the feature matrix so that each of the dimensions, or features, are standardised independently of each other, giving the standardised feature matrix

$$\hat{\mathbf{S}} = [\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2, \dots, \hat{\mathbf{s}}_p]. \quad (5)$$

5 Clustering

Hierarchical clustering is a technique of separating a large data set into smaller groups that better describe the data. An overview of the process can be outlined as

1. Each data set is assigned to a unique cluster.
2. Merge two clusters.
3. Continue merging two clusters until all points are held within a single cluster.

The input for the clustering algorithm is the standardised feature matrix $\hat{\mathbf{S}}$, given in (5).

5.1 Forming Clusters

At any level in the hierarchy there are g clusters, where $g = n$ in the first level, and decreases by 1 in every subsequent level until $g = 1$. A cluster, G_k has m_k data points, in p dimensional space, within it.

$$\left\{ \mathbf{G}_k \right\}_{k=1}^g = \left\{ \hat{\mathbf{s}}_j \right\}_{j=\mathbf{z}_1}^{\mathbf{z}_{m_k}}$$

where $\hat{\mathbf{s}}_j = \{\hat{s}_{j1}, \hat{s}_{j2}, \dots, \hat{s}_{jp}\}$ and \mathbf{z} is a vector containing the indexing values of the data points held within the cluster. All the data from the feature matrix corresponding to a cluster is contained within the $(m_k \times p)$ matrix \mathbf{G}_k .

The centre, or centroid, \mathbf{c} of each cluster can be found by finding the average position of each of the data points in the cluster so that the centroid of the k th cluster is given by

$$\mathbf{c}_k = \frac{1}{m_k} \sum_{j=\mathbf{z}_1}^{\mathbf{z}_{m_k}} \hat{\mathbf{s}}_j.$$

At the next level of the algorithm two of these clusters are fused. The decision as to which two are fused is taken using a proximity measure, the two clusters that will optimise this measure are fused.

In this algorithm the two groups that are fused are the pair that minimise the squared Euclidean distance between cluster centres,

$$\min_{i,j} \left[(\mathbf{c}_i - \mathbf{c}_j)^T (\mathbf{c}_i - \mathbf{c}_j) \right]$$

for $i, j = 1, 2, \dots, g$ and $i \neq j$.

5.2 Choosing the Optimal Level

Having applied a hierarchical clustering algorithm to the data we have a choice of n levels in which to separate the data into clusters. A decision now needs to be taken to choose which level in the hierarchy best describes the data.

There are many measures and techniques that have been suggested to find the best distribution of the data. The method that has been implemented in this application is the Calinski-Harabasz measure [1] which is found to be the most effective measure in [5].

For matrices B and W , representing the between and within group proximities respectively, the choice as to which level is the optimal separation of the data into clusters is taken by measuring the ratio

$$R = \frac{\text{trace}(B)}{(k-1)} \bigg/ \frac{\text{trace}(W)}{(n-k)},$$

at each level of the hierarchy. The optimal level will be the one that maximises this ratio.

Effectively this is simply a ratio between the between group sum of squares, $\text{trace}(B)$, and the within group sum of squares, $\text{trace}(W)$, where n is the number of data points in the distribution, and k is the number of clusters used to describe the data.

It is not necessary to calculate the matrices B and W since the trace of these matrices can be expressed using

$$\text{trace}(B) = \frac{1}{2} \left((k-1)\bar{d}^2 + (n-k)A \right)$$

and

$$\text{trace}(W) = \frac{1}{2} \left((n_1-1)\bar{d}_1^2 + (n_2-1)\bar{d}_2^2 + \dots + (n_k-1)\bar{d}_k^2 \right).$$

Where

$$A = \frac{1}{(n-k)} \sum_{i=1}^k (n_i-1)(\bar{d}^2 - \bar{d}_i^2).$$

The values \bar{d} and \bar{d}_n are found from evaluating the matrix D^2 where

$$D_{i,j}^2 = (\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j)$$

for $i, j = 1, 2, \dots, n$ and $i \neq j$. The value \bar{d}^2 is then found by

$$\bar{d}^2 = \frac{2}{n(n-1)} \sum_{i=2}^n \sum_{j=1}^{i-1} D_{i,j}^2.$$

The values of \bar{d}_m^2 are found in exactly the same way, but only using the values of \mathbf{x} that are in the cluster m .

Therefore, the Calinski-Harabasz ratio can be re-written as

$$R = \frac{\bar{d}^2 + \frac{(n-k)}{(k-1)}A}{\bar{d}^2 - A}.$$

The value of R is calculated for the output at each level of the hierarchical algorithm and the level that maximises the ratio is found. This is chosen as the optimal separation of the data into clusters.

6 Results

In order to test the validity of the algorithm in giving the optimal clustering, it is necessary to use data where this choice is known. For this reason synthetic data sets were generated, for a Monte-Carlo analysis of the algorithm. It is assumed that the optimal clustering is the harmonically related curves. Each data set has either 2,3,5,8 or 10 harmonic sets in.

The algorithm works on curves which are present over the whole length of the data set, i.e., there are no breaks in the data. However, the final application of this algorithm is in the analysis of acoustical data recorded in the ocean. In this environment it is likely that there will be periods of time where the signal cannot be distinguished from the noise, resulting in broken curves. For this application the algorithm will need to be developed to operate in real time on a continuous data stream. Whilst no effort has been made to solve the problem of broken curves, the algorithm has been tested over different time periods in order to determine how the efficiency of the algorithm is effected for shorter periods. With shorter curves the number of features is obviously less, so the distinction between the different families is expected to be less obvious. Another reason why testing on shorter time periods is required is that it is important that the results of any analysis get to the operator with the smallest possible delay after the signal is recorded.

A potential problem with using hierarchical clustering is the number of computations that are required for an increasing number of curves. However, in this application the number of curves that are likely to be found in the data will always be small. It is expected that the number of curves present in any data set will not exceed 40 and it is anticipated that the actual number will be significantly smaller than this. As this number is small it is not considered necessary to consider the complexity of the problem for this application, however the authors are currently researching complexity issues for other applications.

Table 1 shows how the algorithm performed on a large number of test data sets. It can be seen that whilst the accuracy does decrease with the length of the sample, the accuracy over the shorter length is still fairly high, with a sample length of 20 time updates still having an accuracy rate of 70%.

The majority of the incorrect associations gave a number of harmonic sets that was either one above, or below, the actual number present. It is likely that

these errors could have been caused by two families of curves having similar values for one or more of the features, resulting in the distribution over the feature space being indistinct. This type of inaccuracy can be improved by the choice of features used. In these results only frequency range, quadrature and average roughness were considered. Other features may be more appropriate for some of the data sets. It may also be advantageous to try looking at a larger number of features in the analysis.

Another, more likely cause for the errors could be that some of the families have very few curves in them. The algorithm is likely, in this case, to see these points as outliers from one of the larger clusters.

Table 1. Results showing accuracy of algorithm over varying time lengths.

Number of Time Updates	Number of Data Sets	Deviation From Number of Clusters Expected							Percentage of Correct Outcomes
		< -3	-2	-1	0	+1	+2	> +2	
700	60				57	2		1	95
500	60			2	55	1	1	1	92
200	60			1	56	3			93
100	60	1	1	2	56				93
50	60	1	1	7	49		2		82
20	60	1	1	10	42	4	2		70

7 Conclusions and Further Development

The results in Table 1 show that the algorithm performs well within the conditions tested. The fact that the algorithm maintained an accuracy rate of 70% for the shortest sample length tested, and that the accuracy of the results only fell by 2% between data with a sample length of 700 time updates and sets with 100 updates, suggests that the algorithm will be a useful tool for an operator.

Throughout this paper only one method of clustering the data and choosing the optimum number of clusters have been discussed. It may be possible that the accuracy of the algorithm can be improved by using a different clustering technique, or proximity measure within the clustering process [2].

The algorithm is still in development, but this initial investigation has proved that clustering does work. However, there are still many limitations to this approach.

One of the main problems is that the input frequency tracks must currently be continuous over the whole sample length. In reality this is not practical, since the tracks are often broken due to changes in the signal to noise ratio and other effects.

Another problem is that data will continue to be recorded simultaneously to the analysis being performed, meaning that it is not possible to work with the entire data in a single pass of the algorithm. This is a real-time problem that needs to be considered in further development of the algorithm.

Methods of overcoming these problems are being considered. Solutions suggested have included developing a windowing technique, resulting in only a small time sample being evaluated, reducing the delay in producing results for the operator. Confidence in the results can then be improved as time continues, and a larger number of windows have been evaluated. In overcoming the problem of the frequency tracks not being present over the entire sample range, it will be necessary to develop some features, or parameters, that are invariant over time.

The authors believe that the algorithm described in this paper is a new method of finding families of curves. Other methods that find families of curves use template matching, [8] and [7]. The clustering algorithm described compares well with these methods over continuous data, but has the disadvantage of currently not working over broken curves.

Acknowledgement

This research has been jointly funded by the EPSRC and Thales Underwater Systems UK. The authors would like to thank Thales Underwater Systems UK, in particular Roger Benton, for their technical expertise and time given to this project.

References

1. T. Calinski and J. Harabasz: A dendrite method for cluster analysis. *Communications in Statistics* **3**, 1975, 1–27.
2. S.B. Everitt, S. Landau, and M. Leese: *Cluster Analysis*. 4th edition, Arnold Publishers, London, 2001.
3. R. Gnanadesikan, J.R. Kettenring, and S.L. Tsao: Weighting and selection of variables for cluster analysis. *Journal of Classification* **12**, 1995, 113–136.
4. R. Leach: *Measurement Good Practice Guide no. 37: The Measurement of Surface Texture Using Stylus Instruments*. Technical report, National Physical Laboratory, Teddington, UK, 2001.
5. G.W. Milligan and M.C. Cooper: An examination of procedures for determining the number of clusters in a data set. *Psychometrika* **50**, 1985, 159–179.
6. G.W. Milligan and M.C. Cooper: A study of standardisation of variable in cluster analysis. *Journal of Classification* **5**, 1988, 181–204.
7. J.L. Terry and J.C. Mason: Template matching and data fitting in passive sonar harmonic detection. In: *Sonar Signal Processing*. Institute of Acoustics, 2004.
8. V. Van Leijen: A robust family finder algorithm. In: *Underwater Defence Technology*, 2002.

Part II

Numerical Simulation

Particle Flow Simulation by Using Polyharmonic Splines

Armin Iske

Department of Mathematics, University of Hamburg, D-20146 Hamburg, Germany,
iske@math.uni-hamburg.de

Summary. This contribution reports on novel concepts of adaptive particle methods for flow simulation, where scattered data reconstruction by polyharmonic splines plays a key role. Our discussion includes the construction of both Lagrangian and Eulerian particle methods, where two different prototypes are being presented: one semi-Lagrangian particle method (SLPM) and one finite volume particle method (FVPM). It is shown how polyharmonic spline reconstruction can be used in the resampling of the particle models. To this end, basic features of polyharmonic splines are first reviewed, before important aspects concerning their numerical stability and approximation behaviour are discussed. Selected practical aspects concerning the efficient implementation of the resulting numerical algorithms are addressed. Finally, the good performance of the presented particle methods is demonstrated by using two different test case scenarios from real-world applications.

1 Introduction

The numerical simulation of multiscale phenomena in time-dependent evolution processes is of great importance in many relevant applications from science and technology, which, moreover, incorporates many challenging issues concerning the design of suitable computational methods. Efficient, robust and accurate computer simulations require customized multiscale approximation algorithms, where adaptivity plays a key role.

Particle models have provided very flexible discretization schemes for the numerical simulation of multiscale phenomena in various relevant applications from computational science and engineering. In the modelling of time-dependent evolution processes, for instance, particle models are particularly well-suited to cope with rapid variation of domain geometries and anisotropic large-scale deformations.

Moreover, particle models are popular concepts in *meshfree* methods for partial differential equations [14, 15], where mesh-independent modelling concepts are essentially required to reduce the computational complexity of the

utilized numerical algorithms. Indeed, *meshfree* particle methods [35] are currently subject to lively research activities, where several different types of particle-based methods were developed very recently.

To briefly explain one of their basic features, particle models usually work with a finite set of particles, where some specific physical properties or shape functions are attached to each of the individual particles. Moreover, in the simulation of time-dependent evolution processes, the finite particles are usually subject to adaptive modifications during the simulation. The diverse zoo of particle methods includes the following species, to mention but a few.

- Smoothed particle hydrodynamics (SPH) [42];
- Reproducing kernel particle method (RKPM) [29, 36];
- Generalized finite element method (GFEM) [4, 41];
- Particle-partition of unity methods (PPUM) [4, 16, 17, 18, 19, 20, 21, 41];
- Finite mass method (FMM) [13];
- Finite volume particle method (FVPM) [23];
- Finite pointset method (FPM) [33, 49];
- Moving point methods [12];
- Semi-Lagrangian method (SLM) [46, 48];
- Method of characteristics [6, 8, 28];
- Particle methods for the Boltzmann equation [43].

This contribution is not meant to be a comprehensive and systematic exposition of particle methods, but it rather surveys very recent developments of the author and co-authors, where some of the relevant material is detailed through our previous papers [7, 8, 26, 27, 28, 31]. Unlike related papers on the subject, the present article is more focussed on various important aspects concerning the numerical stability and local approximation behaviour of selected multiscale particle methods, where *polyharmonic splines* play a key role.

To be more precise, in the relevant multiscale modelling of time-dependent evolution processes, a finite set of moving particles are utilized, where the particles are subject to dynamic modifications during the simulation. This requires both customized adaption rules for the adaptive modification of the active particle set, and a suitable strategy for the resampling of the particle values. This in turn requires a suitable scheme for local scattered data reconstruction. To this end, we prefer to work with polyharmonic splines, which were recently shown to provide numerically stable reconstructions of arbitrary local approximation order [25] from *Lagrange data*.

In this article, we generalize some of our previous results in [25] to scattered data reconstruction from *Hermite-Birkhoff data*. This problem includes both reconstruction of particle point values and particle average values, which are required in the presented *Eulerian* and *Lagrangian* particle-based simulation methods.

The outline of this article is as follows. In the following Section 2, we briefly review some basic facts concerning hyperbolic conservation laws, being the governing equations for the flow simulation model problems that we

wish to address. This then leads us to two different particle-based discretizations, the *semi-Lagrangian particle method* (SLPM) for passive advection and the Eulerian *finite volume particle method* (FVPM) [23] for nonlinear hyperbolic conservation problems. As shown in Section 3, where both SLPM and FVPM are introduced, either of these fundamentally different discretization schemes relies on scattered data reconstruction. In Section 4 we show how polyharmonic splines can be used to provide a numerically stable reconstruction of arbitrary local approximation order. To this end, new results concerning invariance properties of the reconstruction methods' Lebesgue functions are proven. Finally, numerical examples arising from two real-world test case scenarios are presented, one concerning tracer advection over the arctic stratosphere, Section 5, the other concerning oil reservoir modelling, Section 6.

2 Hyperbolic Problems

Multiscale flow simulation requires suitable approximation algorithms for the numerical solution of time-dependent *hyperbolic conservation laws*

$$\frac{\partial u}{\partial t} + \nabla f(u) = 0, \quad (1)$$

where for some domain $\Omega \subset \mathbb{R}^d$, $d \geq 1$, and a compact time interval $I = [0, T]$, $T > 0$, the solution $u : I \times \Omega \rightarrow \mathbb{R}$ of (1) is sought.

In this problem, $f(u) = (f_1(u), \dots, f_d(u))^T$ denotes a given *flux tensor*, and it is usually assumed that *initial conditions*

$$u(0, x) = u_0(x), \quad \text{for } x \in \Omega, \quad (2)$$

at time $t = 0$ are given.

One special case for (1), (2) is *passive advection*, where the flux f is linear, i.e.,

$$f(u) = \mathbf{v} \cdot u,$$

in which case (1) becomes

$$\frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u = 0, \quad (3)$$

provided that the given *velocity field*

$$\mathbf{v} \equiv \mathbf{v}(t, x) = (v_1(t, x), \dots, v_d(t, x))^T \in \mathbb{R}^d, \quad t \in I, x \in \Omega,$$

is *divergence-free*, i.e.,

$$\operatorname{div} \mathbf{v} = \sum_{j=1}^d \frac{\partial v_j}{\partial x_j} \equiv 0.$$

However, in the general case of (1) the flux function f is, unlike in (3), *nonlinear*. Note that the nonlinear case is much more complicated than the linear one of passive advection. Indeed, in contrast to the linear case, a nonlinear flux function f usually leads to *discontinuities* in the solution u , *shocks*, as observed in many relevant applications, such as fluid flow and gas dynamics. Such discontinuities of the solution u in (1) can easily develop spontaneously even from smooth initial data u_0 in (2).

Therefore, the nonlinear flow simulation requires more sophisticated mathematical and computational methods to numerically solve the Cauchy problem (1), (2). For a comprehensive introduction to numerical methods for hyperbolic problems we recommend the textbook [34].

3 Basic Lagrangian and Eulerian Particle Methods

This section briefly reviews two conceptually different particle-based algorithms for the numerical solution of the hyperbolic problem (1),(2). One basic concept for passive advection is given by the *semi-Lagrangian particle method* (SLPM), to be discussed in Subsection 3.1. The other is the Eulerian *finite volume particle method* (FVPM) [23], leading to a conservative discretization method for (nonlinear) hyperbolic problems. Both concepts, SLPM [6, 8, 7] and FVPM [27], are treated in greater detail in our previous work [6, 8, 7, 27, 28]. Therefore, we prefer to restrict ourselves here to a discussion on the very basic features of the two methods, and so we keep the presentation in this section rather short.

3.1 Semi-Lagrangian Particle Method (SLPM)

Starting point for our proposed particle method SLPM is the Lagrangian form

$$\frac{du}{dt}(t, x(t)) = 0, \quad (4)$$

of the linear equation (3), where $\frac{du}{dt} = \frac{\partial u}{\partial t} + \nabla f(u)$ is the *material derivative*. The discretization of (4) is done w.r.t. time, so that for any time step $t \rightarrow t + \tau$, $\tau > 0$, the resulting semi-Lagrangian particle method (SLPM) [46, 48] has the form

$$\frac{u(t + \tau, \xi) - u(t, \Phi^{t, t+\tau} \xi)}{\tau} = 0, \quad (5)$$

where $\xi \in \Omega$ denotes a particle position at time $t + \tau$, and $\Phi^{t, t+\tau} \xi \in \Omega$ denotes the corresponding *upstream point* of the particle at time t . In the physical interpretation of the particle model, the upstream point $\Phi^{t, t+\tau} \xi$ of ξ is the unique position of a flow particle at time t , whose position at time $t + \tau$ is ξ .

Note that the one-to-one correspondence between $\Phi^{t, t+\tau} \xi$ and ξ can be described by the initial value problem

$$\dot{x} = \frac{dx}{dt} = \mathbf{v}(t, x), \quad x(t + \tau) = \xi, \quad (6)$$

whose unique solution $x(t)$ is determined by the *continuous evolution* $\Phi^{t, t+\tau} : \Omega \rightarrow \Omega$ of the *ordinary differential equation* (ODE) in (6), which explains the notation $\Phi^{t, t+\tau}\xi$ for the upstream point in (5).

The SLPM in [6] works with a finite set $\Xi = \{\xi\}_{\xi \in \Xi}$ of nodes (particle points), where each node ξ corresponds at a time $t \in I$ to one flow particle. In each advection step of SLPM and for each node ξ , an approximation $\Psi^{t, t+\tau}\xi$ to the upstream point $\Phi^{t, t+\tau}\xi$ is first computed, before the required value $u(t + \tau, \xi)$ of the solution is determined by local interpolation. In this concept, $\Psi^{t, t+\tau} : \Omega \rightarrow \Omega$ is referred to as *discrete evolution* of the ODE in (6), where $\Psi^{t, t+\tau}$ is given by a specific numerical algorithm for the initial value problem (6), and so $\Phi^{t, t+\tau} \approx \Psi^{t, t+\tau}$. For details concerning the construction of $\Psi^{t, t+\tau}$ in SLPM, we refer to [6].

The following algorithm reflects the basic advection step of SLPM.

Algorithm 1 Semi-Lagrangian Particle Method (SLPM).

INPUT: Time step $\tau > 0$, nodes Ξ , values $\{u(t, \xi)\}_{\xi \in \Xi}$ at time t .

FOR each $\xi \in \Xi$ **DO**

- (a) Compute upstream point approximation $\Psi^{t, t+\tau}\xi$;
- (b) Determine set $\mathcal{N}_\xi \subset \Xi$ of neighbouring nodes around $\Psi^{t, t+\tau}\xi$;
- (c) Determine value $u(t, \Psi^{t, t+\tau}\xi)$ by local interpolation from data $\{u(t, \nu)\}_{\nu \in \mathcal{N}_\xi}$;
- (d) Advect by letting $u(t + \tau, \xi) = u(t, \Psi^{t, t+\tau}\xi)$.

OUTPUT: Values $\{u(t + \tau, \xi)\}_{\xi \in \Xi}$ at time $t + \tau$.

3.2 Finite Volume Particle Method (FVPM)

To briefly explain the main ingredients of the utilized finite volume particle method (FVPM), we denote for any $\xi \in \Xi$ by $V_\xi \subset \Omega$ the *influence area* of a particle at node ξ . The particle influence areas may, for instance, be given by the Voronoi tiles

$$V_\xi = \left\{ x \in \Omega : \|x - \xi\| = \min_{\nu \in \Xi} \|x - \nu\| \right\} \subset \Omega, \quad \text{for } \xi \in \Xi,$$

of the Voronoi diagram $\mathcal{V}_\Xi = \{V_\xi\}_{\xi \in \Xi}$ for Ξ , in which case \mathcal{V}_Ξ yields by

$$\Omega = \bigcup_{\xi \in \Xi} V_\xi \quad (7)$$

a decomposition of Ω into subdomains $V_\xi \subset \Omega$ with pairwise disjoint interior.

Note that the Voronoi diagram \mathcal{V}_ξ is entirely determined by the geometry of the nodes Ξ . We remark that there are efficient algorithms from computational geometry [45] for the construction and maintenance of the Voronoi diagram

\mathcal{V}_Ξ and its dual Delaunay tessellation. Therefore, the combination between Voronoi diagrams and finite volumes yields through FVPM a very efficient and flexible particle method for the numerical solution of (1),(2). We further remark that the general concept of FVPM [23, 30], allows for overlapping influence areas $\{V_\xi\}_{\xi \in \Xi}$ satisfying (7), in which case, however, FVPM needs to be combined with a partition of unity method (PUM). This provides more flexibility, but it leads to a more complicated FVPM discretization. For more details, we refer to [30].

Now, for any particle located at $\xi \in \Xi$ at time t , its *particle average* is defined by

$$\bar{u}_\xi(t) = \frac{1}{|V_\xi|} \int_{V_\xi} u(t, x) dx, \quad \text{for } \xi \in \Xi \text{ and } t \in I.$$

According to the classical concept of FV [34], for each $\xi \in \Xi$ the average value $\bar{u}_\xi(t)$ is, at time step $t \rightarrow t + \tau$, updated by an explicit numerical method of the form

$$\bar{u}_\xi(t + \tau) = \bar{u}_\xi(t) - \frac{\tau}{|V_\xi|} \sum_\nu F_{\xi, \nu}, \quad (8)$$

where $F_{\xi, \nu}$ denotes the *numerical flux* between particle ξ and a neighbouring particle $\nu \in \Xi \setminus \xi$. The required exchange of information between neighbouring particles is modelled via a generic numerical flux function, which may be implemented by using any suitable FV flux evaluation scheme, such as ADER in [32]. For the sake of brevity, we prefer to omit details concerning the construction of the numerical flux, but refer to the ideas in [32] instead.

The following algorithm reflects the basic time step of FVPM.

Algorithm 2 Finite Volume Particle Method (FVPM).

INPUT: Time step $\tau > 0$, nodes Ξ , particle averages $\{\bar{u}_\xi(t)\}_{\xi \in \Xi}$ at time t .

FOR each $\xi \in \Xi$ **DO**

- (a) Determine set $\mathcal{N}_\xi \subset \Xi \setminus \xi$ of neighbouring nodes around ξ ;
- (b) Compute numerical flux $F_{\xi, \nu}$ for each $\nu \in \mathcal{N}_\xi$;
- (c) Update particle average \bar{u}_ξ for ξ by (8).

OUTPUT: Particle averages $\{\bar{u}_\xi(t + \tau)\}_{\xi \in \Xi}$ at time $t + \tau$.

3.3 WENO Reconstruction

Modern approaches of finite volume discretizations are usually combined with *essentially non-oscillatory* (ENO) [22], or *weighted essentially non-oscillatory* (WENO) [37] reconstruction schemes to obtain conservative, high order numerical methods for hyperbolic conservation laws (1).

To explain how FVPM can be combined with ENO and WENO reconstruction, let us view the influence area V_ξ of any node $\xi \in \Xi$ as the *control volume* of ξ , where the control volume V_ξ is uniquely represented by ξ .

Now the basic idea of ENO schemes is to first select, for each node $\xi \in \Xi$ a small set $\mathcal{S} = \{\mathcal{S}_i\}_{i=1}^k$ of k stencils, where each stencil $\mathcal{S}_i \subset \Xi$ is given by a set of nodes in the neighbourhood of ξ . Then, for each stencil \mathcal{S}_i , $1 \leq i \leq k$, a reconstruction $s_i \equiv s_{\mathcal{S}_i}$ is computed, which interpolates given particle averages $\bar{u}_i \equiv \bar{u}_{\mathcal{S}_i}(t)$ over the control volumes $\{V_\nu\}_{\nu \in \mathcal{S}_i}$ in the stencil \mathcal{S}_i , i.e.,

$$\bar{s}_i(\nu) = \bar{u}_i(\nu), \quad \text{for all } \nu \in \mathcal{S}_i. \quad (9)$$

Among the k different reconstructions s_i , $1 \leq i \leq k$, of the k different stencils, the *smoothest* (i.e. least oscillatory) reconstruction is selected, which constitutes the numerical solution over the control volume V_ξ . The selection of the smoothest s_i among the k reconstructions is done by using a suitable *oscillation indicator* \mathcal{I} to avoid spurious oscillations of the reconstruction.

In the more sophisticated WENO reconstruction, the whole stencil set $\mathcal{S} = \{\mathcal{S}_i\}_{i=1}^k$ is used in order to construct, for a corresponding control volume V_ξ , a *weighted* sum of the form

$$s(x) = \sum_{i=1}^k \omega_i s_i(x), \quad \text{with } \sum_{i=1}^k \omega_i = 1,$$

where the weights $\omega_i = \tilde{\omega}_i / \sum_{j=1}^k \tilde{\omega}_j$, with $\tilde{\omega}_i = (\epsilon + \mathcal{I}(s_i))^{-\rho}$ for $\epsilon, \rho > 0$, are determined by using the aforementioned oscillation indicator \mathcal{I} .

We remark that WENO schemes show, in comparison with ENO schemes, superior convergence to steady-state solutions and higher order accuracy, especially in smooth regions and around extrema of the solution.

Commonly used ENO/WENO schemes work with polynomial reconstruction, which, however, may lead to severe numerical instabilities, especially when the particles are heterogeneously distributed, see [1]. In the following Section 4 we show how to construct a numerically stable reconstruction scheme of arbitrary high order. The utilized reconstruction relies on a variational formulation, which also provides a very natural choice for the required oscillation indicator \mathcal{I} , see Subsection 4.3.

4 Reconstruction by Polyharmonic Splines

Note that either of the proposed particle methods, SLPM and FVPM, relies on local scattered data reconstruction. Indeed, SLPM relies on local *Lagrange interpolation*, where the interpolation problem in step (c) of Algorithm 1 can for $\mathcal{N} \equiv \mathcal{N}_\xi$ and $u(\nu) \equiv u(t, \nu)$ be stated as $s_{\mathcal{N}} = u_{\mathcal{N}}$, i.e.,

$$s(\nu) = u(\nu), \quad \text{for all } \nu \in \mathcal{N}. \quad (10)$$

As regards FVPM, the required WENO reconstruction (9) can for any stencil $\mathcal{N} \subset \mathcal{S}$ and with using $\bar{u}(\nu) \equiv \bar{u}(t, \nu)$ be rewritten as

$$\bar{s}(\nu) = \bar{u}(\nu), \quad \text{for all } \nu \in \mathcal{N}. \quad (11)$$

Note that either reconstruction problem, (10) or (11), requires a suitable method for (local) scattered data reconstruction. To this end, we prefer to work with *polyharmonic splines*, which are powerful methods for scattered data interpolation from multivariate scattered data.

In this section, we show how polyharmonic splines can be used to solve the more general Hermite-Birkhoff reconstruction problem, where the Lagrange interpolation (10) and the particle average reconstruction (11) are only special cases. This yields a unified approach for local scattered data reconstruction by polyharmonic splines.

The discussion in this section first recalls some basic features of polyharmonic spline reconstruction, before recent results concerning the numerical stability and local approximation order of Lagrange interpolation are generalized to Hermite-Birkhoff reconstruction. A short discussion concerning optimality properties of the reconstruction method concludes this section.

4.1 Lagrange Interpolation

Polyharmonic splines, due to Duchon [11], are traditional tools for Lagrange interpolation from multivariate scattered data. According to the polyharmonic spline interpolation scheme, the interpolant s in (10) is of the form

$$s(x) = \sum_{\nu \in \mathcal{N}} c_\nu \phi_{d,m}(\|x - \nu\|) + p(x), \quad p \in \mathcal{P}_m^d, \quad (12)$$

where $\|\cdot\|$ denotes the Euclidean norm on \mathbb{R}^d , and where \mathcal{P}_m^d is the linear space of all d -variate real-valued polynomials of degree at most m . Note that $Q = \binom{m+d}{d}$ is the dimension of \mathcal{P}_m^d . The choice of m in (12) depends on the *order* m of the *polyharmonic spline function*

$$\phi_{d,m}(r) = \left\{ \begin{array}{ll} r^{2m-d} \log(r) & \text{for } d \text{ even,} \\ r^{2m-d} & \text{for } d \text{ odd,} \end{array} \right\} \quad \text{for } 2m > d. \quad (13)$$

4.2 Generalized Hermite-Birkhoff Interpolation

In order to generalize Lagrange interpolation by polyharmonic splines to the more general problem of Hermite-Birkhoff interpolation, let $\Lambda = \{\lambda\}_{\lambda \in \Lambda}$ denote a finite set of linearly independent linear functionals w.r.t. some function space $\mathcal{F} \equiv \mathcal{F}(\mathbb{R}^d)$ containing \mathcal{P}_m^d and $\phi_{m,d}$, so that $u_\Lambda = (\lambda(u))_{\lambda \in \Lambda}$ yields a data vector whose individual entries $\lambda(u)$ are given by action of the dual functional $\lambda \in \mathcal{F}'$ on $u \in \mathcal{F}$. Note that in case of plain Lagrange interpolation of the previous subsection, we have $\lambda_\nu(u) = u(\nu)$, so that $\lambda_\nu = \delta_\nu$ is the Dirac point evaluation functional at some point $\nu \in \Omega$, where we assume $\delta_\nu \in \mathcal{F}'$.

In the general setting of Hermite-Birkhoff interpolation, $\lambda \in \Lambda$ may also be given by point evaluation of a derivative, e.g. $\lambda(u) = D^\alpha u(x)|_{x=\nu}$, for some $\alpha \in \mathbb{N}_0^d$ and $\nu \in \Omega$, or by an *average value*,

$$\lambda(u) = \frac{1}{V} \int_V u(x) dx,$$

of u over some *control volume* $V \subset \Omega$, or by a combination of all. In the following discussion of this section, we restrict ourselves to point evaluations and (particle) averages for λ , in which case the dual functional λ is of order zero.

In short hand notation, the Hermite-Birkhoff reconstruction problem can be stated as $u_\Lambda = s_\Lambda$, i.e.,

$$\lambda(u) = \lambda(s), \quad \text{for all } \lambda \in \Lambda, \quad (14)$$

with assuming

$$s(x) = \sum_{\lambda \in \Lambda} c_\lambda \lambda^y \phi_{m,d}(\|x - y\|) + p(x), \quad p \in \mathcal{P}_m^d, \quad (15)$$

for the form of the reconstruction s in (14), where λ^y in (15) denotes the action of λ on variable $y \in \mathbb{R}^d$.

According to [24], the general Hermite-Birkhoff reconstruction problem $u_\Lambda = s_\Lambda$ can be solved under constraints

$$\sum_{\lambda \in \Lambda} c_\lambda \lambda(p) = 0, \quad \text{for all } p \in \mathcal{P}_m^d, \quad (16)$$

where the solution s is unique, provided that Λ is unisolvent w.r.t. the polynomials \mathcal{P}_m^d , i.e., for $p \in \mathcal{P}_m^d$ we have

$$\lambda(p) = 0 \text{ for all } \lambda \in \Lambda \quad \implies \quad p \equiv 0. \quad (17)$$

We remark that (17) requires that any polynomial $p \in \mathcal{P}_m^d$ can uniquely be reconstructed from its data vector p_Λ . Note that the uniqueness condition (17) is rather weak. We shall from now assume that Λ satisfies (17), so that for any reconstruction problem (14) there is a unique polyharmonic spline reconstruction of the form (15).

4.3 Optimal Recovery

According to Duchon [11], scattered data interpolation by polyharmonic splines is *optimal* in the *Beppo Levi space*

$$\text{BL}^m(\mathbb{R}^d) = \{u : D^\alpha u \in L^2(\mathbb{R}^d) \text{ for all } |\alpha| = m\},$$

being equipped with the semi-norm

$$|u|_{\text{BL}^m}^2 = \sum_{|\alpha|=m} \binom{m}{\alpha} \|D^\alpha u\|_{L^2(\mathbb{R}^d)}^2,$$

so that s in (10) minimizes the Beppo Levi energy $|\cdot|_{\text{BL}^m}$ among all recovery functions u in $\text{BL}^m(\mathbb{R}^d)$, i.e.,

$$|s|_{\text{BL}^m} \leq |u|_{\text{BL}^m}, \quad \text{for all } u \in \text{BL}^m(\mathbb{R}^d) \text{ with } u_{\mathcal{N}} = s_{\mathcal{N}}.$$

We remark that the variational formulation of Duchon's approach has been generalized to *conditionally positive definite functions* in the seminal papers [38, 39, 40] of Madych & Nelson. According to the Madych-Nelson theory, polyharmonic splines are also optimal recovery functions for the reconstruction problem (11) w.r.t. $\text{BL}^m(\mathbb{R}^d)$. In particular,

$$|s|_{\text{BL}^m} \leq |u|_{\text{BL}^m}, \quad \text{for all } u \in \text{BL}^m(\mathbb{R}^d) \text{ with } \bar{u}_{\mathcal{N}} = \bar{s}_{\mathcal{N}},$$

so that the Beppo Levi energy $|\cdot|_{\text{BL}^m}$ is a natural choice for the oscillation indicator \mathcal{I} required in the WENO reconstruction of Subsection 3.3. Therefore, we let $\mathcal{I}(u) = |u|_{\text{BL}^m}$ for the oscillation indicator in the construction of the utilized WENO scheme, see Subsection 3.3.

4.4 Scale-Invariance of the Lebesgue Constant

The *Lebesgue function* $\mathcal{L}(x)$ of the polyharmonic spline reconstruction scheme is defined as

$$\mathcal{L}(x) = \sum_{\lambda \in \Lambda} |\ell_\lambda(x)|, \quad \text{for } x \in \Omega, \quad (18)$$

and, moreover,

$$\mathcal{L} = \max_{x \in \Omega} \mathcal{L}(x)$$

is referred to as the *Lebesgue constant* of the reconstruction on $\Omega \subset \mathbb{R}^d$.

Here, $\{\ell_\lambda\}_{\lambda \in \Lambda}$ in (18) are the *Lagrange basis functions* of the reconstruction problem (14) satisfying

$$\mu(\ell_\lambda) = \delta_{\mu,\lambda} = \begin{cases} 1 & \text{for } \mu = \lambda, \\ 0 & \text{for } \mu \neq \lambda, \end{cases} \quad \text{for } \mu \in \Lambda.$$

Note that due to the uniqueness of the reconstruction, the Lagrange functions are unique. This immediately gives the following generalization of our previous result in [25].

Theorem 1. *The Lagrange basis functions $\{\ell_\lambda\}_{\lambda \in \Lambda}$ are invariant under uniform scalings.*

Proof. Following [25], it is easy to see that the reconstruction space

$$\mathcal{R} = \left\{ s = \sum_{\lambda \in \Lambda} c_\lambda \ell_\lambda^y(\|\cdot - y\|) : \sum_{\lambda \in \Lambda} c_\lambda \lambda(p) = 0 \text{ for all } p \in \mathcal{P}_m^d \right\} \subset \mathcal{F}$$

containing all possible polyharmonic spline reconstructions (15) is invariant under uniform scalings, i.e., for any $h > 0$ we find $\mathcal{R}^h = \mathcal{R}$, where

$$\mathcal{R}^h = \{ \sigma_h(s) : s \in \mathcal{R} \}$$

denotes the scaled reconstruction space, and where σ_h is the dilatation operator, being given by $\sigma_h(s) = s(\cdot/h)$.

Given uniqueness of the Lagrange functions in either space, \mathcal{R} or \mathcal{R}^h , this implies

$$\sigma_h(\ell_\lambda(x)) = \ell_\lambda(x/h) = \ell_\lambda^h(x),$$

where $\{\ell_\lambda^h\}_{\lambda \in \Lambda}$ denotes the Lagrange basis in \mathcal{R}^h . \square

Note that the above theorem immediately implies that the Lebesgue function $\mathcal{L}(x)$, and thus the Lebesgue constant \mathcal{L} , is invariant under uniform scalings. Since the polyharmonic spline reconstruction scheme is also invariant under translations and rotations, this yields the following result.

Corollary 1. *The Lebesgue constant \mathcal{L} of polyharmonic spline reconstruction is invariant under translations, rotations, and uniform scalings.* \square

We remark that the result of Corollary 1 has important consequences for the numerical stability and the approximation behaviour of local polyharmonic spline reconstruction. A comprehensive discussion on this important issue will be provided in a forthcoming paper.

For the purposes of this contribution it is sufficient to say that, due to Corollary 1, the condition number of the polyharmonic spline reconstruction problem (11) is invariant under translations, rotations, and uniform scalings.

This observation allows us to construct a simple preconditioner for stable evaluation of the polyharmonic spline reconstruction s in (11). Moreover, due to the scale-invariance of the Lebesgue constant \mathcal{L} , it can be shown that polyharmonic spline reconstruction has, when using $\phi_{d,m}$ in (11) *local approximation order* $p = m$. For details on this, we refer to our previous paper [25], where corresponding results for local Lagrange interpolation are proven.

5 Tracer Transportation over the Arctic Stratosphere

The proposed advection method SLPM has been applied to a tracer transport problem in the arctic stratosphere. In this section, we briefly explain a typical test case scenario. For further details concerning the chosen test case, we refer to our previous paper [7] and to the work by Behrens [5].

When investigating ozone depletion over the arctic, one interesting question is whether air masses with low ozone concentration are advected into southern regions. In our simplified advection model, realistic wind fields are considered, leading to fine filamentation of the tracer cloud, which complies with corresponding phenomena in previous airborne observations [9].

Wind data were taken from the high-resolution regional climate model (HIRHAM) [10]. HIRHAM resolves the arctic region with a horizontal resolution of 0.5° . It is forced at the lateral and lower boundaries by ECMWF reanalysis data. We consider the transport of a passive tracer at 73.4 hPa in the vortex. This corresponds to an altitude of 18 km. The wind field reproduces the situation in January 1990. Because stratospheric motion is thought to be constrained largely within horizontal layers, we use a two-dimensional horizontal transport scheme here. Wind data represent vector fields in the corresponding planar layer of the three-dimensional HIRHAM model. The wind field and the initial tracer distribution for the advection experiment are shown in Figure 1.

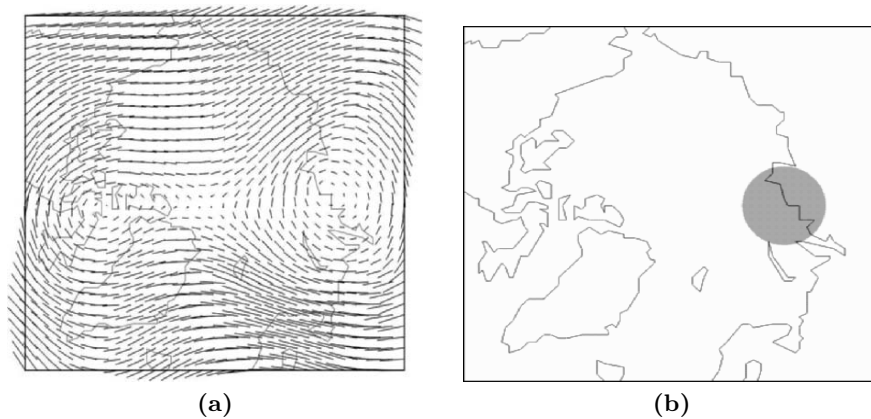


Fig. 1. (a) Wind field and initial situation for tracer advection. The artificial tracer cloud is positioned in the center of the polar vortex. (b) Continental outlines are given for orientation (Greenland in the lower left part).

A snapshot of our resulting simulation is shown in Figure 2. For a more comprehensive comparison with a comparable finite element method we refer to [7]. Note that our simulation achieves to capture the features of the tracer fairly well with a very accurate reproduction of the filamentation. The corresponding node distribution is also shown in Figure 2 (b). Note that the adaptive refinement and coarsening of the nodes essentially leads to a heterogeneous node distribution [7]. This captures finer details of the tracer quite effectively at reasonable computational costs.

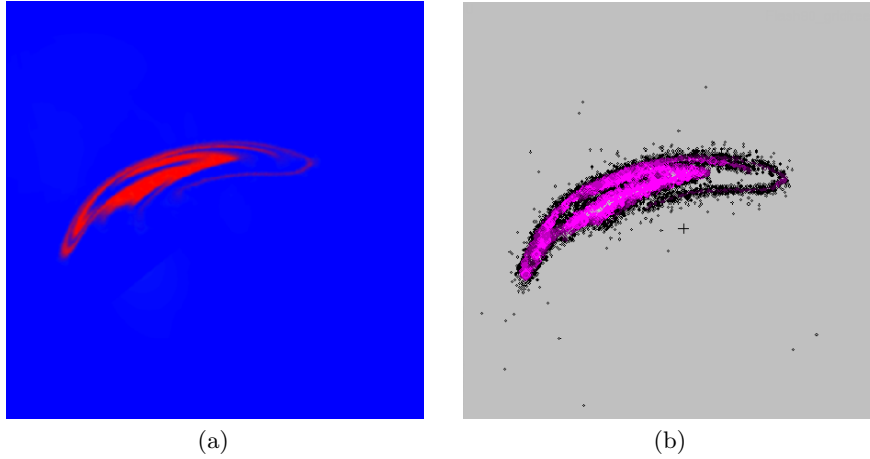


Fig. 2. (a) Result from our particle method SLPM for the stratospheric transport problem. The snapshots show the situation after 15 days of model time. Fine filaments can be observed the simulations. The corresponding node distribution is shown in (b).

6 Oil Reservoir Simulation: The Five-Spot Problem

In order to illustrate the good performance of our finite volume particle method (FVPM), we consider using one popular test case scenario from hydrocarbon reservoir modelling, termed the *five-spot problem*, where our method has been shown to be competitive with two leading commercial reservoir simulators, ECLIPSE and FrontSim of Schlumberger. For a comprehensive comparison between our related particle simulators with ECLIPSE and FrontSim, we refer to our previous papers [28, 31]. In this section, we merely show some selected numerical results concerning our particle-based simulator, being based on FVPM.

6.1 The Five-Spot Problem

The following variant of the five-spot problem in two dimensions, $d = 2$, may be summarized as follows. The computational domain $\Omega = [-0.5, 0.5]^2$ is corresponding to a bounded reservoir, where we assume, for the sake of simplicity, unit permeability of a *homogeneous* porous medium.

Initially, the pores of the reservoir are saturated with non-wetting fluid (oil), before wetting fluid (water) is injected through one injection well, being placed at the center $\mathbf{o} = (0, 0)$ of Ω . During the simulation, the non-wetting fluid (oil) is displaced by the wetting fluid (water) towards the four corner points

$$\mathcal{C} = \{(-0.5, -0.5), (-0.5, 0.5), (0.5, -0.5), (0.5, 0.5)\}$$

of the square domain Ω .

The five-spot problem requires solving the following set of three coupled equations: the *Buckley-Leverett equation*

$$\frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla f(u) = 0, \quad (19)$$

with fractional flow function

$$f(u) = \frac{u^2}{u^2 + \mu(1-u)^2}, \quad (20)$$

$\mu = \mu_w/\mu_o$ being the ratio of the two fluids' viscosities, μ_w (water) and μ_o (oil), together with the *incompressibility relation*

$$\nabla \cdot \mathbf{v}(t, x) = 0, \quad (21)$$

and *Darcy's law*

$$\mathbf{v}(t, x) = -M(u) \nabla p(t, x), \quad (22)$$

describes the flow of two immiscible incompressible fluids, water and oil, through a porous *homogeneous* medium, in the absence of capillary pressure and gravitational effects (see also [3, 44, 47]).

The solution u of (19),(21),(22) is the *saturation* of the wetting fluid (water). Hence, the value $u(t, x)$ is, at a time t and at a point x , the fraction of available volume (in the pores of the medium) filled with water, and so $u = 1$ means pure water, and $u = 0$ means pure oil.

We consider solving the above equation system (19),(21),(22) on Ω , in combination with the initial condition

$$u_0(x) = \begin{cases} 1 & \text{for } \|x - \mathbf{o}\| \leq R, \\ 0 & \text{otherwise,} \end{cases} \quad (23)$$

where we let $R = 0.02$ for the radius of the injection well at the center $\mathbf{o} \in \Omega$.

But our aim is to merely solve the Cauchy problem (19),(23) for the Buckley-Leverett equation. This is because we wish to evaluate the performance of our simulator as an adaptive saturation solver on unstructured particle sets. Therefore, we decided to work with the following simplifications of the five-spot model problem.

Firstly, following along the lines of Albright [2], we assume unit mobility, $M \equiv 1$. Secondly, we work with a *stationary* pressure field, $p(x) \equiv p(\cdot, x)$, given by

$$p(x) = \sum_{\mathbf{c} \in \mathcal{C}} \log(\|x - \mathbf{c}\|) - \log(\|x - \mathbf{o}\|), \quad \text{for all } x \in \Omega, t \in I, \quad (24)$$

which yields the *stationary* velocity field

$$\mathbf{v} = -\nabla \cdot p, \quad (25)$$

due to Darcy's law (22), and with the assumption $M \equiv 1$. It is easy to see that the velocity field \mathbf{v} is in this case divergence-free, i.e., \mathbf{v} in (25) satisfies the incompressibility relation (21). Figure 3 shows the contour lines of the pressure field p together with the streamlines of the velocity field \mathbf{v} , resulting from Darcy's law (22).

Note that by these two simplifications, the elliptic equations (21),(22) uncouple from the Buckley-Leverett equation (19). This allows us to neglect the pressure equation (22), so that we restrict ourselves to solving the flow equation (19). The taken simplifications are quite reasonable, as further supported by numerical comparisons in [28, 31] with two commercial reservoir simulators, ECLIPSE and FrontSim, each of which solves the coupled set of equations (19),(21),(22).

6.2 Adaptive Particle Flow Simulation

We apply our adaptive particle method to the Cauchy problem (19),(23) for the Buckley-Leverett equation. Recall that this is in order to model the propagation of the shock front, which is of primary importance in the relevant application, where the accurate approximation of the shock front requires particular care. This is in our method mainly accomplished by the adaptive modification of the nodes during the simulation. For details concerning the construction of the required adaption rules, we refer to [7].

Now let us turn straight to our numerical results, provided by our particle advection scheme. In our simulation, we decided to select a constant time step size $\tau = 5 \cdot 10^{-5}$, and the simulation comprises 2100 time steps, so that $I = [0, 2100\tau]$. Moreover, we let $\mu = 0.5$ for the viscosity ratio of water and oil, appearing in the fractional flow function (20).

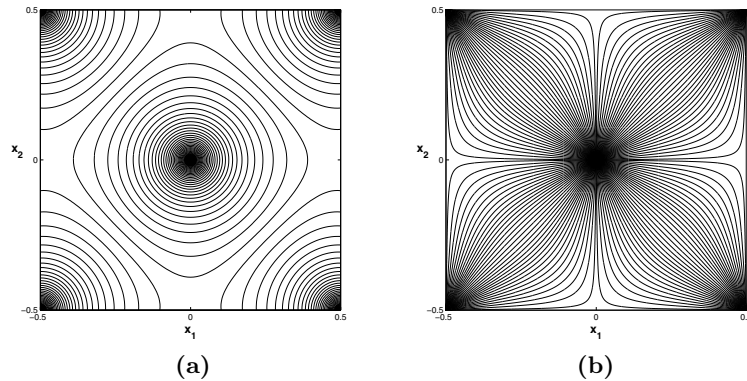


Fig. 3. Five-spot problem. (a) Contours of the pressure field, (b) streamlines of the velocity field.

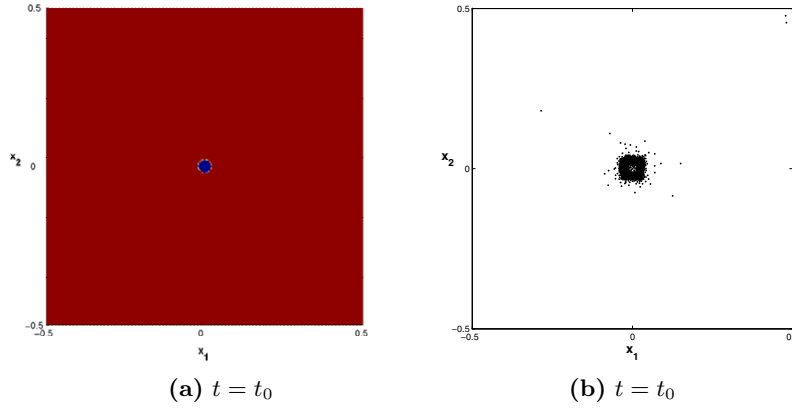


Fig. 4. Five-spot problem. (a) Initial condition, (b) initial node distribution.

The initial conditions $u(0, x)$ are shown in Figure 4, where also the initial node distribution is shown. Moreover, Figure 5 shows the water saturation u during the simulation at three different times, $t = t_{420}$, $t = t_{1260}$, and $t = t_{2100}$. Figure 5 shows also the corresponding node distribution. The corresponding color code for the water saturation is shown at the right margin of Figure 5, respectively.

Note that the shock front, at the interface between the non-wetting fluid (oil, $u \equiv 0$) and the wetting fluid (water, $u \equiv 1$), is moving from the center towards the four corner points of the computational domain Ω . This way, the non-wetting fluid (oil) is effectively displaced by the wetting fluid (water) into the four production wells, as expected.

Due to the adaptive distribution of the nodes, the shock front propagation of the solution u is captured very well. This helps to reduce the required computational costs while maintaining the accuracy, due to a higher resolution around the shock front. The effective distribution of the nodes around the shock supports the utility of the adaption rules, proposed in our previous paper [7], yet once more.

Acknowledgement

The fruitful joint collaboration with Jörn Behrens and Martin Käser in previous work on related particle methods is greatly appreciated. The author was partly supported by the European Union within the project NetAGES (Network for Automated Geometry Extraction from Seismic), contract no. IST-1999-29034.

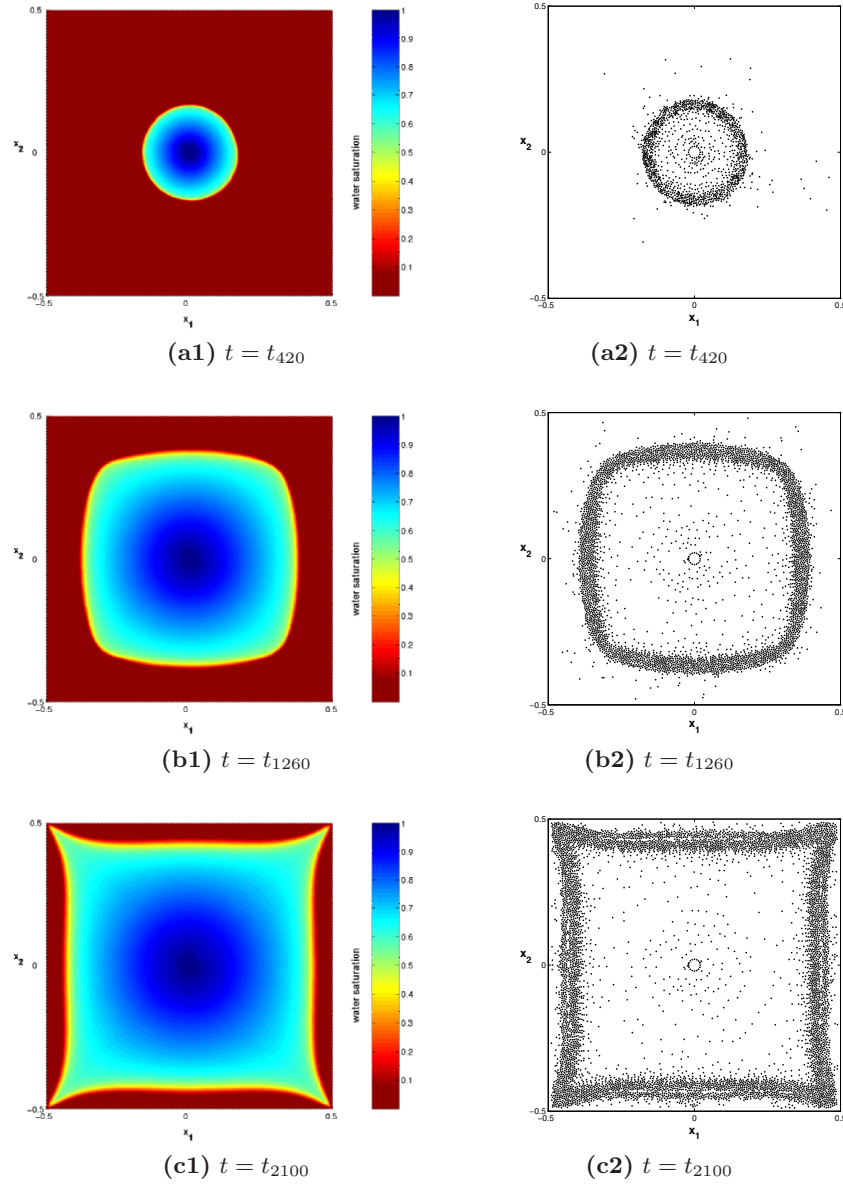


Fig. 5. Five-spot problem. Solution obtained by our particle simulation. The color plots in indicate the water saturation u during the simulation at three different times, (a1) $t = t_{420}$, (b1) $t = t_{1260}$, (c1) $t = t_{2100}$. The corresponding adaptive node distributions are shown in (a2),(b2),(c2).

References

1. R. Abgrall: On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation, *J. Comput. Phys.* **144**, 1994, 45–58.
2. N. Albright, P. Concus, and W. Proskurowski: Numerical solution of the multi-dimensional Buckley-Leverett equation by a sampling method. Paper SPE 7681, Soc. Petrol. Eng. Fifth Symp. on Reservoir Simulation, Denver, Feb. 1979.
3. K. Aziz and A. Settari: *Petroleum Reservoir Simulation*. Applied Science, London, 1979.
4. I. Babuška and J.M. Melenk: The partition of unity method. *Int. J. Numer. Meths. Eng.* **40**, 1997, 727–758.
5. J. Behrens: Atmospheric and ocean modeling with an adaptive finite element solver for the shallow-water equations. *Applied Numerical Mathematics* **26**, 1998, 217–226.
6. J. Behrens and A. Iske: Grid-free adaptive semi-Lagrangian advection using radial basis functions. *Computers and Mathematics with Applications* **43**(3-5), 2002, 319–327.
7. J. Behrens, A. Iske, and S. Pöhn: Effective node adaption for grid-free semi-Lagrangian advection. In: *Discrete Modelling and Discrete Algorithms in Continuum Mechanics*, T. Sonar and I. Thomas (eds.), Logos, 2001, 110–119.
8. J. Behrens, A. Iske, and M. Käser: Adaptive meshfree method of backward characteristics for nonlinear transport equations. In: *Meshfree Methods for Partial Differential Equations*, M. Griebel and M.A. Schweitzer (eds.), Springer, 2003, 21–36.
9. A. Bregman, A. F. J. van Velthoven, F. G. Wienhold, H. Fischer, T. Zenker, A. Waibel, A. Frenzel, F. Arnold, F., G. W. Harris, M. J. A. Bolder, and J. Lelieveld: Aircraft measurements of O_3 , HNO_3 , and N_2O in the winter arctic lower stratosphere during the stratosphere-troposphere experiment by aircraft measurements (STREAM) 1. *J. Geophys. Res.* **100**, 1995, 11245–11260.
10. K. Dethloff, A. Rinke, R. Lehmann, J. H. Christensen, M. Botzet, and B. Machenhauer: Regional climate model of the arctic atmosphere. *J. Geophys. Res.* **101**, 1996, 23401–23422.
11. J. Duchon: Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In: *Constructive Theory of Functions of Several Variables*, W. Schempp and K. Zeller (eds.), Springer, 1977, 85–100.
12. C.L. Farmer: A moving point method for arbitrary Peclet number multidimensional convection-diffusion equations. *IMA Journal of Numerical Analysis* **5**, 1985, 465–480.
13. C. Gauger, P. Leinen, and H. Yserentant: The finite mass method. *SIAM J. Numer. Anal.* **37**, 2000, 1768–1799.
14. M. Griebel and M.A. Schweitzer (eds.): *Meshfree Methods for Partial Differential Equations*. Lecture Notes in Computational Science and Engineering, vol. 26, Springer, 2003.
15. M. Griebel and M.A. Schweitzer (eds.): *Meshfree Methods for Partial Differential Equations II*. Lecture Notes in Computational Science and Engineering, vol. 43, Springer, 2005.
16. M. Griebel and M.A. Schweitzer: A particle-partition of unity method for the solution of elliptic, parabolic and hyperbolic PDE. *SIAM J. Sci. Comp.* **22**(3), 2000, 853–890.

17. M. Griebel and M.A. Schweitzer: A particle-partition of unity method-part II: efficient cover construction and reliable integration. *SIAM J. Sci. Comp.* **23**(5), 2002, 1655–1682.
18. M. Griebel and M.A. Schweitzer: A particle-partition of unity method-part III: a multilevel solver. *SIAM J. Sci. Comp.* **24**(2), 2002, 377–409.
19. M. Griebel and M.A. Schweitzer: A particle-partition of unity method-part IV: parallelization. In *Meshfree Methods for Partial Differential Equations*, M. Griebel and M.A. Schweitzer (eds.), Springer, 2002, 161–192.
20. M. Griebel and M.A. Schweitzer: A particle-partition of unity method-part V: boundary conditions. In: *Geometric Analysis and Nonlinear Partial Differential Equations*, S. Hildebrandt and H. Karcher (eds.), Springer, 2002, 517–540.
21. M. Griebel, P. Oswald, and M.A. Schweitzer: a particle-partition of unity method-part VI: a p -robust multilevel solver. In: *Meshfree Methods for Partial Differential Equations II*, M. Griebel and M.A. Schweitzer (eds.), Springer, 2005, 71–92.
22. A. Harten, B. Engquist, S. Osher, and S. Chakravarthy: Uniformly high order essentially non-oscillatory schemes, III. *J. Comput. Phys.* **71**, 1987, 231–303.
23. D. Hietel, K. Steiner, and J. Struckmeier: A finite-volume particle method for compressible flows. *Math. Mod. Meth. Appl. Sci.* **10**(9), 2000, 1363–1382.
24. A. Iske: Reconstruction of functions from generalized Hermite-Birkhoff data. In: *Approximation Theory VIII, vol. 1: Approximation and Interpolation*, C.K. Chui and L.L. Schumaker (eds.), World Scientific, Singapore, 1995, 257–264.
25. A. Iske: On the approximation order and numerical stability of local Lagrange interpolation by polyharmonic splines. In: *Modern Developments in Multivariate Approximation*, W. Haussmann, K. Jetter, M. Reimer, and J. Stöckler (eds.), International Series of Numerical Mathematics **145**, 2003, Birkhäuser, Basel, 153–165.
26. A. Iske: Adaptive irregular sampling in meshfree flow simulation. In: *Sampling, Wavelets, and Tomography*, J. Benedetto and A. Zayed (eds.), Applied and Numerical Harmonic Analysis, Birkhäuser, Boston, 2004, 289–309.
27. A. Iske: Multiscale flow simulation by adaptive finite volume particle methods. *Proc. Appl. Math. Mech. (PAMM)* **5**(1), 2005, 771–772.
28. A. Iske and M. Käser: Two-phase flow simulation by AMMoC, an adaptive mesh-free method of characteristics. *Computer Modeling in Engineering & Sciences (CMES)* **7**(2), 2005, 133–148.
29. S. Jun, W.K. Liu, and T. Belytschko: Explicit reproducing kernel particle methods for large deformation problems. *Int. J. Numer. Meth. Engrg.* **41**, 1998, 137–166.
30. M. Junk: Do finite volume methods need a mesh? In: *Meshfree Methods for Partial Differential Equations*, M. Griebel, M.A. Schweitzer (eds.), Springer, 2003, 223–238.
31. M. Käser and A. Iske: Reservoir Flow Simulation by Adaptive ADER Schemes. In: *Mathematical Methods and Modelling in Hydrocarbon Exploration and Production*, A. Iske and T. Randen (eds.), Springer, 2005, 339–388.
32. M. Käser and A. Iske: ADER schemes on adaptive triangular meshes for scalar conservation laws. *Journal of Computational Physics* **205**(2), 2005, 486–508.
33. J. Kuhnert: An upwind finite pointset method (FPM) for compressible Euler- and Navier-Stokes equations. In: *Meshfree Methods for Partial Differential Equations*, M. Griebel and M.A. Schweitzer (eds.), 2003, 239–249.

34. R.L. LeVeque: *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, Cambridge, 2002.
35. S. Li and W.K. Liu: *Meshfree Particle Methods*. Springer, 2004.
36. W.K. Liu and S. Jun: Multiple scale reproducing kernel particle methods for large deformation problems. *Int. J. Numer. Meth. Engrg.* **41**, 1998, 1339–1362.
37. X. Liu, S. Osher, and T. Chan: Weighted essentially non-oscillatory schemes. *J. Comput. Phys.* **115**, 1994, 200–212.
38. W.R. Madych and S.A. Nelson: Multivariate interpolation: a variational theory. Manuscript, 1983.
39. W.R. Madych and S.A. Nelson: Multivariate interpolation and conditionally positive definite functions. *Approx. Theory Appl.* **4**, 1988, 77–89.
40. W.R. Madych and S.A. Nelson: Multivariate interpolation and conditionally positive definite functions II, *Math. Comp.* **54**, 1990, 211–230.
41. J.M. Melenk and I. Babuška: The partition of unity finite element method: basic theory and applications. *Comput. Meths. Appl. Mech. Engrg.* **139**, 1996, 289–314.
42. J.J. Monaghan: Smoothed particle hydrodynamics. *Rep. Prog. Phys.* **68**, 2005, 1703–1759.
43. H. Neunzert and J. Struckmeier: Particle methods for the Boltzmann equation. *Acta Numerica*, Cambridge, 1995, 417–457.
44. D.W. Peaceman: *Fundamentals of Numerical Reservoir Simulation*. Elsevier, Amsterdam, 1977.
45. F.P. Preparata and M.I. Shamos: *Computational Geometry*. Springer, New York, 1988.
46. A. Robert: A stable numerical integration scheme for the primitive meteorological equations. *Atmosphere-Ocean* **19**, 1981, 35–46.
47. A.E. Scheidegger: *The Physics of Flow in Porous Media*. University of Toronto Press, Toronto, 1974.
48. A. Staniforth and J. Côté: Semi-Lagrangian integration schemes for atmospheric models – a review. *Mon. Wea. Rev.* **119**, 1991, 2206–2223.
49. S. Tiwari, J. Kuhnert: Finite pointset method based on the projection method for simulations of the incompressible Navier-Stokes equations. In: *Meshfree Methods for Partial Differential Equations*, M. Griebel, M.A. Schweitzer (eds.), 2003, 373–387.

Enhancing SPH using Moving Least-Squares and Radial Basis Functions

Robert Brownlee¹, Paul Houston², Jeremy Levesley¹, and Stephan Rosswog³

¹ Department of Mathematics, University of Leicester, Leicester, LE1 7RH, UK, {r.brownlee,j.levesley}@mcs.le.ac.uk

² School of Mathematical Sciences, University of Nottingham, Nottingham NG7 2RD, UK, paul.houston@nottingham.ac.uk

³ School of Engineering and Science, International University Bremen, D-28759 Bremen, Germany, s.rosswog@iu-bremen.de

Summary. In this paper we consider two sources of enhancement for the meshfree Lagrangian particle method *smoothed particle hydrodynamics* (SPH) by improving the accuracy of the particle approximation. Namely, we will consider *shape functions* constructed using: moving least-squares approximation (MLS); radial basis functions (RBF). Using MLS approximation is appealing because polynomial consistency of the particle approximation can be enforced. RBFs further appeal as they allow one to dispense with the *smoothing-length* – the parameter in the SPH method which governs the number of particles within the support of the shape function. Currently, only ad hoc methods for choosing the smoothing-length exist. We ensure that any enhancement retains the conservative and meshfree nature of SPH. In doing so, we derive a new set of variationally-consistent hydrodynamic equations. Finally, we demonstrate the performance of the new equations on the Sod shock tube problem.

1 Introduction

Smoothed particle hydrodynamics (SPH) is a meshfree Lagrangian particle method primarily used for solving problems in solid and fluid mechanics (see [10] for a recent comprehensive review). Some of the attractive characteristics that SPH possesses include: the ability to handle problems with large deformation, free surfaces and complex geometries; truly meshfree nature (no background mesh required); exact conservation of momenta and total energy. On the other hand, SPH suffers from several drawbacks: an instability in tension; difficulty in enforcing essential boundary conditions; fundamentally based on inaccurate kernel approximation techniques. This paper addresses the last of these deficiencies by suggesting improved particle approximation procedures. Previous contributions in this direction (reviewed in [2]) have focused on corrections of the existing SPH particle approximation (or its deriva-

tives) by enforcing polynomial consistency. As a consequence, the conservation of relevant physical quantities by the discrete equations is usually lost.

The outline of the paper is as follows. In the next section we review how SPH equations for the non-dissipative motion of a fluid can be derived. In essence this amounts to a discretization of the Euler equations:

$$\frac{d\rho}{dt} = -\rho\nabla \cdot v, \quad \frac{dv}{dt} = -\frac{1}{\rho}\nabla P, \quad \frac{de}{dt} = -\frac{P}{\rho}\nabla \cdot v, \quad (1)$$

where $\frac{d}{dt}$ is the total derivative, ρ , v , e and P are the density, velocity, thermal energy per unit mass and pressure, respectively. The derivation is such that important conservation properties are satisfied by the discrete equations. Within the same section we derive a new set of variationally-consistent hydrodynamic equations based on improved particle approximation. In Sect. 3 we construct specific examples – based on moving least-squares approximation and radial basis functions – to complete the newly derived equations. The paper finishes with Sect. 4 where we demonstrate the performance of the new methods on the Sod shock tube problem [12] and make some concluding remarks.

To close this section, we briefly review the SPH particle approximation technique on which the SPH method is fundamentally based and which we purport to be requiring improvement. From a set of scattered particles $\{x_1, \dots, x_N\} \subset \mathbb{R}^d$, SPH particle approximation is achieved using

$$Sf(x) = \sum_{j=1}^N f(x_j) \frac{m_j}{\rho_j} W(|x - x_j|, h), \quad (2)$$

where m_j and ρ_j denotes the mass and density of the j th particle, respectively. The function W is a normalised kernel function which approximates the δ -distribution as the *smoothing-length*, h , tends to zero. The function $\frac{m_j}{\rho_j} W(|x - x_j|, h)$ is called an SPH *shape function* and the most popular choice for W is a compactly supported cubic spline kernel with support $2h$. The parameter h governs the extent to which contributions from neighbouring particles are allowed to smooth the approximation to the underlying function f . Allowing a spatiotemporally varying smoothing-length increases the accuracy of an SPH simulation considerably. There are a selection of ad hoc techniques available to accomplish this, although often terms arising from the variation in h are neglected in the SPH method. The approximating power of the SPH particle approximation is perceived to be poor. The SPH shape functions fail to provide a partition of unity so that even the constant function is not represented exactly. There is currently no approximation theory available for SPH particle approximation when the particles are in general positions. The result of a shock tube simulation using the SPH equations derived in Sect. 2 is shown in Fig. 1 (see Sect. 4 for the precise details of the simulation). The difficulty that SPH has at the contact discontinuity ($x \approx 0.2$) and the head of the rarefaction

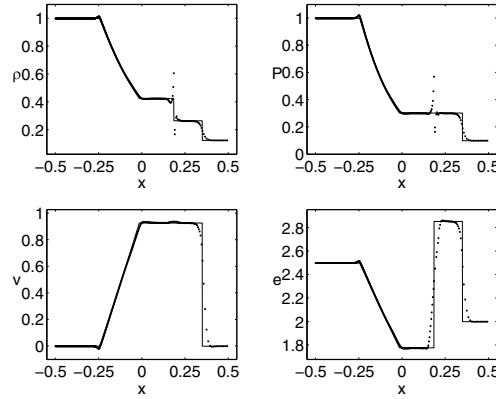


Fig. 1. Shock tube simulation ($t=0.2$) using SPH.

wave ($x \approx -0.25$) is attributed to a combination of the approximation (2) and the variable smoothing-length not being self-consistently incorporated.

2 Variationally-Consistent Hydrodynamic Equations

It is well known (see [10] and the references cited therein) that the most common SPH equations for the non-dissipative motion of a fluid can be derived using the Lagrangian for hydrodynamics and a variational principle. In this section we review this procedure for a particular formulation of SPH before deriving a general set of variationally-consistent hydrodynamic equations.

The aforementioned Lagrangian is a particular functional of the dynamical coordinates: $L(x, v) = \int \rho(v^2/2 - e) dx$, where x is the position, v is the velocity, ρ is the density, e is the thermal energy per unit mass and the integral is over the volume being discretized. Given N particles $\{x_1 \dots, x_N\} \subset \mathbb{R}^d$, the SPH discretization of the Lagrangian, also denoted by L , is given by

$$L = \sum_{j=1}^N m_j \left(\frac{v_j^2}{2} - e_j \right), \quad (3)$$

where m_j has replaced $\rho_j V_j$ to denote particle mass (assumed to be constant), and V_j is a volume associated with each particle. Self-evidently, the notation f_j is used to denote the function f evaluated at the j th particle.

The Euler-Lagrange equations give rise to SPH equations of motion provided each quantity in (3) can be written directly as a function of the particle coordinates. By setting $f = \rho$ in (2) and evaluating at x_j , we can obtain an expression for ρ_j directly as a function of the particle coordinates. Therefore, because we assume that $e_j = e_j(\rho_j)$, the Euler-Lagrange equations are amenable. Furthermore, in using this approach, conservation of momenta and

total energy are guaranteed via Noether's symmetry theorem. However, when we consider improved particle approximation, the corresponding expression for density depends on the particle coordinates in an implicit manner, so that the Euler–Lagrange equations are not directly amenable. To circumvent this difficulty, one can use the principle of stationary action directly to obtain SPH equations of motion – the *action*,

$$S = \int L dt,$$

being the time integral of L . The principle of stationary action demands that the action is invariant with respect to small changes in the particle coordinates (i.e., $\delta S = 0$). The Euler–Lagrange equations are a consequence of this variational principle. In [10] it is shown that if an expression for the time rate of change of ρ_j is available, then, omitting the detail, this variational principle gives rise to SPH equations of motion.

To obtain an expression for the time rate of change of density we can discretize the first equation of (1) using (2) by collocation. By assuming that the SPH shape functions form a partition of unity we commit error but are able to artificially provide the discretization with invariance to a constant shift in velocity (Galilean invariance):

$$\frac{d\rho_i}{dt} = -\rho_i \sum_{j=1}^N \frac{m_j}{\rho_j} (v_j - v_i) \cdot \nabla_i W(|x_i - x_j|, h_i), \quad i = 1, \dots, N, \quad (4)$$

where ∇_i is the gradient with respect to the coordinates of the i th particle. The equations of motion that are variationally-consistent with (4) are

$$\frac{dv_i}{dt} = -\frac{1}{\rho_i} \sum_{j=1}^N \frac{m_j}{\rho_j} \left(P_i \nabla_i W(|x_i - x_j|, h_i) + P_j \nabla_i W(|x_i - x_j|, h_j) \right), \quad (5)$$

for $i = 1, \dots, N$, where P_i denotes the pressure of the i th particle (provided via a given equation of state). Using the first law of thermodynamics, the equation for the rate of change of thermal energy is given by

$$\frac{de_i}{dt} = \frac{P_i}{\rho_i^2} \frac{d\rho_i}{dt}, \quad i = 1, \dots, N. \quad (6)$$

As already noted, a beneficial consequence of using the Euler–Lagrange equations is that one automatically preserves, in the discrete equations, fundamental conservation properties of the original system (1). Since we have not done this, conservation properties are not necessarily guaranteed by our discrete equations (4)–(6). However, certain features of the discretization (4) give us conservation. Indeed, by virtue of (4) being Galilean invariant, one conserves linear momentum and total energy (assuming perfect time integration). Remember that Galilean invariance was installed under the erroneous

assumption that the SPH shape functions provide a partition of unity. Angular momentum is also explicitly conserved by this formulation due to W being symmetric.

Now, we propose to enhance SPH by improving the particle approximation (2). Suppose we have constructed shape functions ϕ_j that provide at least a partition of unity. With these shape functions we form a quasi-interpolant:

$$Sf = \sum_{j=1}^N f(x_j)\phi_j, \quad (7)$$

which we implicitly assume provides superior approximation quality than that provided by (2). We defer particular choices for ϕ_j until the next section. The discretization of the continuity equation now reads

$$\frac{d\rho_i}{dt} = -\rho_i \sum_{j=1}^N (v_j - v_i) \cdot \nabla \phi_j(x_i), \quad i = 1, \dots, N, \quad (8)$$

where, this time, we have supplied genuine Galilean invariance, without committing an error, using the partition of unity property of ϕ_j . As before, the principle of stationary action provides the equations of motion and conservation properties of the resultant equations reflect properties present in the discrete continuity equation (8).

To obtain (3), two assumptions were made. Firstly, the SPH shape functions were assumed to form a sufficiently good partition of unity. Secondly, it was assumed that the kernel approximation $\int fW(|\cdot - x_j|, h) dx \approx f(x_j)$, was valid. For our general shape functions the first of these assumptions is manifestly true. The analogous assumption we make to replace the second is that the error induced by the approximations

$$\int f\phi_j dx \approx f_j \int \phi_j dx \approx f_j V_j, \quad j = 1, \dots, N, \quad (9)$$

is negligible. With the assumption (9), the approximate Lagrangian associated with ϕ_j is identical in form to (3). Neglecting the details once again, which can be recovered from [10], the equations of motion variationally-consistent with (8) are

$$\frac{dv_i}{dt} = \frac{1}{m_i} \sum_{j=1}^N \frac{m_j}{\rho_j} P_j \nabla \phi_i(x_j), \quad i = 1, \dots, N, \quad (10)$$

The equations (6), (8) and (10) constitute a new set of variationally-consistent hydrodynamic equations. They give rise to the formulation of SPH derived earlier under the transformation $\phi_j(x_i) \mapsto \frac{m_j}{\rho_j} W(|x_i - x_j|, h_i)$. The equations of motion (10) appear in [8] but along side variationally-inconsistent companion equations. The authors advocate using a variationally-consistent

set of equation because evidence from the SPH literature (e.g., [3, 9]) suggests that not doing so can lead to poor numerical results.

Linear momentum and total energy are conserved by the new equations, and this can be verified immediately using the partition of unity property of ϕ_j . The ϕ_j will not be symmetric. However, if it is also assumed that the shape functions reproduce linear polynomials, namely, $\sum x_j \phi_j(x) = x$, then it is simple to verify that angular momentum is also explicitly conserved.

3 Moving Least-Squares and Radial Basis Functions

In this section we construct quasi-interpolants of the form (7). In doing so we furnish our newly derived hydrodynamic equations (6), (8) and (10) with several examples.

Moving least-squares (MLS). The preferred construction for MLS shape functions, the so-called Backus–Gilbert approach [4], seeks a quasi-interpolant of the form (7) such that:

- $Sp = p$ for all polynomials p of some fixed degree;
- $\phi_j(x)$, $j = 1, \dots, N$, minimise the quadratic form $\sum \phi_j^2(x) \left[w\left(\frac{|x-x_j|}{h}\right) \right]^{-1}$,

where w is a fixed *weight function*. If w is continuous, compactly supported and positive on its support, this quadratic minimisation problem admits a unique solution. Assuming f has sufficient smoothness, the order of convergence of the MLS approximation (7) directly reflects the degree of polynomial reproduced [14].

The use of MLS approximation in an SPH context has been considered before. Indeed, Belytschko et al. [2] have shown that correcting the SPH particle approximation up to linear polynomials is equivalent to an MLS approximation with $w(|\cdot - x_j|/h) = W(|\cdot - x_j|, h)$. There is no particular reason to base the MLS approximation on an SPH kernel. We find that MLS approximations based on Wendland functions [13], which have half the natural support of a typical SPH kernel, produce results which are less noisy. Dilts [7, 8] employs MLS approximation too. Indeed, in [7], Dilts makes an astute observation that addresses an inconsistency that arises due to (9) – we have the equations

$$\frac{dV_i}{dt} = V_i \sum_{j=1}^N (v_j - v_i) \cdot \nabla \phi_j(x_i) \quad \text{and} \quad \frac{dV_i}{dt} \approx \frac{d}{dt} \left(\int \phi_i(x) dx \right).$$

Dilts shows that if h_i is evolved according to $h_i \propto V_i^{1/d}$ then there is agreement between the right-hand sides of these equations when a one-point quadrature of $\int \phi_i dx$ is employed. Thus, providing some theoretical justification for choosing this particular variable smoothing-length over other possible choices.

Radial basis functions (RBFs). To construct an RBF interpolant to an unknown function f on x_1, \dots, x_N , one produces a function of the form

$$If = \sum_{j=1}^N \lambda_j \psi(|\cdot - x_j|), \quad (11)$$

where the λ_j are found by solving the linear system $If(x_i) = f(x_i)$, $i = 1, \dots, N$. The *radial basis function*, ψ , is a pre-specified univariate function chosen to guarantee the solvability of this system. Depending on the choice of ψ , a low degree polynomial is sometimes added to (11) to ensure solvability, with the additional degrees of freedom taken up in a natural way. This is the case with the *polyharmonic splines*, which are defined, for $m > d/2$, by $\psi(|x|) = |x|^{2m-d} \log |x|$ if d is even and $\psi(|x|) = |x|^{2m-d}$ otherwise, and a polynomial of degree $m - 1$ is added. The choice $m \geq 2$ ensures the RBF interpolant reproduces linear polynomials as required for angular momentum to be conserved by the equations of motion. As with MLS approximation, one has certain strong assurances regarding the quality of the approximation induced by the RBF interpolant (e.g. [6] for the case of polyharmonic splines).

In its present form (11), the RBF interpolant is not directly amenable. One possibility is to rewrite the interpolant in *cardinal form* so that it coincides with (7). This naively constitutes much greater computational effort. However, there are several strategies for constructing approximate cardinal RBF shape functions (e.g. [5]) and fast evaluation techniques (e.g. [1]) which reduce this work significantly. The perception of large computational effort is an attributing factor as to why RBFs have not been considered within an SPH context previously. Specifically for polyharmonic splines, another possibility is to construct shape functions based on discrete m -iterated Laplacians of ψ . This is sensible because the continuous iterated Laplacian, when applied ψ , results in the δ -distribution (up to a constant). This is precisely the approach we take in Sect. 4 where we employ cubic B-spline shape functions for one of our numerical examples. The cubic B-splines are discrete bi-Laplacians of the shifts of $|\cdot|^3$, and they gladly reproduce linear polynomials.

In addition to superior approximation properties, using globally supported RBF shape functions has a distinct advantage. One has dispensed with the smoothing-length entirely. Duely, issues regarding how to correctly vary and self-consistently incorporate the smoothing-length vanish. Instead, a natural ‘support’ is generated related to the relative clustering of particles.

4 Numerical Results

In this section we demonstrate the performance of the scheme (6), (8) and (10) using both MLS and RBF shape functions. The test we have selected has become a standard one-dimensional numerical test in compressible fluid flow – the Sod shock tube [12]. The problem consists of two regions of ideal gas,

one with a higher pressure and density than the other, initially at rest and separated by a diaphragm. The diaphragm is instantaneously removed and the gases allowed to flow resulting in a rarefaction wave, contact discontinuity and shock. We set up 450 equal mass particles in $[-0.5, 0.5]$. The gas occupying the left-hand and right-hand sides of the domain are given initial conditions $(P_L, \rho_L, v_L) = (1.0, 1.0, 0.0)$ and $(P_R, \rho_R, v_R) = (0.1, 0.125, 0.0)$, respectively. The initial condition is not smoothed.

With regards to implementation, artificial viscosity is included to prevent the development of unphysical oscillations. The form of the artificial viscosity mimics that of the most popular SPH artificial viscosity and is applied with a switch which reduces the magnitude of the viscosity by a half away from the shock. A switch is also used to administer an artificial thermal conductivity term, also modelled in SPH. Details of both dissipative terms and their respective switches can be accessed through [10]. Finally, we integrate, using a predictor–corrector method, the equivalent hydrodynamic equations

$$\frac{dV_i}{dt} = V_i \sum_{j=1}^N (v_j - v_i) \cdot \nabla \phi_j(x_i), \quad (12)$$

$$\frac{dv_i}{dt} = \frac{1}{m_i} \sum_{j=1}^N V_j P_j \nabla \phi_i(x_j), \quad \frac{de_i}{dt} = -\frac{P_i}{m_i} \frac{dV_i}{dt},$$

together with $\frac{dx_i}{dt} = v_i$, to move the particles. To address the consistency issue regarding particle volume mentioned earlier – which is partially resolved by evolving h in a particular way when using MLS approximation – we periodically update the particle volume predicted by (12) with $\int \phi_i dx$ if there is significant difference between these two quantities. To be more specific, the particle volume V_i is updated if $|V_i - \int \phi_i dx|/V_i \geq 1.0 \times 10^{-3}$.

We first ran a simulation with linearly complete MLS shape functions. The underlying univariate function, w , was selected to be a Wendland function with C^4 -smoothness. The smoothing-length was evolved by taking a time derivative of the relationship $h_i \propto V_i$ and integrating it alongside the other equations, the constant of proportionality was chosen to be 2.0. The result is shown in Fig. 2. The agreement with the analytical solution (solid line) is excellent, especially around the contact discontinuity and the head of the rarefaction wave. Next, we constructed RBF shape functions. As we mentioned in Sect. 3, for this one-dimensional problem we employ cubic B-spline because they constitute discrete bi-Laplacians of the shifts of the globally supported basis function, $\psi = |\cdot|^3$. The result of this simulation is shown in Fig. 3. Again, the agreement with the analytical solution is excellent.

In the introduction an SPH simulation of the shock tube was displayed (Fig. 1). There, we integrated (4)–(6) and h was updated by taking a time derivative of the relationship $h_i = 2.0m_i/\rho_i$. To keep the comparison fair, the same initial condition, particle setup and dissipative terms were used. As

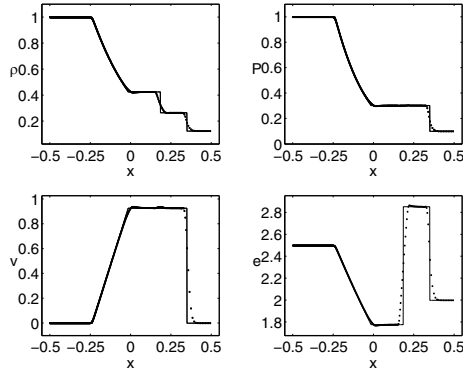


Fig. 2. Shock tube simulation ($t=0.2$) using linearly complete MLS shape functions.

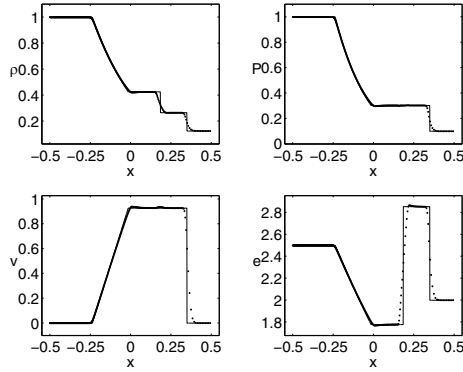


Fig. 3. Shock tube simulation ($t=0.2$) using cubic B-spline shape functions.

previously noted, this formulation of SPH performs poorly on this problem, especially around the contact discontinuity. Furthermore, we find that this formulation of SPH does not converge in the L_∞ -norm for this problem. At a fixed time ($t = 0.2$), plotting number of particles, N , versus L_∞ -error in pressure, in the region of the computational domain where the solution is smooth reveals an approximation order of around $2/3$, attributed to the low regularity of the analytical solution, for the MLS and RBF methods, whereas our SPH simulation shows no convergence. This is not to say that SPH can not perform well on this problem. Indeed, Price [11] shows that, for a formulation of SPH where density is calculated via summation and variable smoothing-length terms correctly incorporated, the simulation does exhibit convergence in pressure. The SPH formulation we have used is fair for comparison with the MLS and RBF methods since they all share a common derivation. In particular, we are integrating the continuity equation in each case.

To conclude, we have proposed a new set of discrete conservative variationally-consistent hydrodynamic equations based on a partition of unity. These

equations, when actualised with MLS and RBF shape functions, outperform the SPH method on the shock tube problem. Further experimentation and numerical analysis of the new methods is a goal for future work.

Acknowledgements

The first author would like to acknowledge Joe Monaghan, whose captivating lectures at the XIth NA Summer School held in Durham in July 2004 provided much inspiration for this work. Similarly, the first author would like to thank Daniel Price for his useful discussions, helpful suggestions and hospitality received during a visit to Exeter in May 2005. This work is supported by EPSRC grants GR/S95572/01 and GR/R76615/02.

References

1. R.K. Beaton, W.A. Light: Fast evaluation of radial basis functions: methods for two-dimensional polyharmonic splines. *IMA J. Numer. Anal.* **17**, 1997, 343–372.
2. T. Belytschko, B. Krongauz, D. Organ, M. Fleming, P. Krysl: Meshless methods: an overview and recent developments. *Comp. Meth. Appl. Mech. Engrg.* **139**, 1996, 3–47.
3. J. Bonet, T.S.L. Lok: Variational and momentum preservation aspects of smooth particle hydrodynamic formulations. *Comp. Meth. Appl. Mech. Engrg.* **180**, 1999, 97–115.
4. L.P. Bos and K. Šalkauskas: Moving least-squares are Backus-Gilbert optimal. *J. Approx. Theory* **59**(3), 1989, 267–275.
5. D. Brown, L. Ling, E. Kansa, and J. Levesley: On approximate cardinal preconditioning methods for solving PDEs with radial basis functions. *Eng. Anal. Bound. Elem.* **29**, 2005, 343–353.
6. R.A. Brownlee and W.A. Light: Approximation orders for interpolation by surface splines to rough functions. *IMA J. Numer. Anal.* **24**(2), 2004, 179–192.
7. G.A. Dilts: Moving-least-squares-particle hydrodynamics. I. consistency and stability. *Int. J. Numer. Meth. Engrg.* **44**(8), 1999, 1115–1155.
8. G.A. Dilts: Moving least-squares particle hydrodynamics. II. conservation and boundaries. *Int. J. Numer. Meth. Engrg.* **48**(10), 2000, 1503–1524.
9. S. Marri and S.D.M. White: Smoothed particle hydrodynamics for galaxy-formation simulations: improved treatments of multiphase gas, of star formation and of supernovae feedback. *Mon. Not. R. Astron. Soc.* **345**, 2003, 561–574.
10. J.J. Monaghan: Smoothed particle hydrodynamics. *Rep. Prog. Phys.* **68**, 2005, 1703–1759.
11. D. Price: *Magnetic Fields in Astrophysics*. Ph.D., University of Cambridge, 2004.
12. G.A. Sod: A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *J. Comput. Phys.* **27**(1), 1978, 1–31.
13. H. Wendland: Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Adv. Comput. Math.* **4**(4), 1995, 389–396.
14. H. Wendland: Local polynomial reproduction and moving least squares approximation. *IMA J. Numer. Anal.* **21**(1), 2001, 285–300.

Stepwise Calculation of the Basin of Attraction in Dynamical Systems Using Radial Basis Functions

Peter Giesl

Munich University of Technology, Department of Mathematics, Germany,
giesl@ma.tum.de

Summary. This paper deals with the application of radial basis functions to dynamical systems. More precisely, we discuss the approximation of the solution of a Cauchy problem, a linear first-order partial differential equation with non-constant coefficients, using radial basis functions in Section 1. In Section 2 we introduce a dynamical system given by a system of ordinary differential equations and define the basin of attraction of an equilibrium. The ODE is the characteristic equation of the PDE of Section 1. On the other hand, a solution and even an approximate solution of the PDE is a Lyapunov function of the ODE, i.e. its orbital derivative is negative. Lyapunov functions serve to determine the basin of attraction through level sets. In Section 3 we use the approximative solutions of the PDE as Lyapunov functions to determine the basin of attraction. We show, how this procedure can be applied stepwise and illustrate this by an example.

1 Radial Basis Functions and a Cauchy Problem

In this section we discuss the approximation of the solution of a Cauchy problem using radial basis functions. We consider a linear first-order partial differential equation with Cauchy conditions. The difference to other approaches, cf. [1] and [2], is that the partial differential equation has non-constant coefficients. We use Wendland's functions as radial basis functions. In this section we provide the setting, prove positive definiteness of the interpolation matrix and an error estimate.

Consider the linear first-order partial differential equation with non-constant coefficients for the function u

$$\begin{aligned} \sum_{k=1}^d f_k(x) \frac{\partial u}{\partial x_k}(x) &= -c & \text{for } x \in \Omega, \\ u(x) &= c_0 & \text{for } x \in \Gamma. \end{aligned} \tag{1}$$

Here, $f \in C^\sigma(\mathbb{R}^d, \mathbb{R}^d)$, $\sigma \geq 1$, $d \in \mathbb{N}$ and the constants $c \in \mathbb{R}$ and $c_0 \in \mathbb{R}^+$ are given. $\Omega \subset \mathbb{R}^d$ is an open set and $\Gamma \subset \Omega$ is a non-characteristic hypersurface

of the form $\Gamma = \{x \in \mathbb{R}^d \mid g(x) = \rho\}$ with suitable function g . For the moment we assume existence and uniqueness of a solution $u \in C^\sigma(\Omega, \mathbb{R})$, cf. also Proposition 4.

We seek to approximate the solution of (1) using radial basis functions. Radial basis functions have been used to solve partial differential equations in e.g. [7], [5] and [1]. We follow [1] and define the linear operators

$$\begin{aligned} Lu(x) &:= \sum_{k=1}^d f_k(x) \frac{\partial u}{\partial x_k}(x), \\ L_0 u(x) &:= u(x). \end{aligned}$$

In contrast to [1] the operator L is not translation-invariant. We can write (1) as the following mixed linear problem

$$\begin{aligned} Lu(x) &= -c & \text{for } x \in \Omega, \\ L_0 u(x) &= c_0 & \text{for } x \in \Gamma. \end{aligned} \quad (2)$$

We fix a radial basis function $\Psi \in C^2(\mathbb{R}^d, \mathbb{R})$ of the form $\Psi(x) = \psi(\|x\|)$; in this paper we will use Wendland's functions, cf. [6]. We approximate the solution u by a function s , also called the reconstruction of u . We fix grids $X_N = \{x_1, \dots, x_N\} \subset \Omega$ and $\Xi_M = \{\xi_1, \dots, \xi_M\} \subset \Gamma$ and use the following mixed ansatz for s :

$$s(x) = \sum_{i=1}^N \beta_i (\delta_{x_i} \circ L)^y \Psi(x - y) + \sum_{j=1}^M \gamma_j (\delta_{\xi_j} \circ L_0)^y \Psi(x - y). \quad (3)$$

Here, δ_x denotes Dirac's δ -distribution, and the superscript y denotes the application of the operator with respect to y . The coefficients $\beta_i, \gamma_j \in \mathbb{R}$ are chosen such that s satisfies (2) for all grid points, i.e.

$$\begin{aligned} Ls(x_i) = Lu(x_i) &= -c, \\ L_0 s(\xi_j) = L_0 u(\xi_j) &= c_0 \end{aligned} \quad (4)$$

holds. Equations (4) are equivalent to the system of linear equations

$$\begin{pmatrix} A & B \\ B^T & C \end{pmatrix} \begin{pmatrix} \beta \\ \gamma \end{pmatrix} = \alpha, \quad (5)$$

where $\alpha = (-c, \dots, -c, c_0, \dots, c_0)^T$, and $A = (a_{jk})$, $B = (b_{jk})$ and $C = (c_{jk})$ are given by

$$\begin{aligned} a_{jk} &= (\delta_{x_j} \circ L)^x (\delta_{x_k} \circ L)^y \Psi(x - y), & b_{jk} &= (\delta_{x_j} \circ L)^x (\delta_{x_k} \circ L_0)^y \Psi(x - y) \\ \text{and } c_{jk} &= (\delta_{x_j} \circ L_0)^x (\delta_{x_k} \circ L_0)^y \Psi(x - y). \end{aligned}$$

We will show later, cf. Proposition 3, that the interpolation matrix in (5) is positive definite and thus the system (5) has a unique solution (β, γ) , which determines s by (3).

For the rest of this paper Ψ is defined by Wendland's compactly supported radial basis functions. For the example of Section 3 we use $\Psi(x) = \psi_{4,2}(\mu\|x\|)$ with $\mu > 0$ in \mathbb{R}^2 where $\psi_{4,2}(r) = (1-r)_+^6 [35r^2 + 18r + 3]$.

Definition 1 (Wendland's functions, [6]). We set $\Psi(x) = \psi_{l,k}(\mu\|x\|)$, where $\mu > 0$ and $\psi_{l,k}$ is a Wendland function, cf. [6], with $k \in \mathbb{N}$ and $l := \lceil \frac{d}{2} \rceil + k + 1$.

We recall some properties of Wendland's functions.

Proposition 1. Let $\Psi(x)$ be as in Definition 1. Then

1. $\Psi \in C^{2k}(\mathbb{R}^d, \mathbb{R})$ and Ψ has compact support.
2. For the Fourier transform $\hat{\Psi}(\omega) = \int_{\mathbb{R}^d} \Psi(x)e^{-ix^T\omega} dx$ we have

$$C_1 (1 + \|\omega\|^2)^{-\frac{d+1}{2}-k} \leq \hat{\Psi}(\omega) \leq C_2 (1 + \|\omega\|^2)^{-\frac{d+1}{2}-k} \tag{6}$$

with positive constants C_1, C_2 .

We define the native space and its dual. In the following $\mathcal{S}'(\mathbb{R}^d)$ denotes the dual of the Schwartz space $\mathcal{S}(\mathbb{R}^d)$ of rapidly decreasing functions.

Definition 2. We define the Hilbert space

$$\mathcal{F}^* := \left\{ \lambda \in \mathcal{S}'(\mathbb{R}^d) \mid \int_{\mathbb{R}^d} |\hat{\lambda}(\omega)|^2 \hat{\Psi}(\omega) d\omega < \infty \right\}$$

with the scalar product

$$\langle \lambda, \mu \rangle_{\mathcal{F}^*} := (2\pi)^{-d} \int_{\mathbb{R}^d} \hat{\lambda}(\omega) \overline{\hat{\mu}(\omega)} \hat{\Psi}(\omega) d\omega.$$

The native space \mathcal{F} is identified with the dual \mathcal{F}^{**} of \mathcal{F}^* . The norm is given by

$$\|g\|_{\mathcal{F}} := \sup_{\lambda \in \mathcal{F}^*, \lambda \neq 0} \frac{|\lambda(g)|}{\|\lambda\|_{\mathcal{F}^*}}.$$

The native space in the case of Wendland's functions is the well-known Sobolev space due to (6).

Proposition 2. If Ψ is as in Definition 1, then

$$\mathcal{F}^* = H^{-\frac{d+1}{2}-k}(\mathbb{R}^d),$$

where $H^{-\frac{d+1}{2}-k}(\mathbb{R}^d)$ denotes the Sobolev space. Moreover,

$$C_0^\sigma(\mathbb{R}^d) \subset \mathcal{F} = H^{\frac{d+1}{2}+k}(\mathbb{R}^d)$$

with $\mathbb{N} \ni \sigma \geq \sigma^* := \frac{d+1}{2} + k$. Here $C_0^\sigma(\mathbb{R}^d)$ denotes the C^σ -functions with compact support.

In Proposition 3 we show the positive definiteness of the interpolation matrix $\begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$. For distributions $\lambda \in \mathcal{S}'(\mathbb{R}^d)$ we define as usual $\langle \check{\lambda}, \varphi \rangle := \langle \lambda, \check{\varphi} \rangle$ and $\langle \hat{\lambda}, \varphi \rangle := \langle \lambda, \hat{\varphi} \rangle$ with $\varphi \in C_0^\infty(\mathbb{R}^d)$, where $\check{\varphi}(x) = \varphi(-x)$ and $\hat{\varphi}(\omega) = \int_{\mathbb{R}^d} \varphi(x) e^{-ix^T \omega} dx$ denotes the Fourier transform. $\mathcal{E}'(\mathbb{R}^d)$ denotes the space of distributions with compact support.

Proposition 3 (Positive definiteness). *Let Ψ be as in Definition 1. Let $X_N = \{x_1, \dots, x_N\}$ and $\Xi_M = \{\xi_1, \dots, \xi_M\}$ be grids such that $f(x_i) \neq 0$ holds for all $i = 1, \dots, N$ and such that $x_i = x_j$ implies $i = j$ and $\xi_i = \xi_j$ implies $i = j$.*

Then the interpolation matrix $\begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$, cf. (5), is positive definite.

Proof. For $\lambda = \sum_{j=1}^N \beta_j (\delta_{x_j} \circ L) + \sum_{k=1}^M \gamma_k (\delta_{\xi_k} \circ L_0) \in \mathcal{F}^* \cap \mathcal{E}'(\mathbb{R}^d)$ we have

$$\begin{aligned} (\beta, \gamma) \begin{pmatrix} A & B \\ B^T & C \end{pmatrix} \begin{pmatrix} \beta \\ \gamma \end{pmatrix} &= \lambda^x \bar{\lambda}^y \Psi(x - y) = \|\lambda\|_{\mathcal{F}^*}^2 \\ &= (2\pi)^{-d} \int_{\mathbb{R}^d} |\hat{\lambda}(\omega)|^2 \hat{\Psi}(\omega) d\omega \geq 0 \end{aligned}$$

by (6). Hence, the matrix is positive semidefinite.

Now we show that $(\beta, \gamma) \begin{pmatrix} A & B \\ B^T & C \end{pmatrix} \begin{pmatrix} \beta \\ \gamma \end{pmatrix} = 0$ implies $\beta = 0$ and $\gamma = 0$.

If $(\beta, \gamma) \begin{pmatrix} A & B \\ B^T & C \end{pmatrix} \begin{pmatrix} \beta \\ \gamma \end{pmatrix} = (2\pi)^{-d} \int_{\mathbb{R}^d} |\hat{\lambda}(\omega)|^2 \hat{\Psi}(\omega) d\omega = 0$, then $\hat{\lambda}(\omega) = 0$ for all $\omega \in \mathbb{R}^d$; note that $\hat{\lambda}(\omega)$ is an analytic function and $\hat{\Psi}(\omega) > 0$ holds for all $\omega \in \mathbb{R}^d$ by (6). By Fourier transformation in $\mathcal{S}'(\mathbb{R}^d)$ we have $\mathcal{S}'(\mathbb{R}^d) \ni \lambda = 0$, i.e.

$$\lambda(h) = \sum_{j=1}^N \beta_j \langle \nabla h(x_j), f(x_j) \rangle + \sum_{k=1}^M \gamma_k h(\xi_k) = 0 \quad (7)$$

for all test functions $h \in \mathcal{S}(\mathbb{R}^d)$. Fix a $j \in \{1, \dots, N\}$. Either there is a point $\xi_{j^*} = x_j$ with $j^* \in \{1, \dots, M\}$; then there is a neighborhood $B_\delta(x_j) = \{x \in \mathbb{R}^d \mid \|x - x_j\| < \delta\}$ such that $x_i \notin B_\delta(x_j)$ holds for all $i \neq j$ and $\xi_i \notin B_\delta(x_j)$ holds for all $i \neq j^*$. Otherwise we can choose $B_\delta(x_j)$ such that $x_i \notin B_\delta(x_j)$ holds for all $i \neq j$ and $\xi_i \notin B_\delta(x_j)$ holds for all i . In both cases define the function $h(x) = \langle x - x_j, f(x_j) \rangle$ for $x \in B_{\frac{\delta}{2}}(x_j)$ and $h(x) = 0$ for $x \notin B_\delta(x_j)$, and extend it smoothly such that $h \in \mathcal{S}(\mathbb{R}^d)$. Then (7) yields in both cases

$$0 = \lambda(h) = \beta_j \|f(x_j)\|^2.$$

Since $f(x_j) \neq 0$, $\beta_j = 0$. This argumentation holds for all $j = 1, \dots, N$ and thus $\beta = 0$.

The argumentation for γ is similar. Hence, the matrix $\begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$ is positive definite.

Theorem 1 (Error estimate). *Assume, (1) has a solution $u \in C_0^\sigma(\mathbb{R}^d, \mathbb{R}^d)$, where $\mathbb{N} \ni \sigma \geq \sigma^* := \frac{d+1}{2} + k$ and $k \in \mathbb{N}$ denotes the parameter of the Wendland function $\psi_{l,k}(r)$, cf. [6], with $l := \lceil \frac{d}{2} \rceil + k + 1$, cf. Definition 1. Let $\Gamma \subset K \subset \Omega$ be a compact set.*

Then there are c^, c_0^* such that for all grids $X_N := \{x_1, \dots, x_N\} \subset K$ with fill distance h in K and $\Xi_M := \{\xi_1, \dots, \xi_M\} \subset \Gamma$ with fill distance h_0 in Γ such that*

$$|Ls(x) - Lu(x)| \leq c^* h^\kappa \text{ for all } x \in K, \quad (8)$$

$$|L_0s(x) - L_0u(x)| \leq c_0^* h_0 \text{ for all } x \in \Gamma \quad (9)$$

holds, where $\kappa = \frac{1}{2}$ for $k = 1$ and $\kappa = 1$ for $k \geq 2$ and $s \in C^{2k-1}(\mathbb{R}^d, \mathbb{R})$ is the reconstruction of u , cf. (3), with respect to the grids X_N, Ξ_M and $\Psi(x) = \psi_{l,k}(\mu\|x\|)$ with Wendland's function as in Definition 1.

Proof. We have $u, s \in \mathcal{F}$. For $x \in K$ let $x_j \in X_N$ be a grid point satisfying $\|x - x_j\| \leq h$. Set $\lambda = \delta_x \circ L \in \mathcal{F}^*$ and $\mu = \delta_{x_j} \circ L \in \mathcal{F}^*$. Then

$$\begin{aligned} |\lambda(s) - \lambda(u)| &= |(\lambda - \mu)(s - u)| \leq \|\lambda - \mu\|_{\mathcal{F}^*} \cdot \|s - u\|_{\mathcal{F}} \\ &\leq \|\lambda - \mu\|_{\mathcal{F}^*} \cdot \|u\|_{\mathcal{F}}. \end{aligned}$$

For the term $\|\lambda - \mu\|_{\mathcal{F}^*}^2 = (\lambda - \mu)^x (\lambda - \mu)^y \Psi(x - y)$ we use Taylor expansion. A similar argumentation holds for (9). For details cf. [4].

2 Application to Dynamical Systems

In this section we explain the meaning of the operator L and the solution u of (1) in the context of dynamical systems.

Consider the autonomous ordinary differential equation of first order with initial condition

$$\dot{x} = f(x), \quad x(0) = \xi \quad (10)$$

with $f \in C^\sigma(\mathbb{R}^d, \mathbb{R}^d)$ as in the last section. Since $\sigma \geq 1$, local existence and uniqueness of a solution $x(t)$ of (10) are guaranteed. A solution of (10) exists on a maximal time interval (T^-, T^+) with $T^- \in \mathbb{R}^- \cup \{-\infty\}$ and $T^+ \in \mathbb{R}^+ \cup \{\infty\}$. If $T^+ \neq \infty$, then $\lim_{t \nearrow T^+} |x(t)| = \infty$.

Furthermore, we assume that $f(0) = 0$ holds and that all eigenvalues of the Jacobian $Df(0)$ have negative real parts. Then 0 is an asymptotically stable equilibrium of (10), i.e. $x(t) = 0$ is a constant solution of (10) and, moreover, adjacent solutions exist for all $t \geq 0$, stay near 0 and tend to 0 as $t \rightarrow \infty$. Thus, we can define the basin of attraction of the equilibrium 0. In the following we seek to determine this set.

Definition 3 (Basin of attraction). *The basin of attraction of the asymptotically stable equilibrium 0 of (10) is defined by*

$$A(0) = \{\xi \in \mathbb{R}^d \mid \text{the solution } x(t) \text{ of (10) exists for all } t \geq 0 \\ \text{and } \lim_{t \rightarrow \infty} x(t) = 0\}.$$

$A(0)$ is a non-empty and open set.

Relation between the ODE (10) and the PDE (1).

The ODE $\dot{x} = f(x)$ and the PDE $Lu(x) = -c$ with $Lu(x) = \sum_{k=1}^d f_k(x) \frac{\partial u}{\partial x_k}(x)$ are linked in several ways. First of all, the ODE is the characteristic equation of the PDE and solutions of the ODE are characteristic curves of the PDE. In the following we will study the meaning of the PDE for the ODE. In particular, we will investigate the meaning of the operator L and prove the existence of a solution u of the PDE for the set $\Omega = A(0) \setminus \{0\}$. Moreover, the solution u of the PDE turns out to be a Lyapunov function, cf. Theorem 2.

Definition 4 (Orbital derivative). Let $u \in C^1(\mathbb{R}^d, \mathbb{R})$. Then $Lu(x) = \langle \nabla u(x), f(x) \rangle$, cf. (2), is called the orbital derivative of u with respect to (10).

The orbital derivative is the derivative of u along solutions of (10) since $\frac{d}{dt}u(x(t))\big|_{t=0} = \langle \nabla u(x(t)), \dot{x}(t) \rangle\big|_{t=0} \stackrel{(10)}{=} \langle \nabla u(\xi), f(\xi) \rangle = Lu(\xi)$. The solution u of (1) is thus decreasing along solutions at constant rate $-c$.

Note that the assumptions of the following Proposition 4 are satisfied, e.g., if Γ is the level set of a Lyapunov function within the basin of attraction, cf. Section 3. Solutions with initial value in $A(0)$ exist for all $t \geq 0$ by definition of $A(0)$.

Proposition 4. Let $\Omega = A(0) \setminus \{0\}$ and $\Gamma \subset \Omega$ such that for each $\xi \in \Omega$ there is one and only one $t \in \mathbb{R}$ such that $x(t) \in \Gamma$, where $x(t)$ is the solution of (10). Then (1) has a unique solution $u \in C^\sigma(\Omega, \mathbb{R})$.

Proof. The solution u of the non-characteristic Cauchy problem (1) is obtained by the method of characteristics: Define $u(\xi)$ for $\xi \in \Gamma$ by $u(\xi) = c_0$. Solutions $x(t)$ of (10) with $\xi \in \Gamma$ are characteristic curves and we set $u(x(t)) = u(\xi) - ct$. Hence, u is defined for all $x \in \Omega$ and is C^σ . For details cf. [4].

From the construction it is clear that $u(x)$ tends to $-\infty$ as $x \rightarrow 0$ and hence u is not defined in 0. We have proved existence, uniqueness and smoothness of the solution u of (1), but the proof does not serve to explicitly construct u , since the solution $x(t)$ of the characteristic equation (10) is not known in general. However, we can find an approximate solution using radial basis function, cf. Section 1.

Lyapunov Functions.

Functions with negative orbital derivative (not necessarily constant) are called Lyapunov functions and serve to determine the basin of attraction $A(0)$

through their level sets. Condition 3. in the following Theorem 2 means that K is bounded by a level set of v . For the proof of Theorem 2 one shows that the compact set K is positively invariant (solutions with initial value in K remain in K for all positive times) and thus in particular all solutions starting in K are defined for all $t \geq 0$.

Theorem 2 (Lyapunov functions). *Consider (10). Let $v \in C^1(\mathbb{R}^d, \mathbb{R})$ be a function and $K \subset \mathbb{R}^d$ be a compact set with neighborhood B such that*

1. $0 \in \overset{\circ}{K}$,
2. $Lv(x) < 0$ holds for all $x \in K \setminus \{0\}$,
3. $K = \{x \in B \mid v(x) \leq R\}$.

Then $K \subset A(0)$.

The key idea is that not only the function u satisfies $Lu(x) = -c < 0$, but also the reconstruction s satisfies $Ls(x) \leq Lu(x) + c^*h^\kappa = -c + c^*h^\kappa < 0$ for $h < \left(\frac{c}{c^*}\right)^{\frac{1}{\kappa}}$, i.e. if the grid is dense enough, by the error estimate (8). Hence, also the reconstruction s is a Lyapunov function and serves to determine the basin of attraction $A(0)$ through its level sets by Theorem 2.

However, we have problems near 0 and near $\partial A(0)$, since u is only defined in $\Omega = A(0) \setminus \{0\}$. Hence, the estimate (8) holds for any compact subset of $A(0) \setminus \{0\}$. The problem near 0 will be overcome by linearization, cf. Step 0.

3 Stepwise Calculation of the Basin of Attraction

Step 0.

We start with a Lyapunov function s_0 which is a Lyapunov function for the linear system $\dot{x} = Df(0)x$, i.e. the linearization of (10) at 0. The function s_0 will turn out to be a Lyapunov function for the nonlinear system (10) in some neighborhood $B_{R_0}^{s_0}$ of 0.

Lemma 1. *The matrix equation*

$$Df(0)^T P + PDf(0) = -I$$

has a unique solution $P \in \mathbb{R}^{d \times d}$, which is symmetric and positive definite. Define $s_0(x) = x^T P x$. Then there is an $R_0 > 0$ such that

$$Ls_0(x) < 0 \quad \text{holds for } x \in B_{R_0}^{s_0} \setminus \{0\},$$

where $B_{R_0}^{s_0} := \{x \in \mathbb{R}^d \mid s_0(x) < R_0\}$.

By Theorem 2 we have $B_0 := B_{R_0}^{s_0} \subset A(0)$. We proceed with the next step.

Step n, n ≥ 1.

Assume that s_{n-1} is a function and B_{n-1} an open, bounded set with $LS_{n-1}(x) < 0$ for all $x \in B_{n-1} \setminus \{0\}$. Set $\Gamma_n := \partial B_{n-1}$. Choose the compact set \tilde{K} such that $\Gamma_n \subset \tilde{K} \subset \tilde{K} \subset A(0) \setminus \{0\}$. In practical applications, $A(0)$ is not known a priori and thus it is not possible to show $\tilde{K} \subset A(0) \setminus \{0\}$ a priori. Determine the reconstruction $s = \tilde{s}_n$ of the function u with any $c > 0$ and $c_0 = 1$. If the grid X_N has a fill distance such that $h < (\frac{c}{c^*})^{\frac{1}{\kappa}}$ holds, then $L\tilde{s}_n(x) < 0$ holds for all $x \in \tilde{K}$. \tilde{s}_n and s_{n-1} can be glued together to a function s_n using a continuation. For the function s_n we have $LS_n(x) < 0$ for all $x \in (\tilde{K} \cup B_{n-1}) \setminus \{0\}$. Moreover, level sets of \tilde{s}_n are level sets of s_n . This is proved by a partition of unity, cf. [4]. Since the level sets of s_n are also level sets of \tilde{s}_n , there is no need to compute the function s_n in examples. If the fill distance h_0 is small enough, one can find a constant R_n such that $B_{n-1} \subset B_n := B_{R_n}^{s_n} \subset \tilde{K} \cup B_{n-1}$ holds. Hence, $B_n \subset A(0)$ holds by Theorem 2.

With Theorem 1 one can show that the method works if \tilde{K} , h and h_0 are chosen properly. One can even obtain each compact subset of the basin of attraction by this method, provided that $\sup_{x \in A(0)} \|f(x)\| < \infty$ holds; for details cf. [4]. The latter condition can easily be satisfied by studying an equivalent system.

Example.

As an example we apply the method to the ODE

$$\begin{cases} \dot{x} = x(-1 + 4x^2 + \frac{1}{4}y^2) + \frac{1}{8}y^3 \\ \dot{y} = y(-1 + \frac{5}{2}x^2 + \frac{3}{8}y^2) - 6x^3 \end{cases}$$

Step 0: We have $P = \frac{1}{2}I$, $s_0(x) = \frac{1}{2}\|x\|^2$ and $R_0 = 0.045$, cf. Figure 1, right. Thus, we obtain a subset B_0 of the basin of attraction $A(0)$.

Step 1: We solve $Lu(x) = -1$ using the radial basis function $\Psi(x) = \psi_{4,2}(1.5\|x\|)$ and choose a hexagonal grid X_N with $N = 70$ points and a grid Ξ_M with $M = 10$ points, cf. Figure 2, left. In this step, the approximation s_1 satisfies $LS_1(x) < 0$ near $x = 0$ and a continuation is not necessary. We choose $R_1 = 1.7$, cf. Figure 2, right. Thus, we obtain a subset B_1 of the basin of attraction $A(0)$.

Step 2: We solve $Lu(x) = -1$ using the radial basis function $\Psi(x) = \psi_{4,2}(1.7\|x\|)$ and choose a hexagonal grid plus two additional points with $N = 132$ points altogether. Moreover, we choose a grid Ξ_M with $M = 20$ points, cf. Figure 3, left. In this step, the approximation \tilde{s}_2 does not satisfy $L\tilde{s}_2(x) < 0$ near $x = 0$ and a continuation is necessary. However, since level sets of \tilde{s}_2 and s_2 are the same we do not need to calculate the continuation s_2 but we rather use the level sets of \tilde{s}_2 of level $R_2 = 1.5$, cf. Figure 3, right.

Thus, we obtain a subset B_2 of the basin of attraction $A(0)$. Note that the level set $\tilde{s}_2(x) = 1$ is different from I_2 . By construction, however, $\tilde{s}_2(x) = 1$ holds for all points $x \in \Xi_M$.

In Figure 1, left we compare the three subsets B_0 , B_1 and B_2 with the numerically calculated basin of attraction $A(0)$, the boundary of which is an unstable periodic orbit.

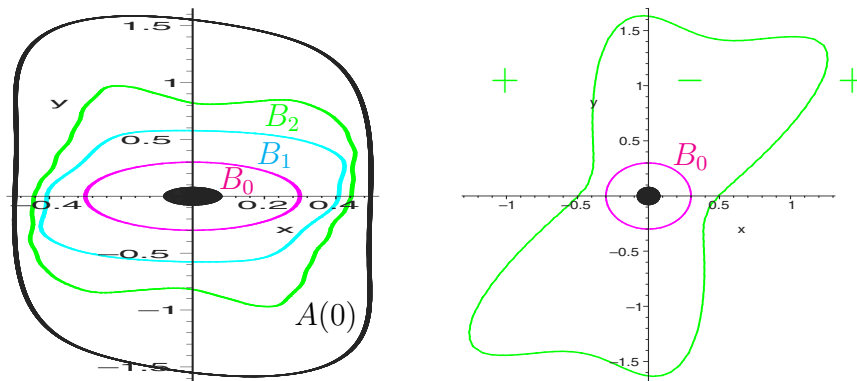


Fig. 1. Left: Comparison of the subsets B_0 , B_1 and B_2 obtained in the respective steps of the method with the numerically calculated basin of attraction $A(0)$ (black), the boundary of which is an unstable periodic orbit in this example, cf. (11). Right: the zeroth step with the quadratic Lyapunov function $s_0(x)$ of Lemma 1. The figure shows the sign of $s_0'(x)$ and the set $B_0 = \{x \in \mathbb{R}^2 \mid s_0(x) < R_0\}$ with $R_0 = 0.045$.

References

1. C. Franke and R. Schaback: Convergence order estimates of meshless collocation methods using radial basis functions. *Adv. Comp. Math.* **8**, 1998, 381–399.
2. C. Franke and R. Schaback: Solving partial differential equations by collocation using radial basis functions. *Appl. Math. Comp.* **93**(1), 1998, 73–82.
3. P. Giesl: Approximation of domains of attraction and Lyapunov functions using radial basis functions. In: *Proceedings of the NOLCOS 2004 Conference*, Stuttgart, Germany, vol. II, 2004, 865–870.
4. P. Giesl: *Construction of Lyapunov Functions Using Radial Basis Functions*. Habilitation Thesis, Munich University of Technology, 2005.
5. A. Iske: Reconstruction of functions from generalized Hermite-Birkhoff data. In: *Approximation Theory VIII*, C.K. Chui, L.L. Schumaker (eds.), 1995, 257–264.
6. H. Wendland: Error estimates for interpolation by compactly supported radial basis functions of minimal degree. *J. Approx. Theory* **93**, 1998, 258–272.
7. Z. Wu: Hermite-Birkhoff interpolation of scattered data by radial basis functions. *Approx. Theory Appl.* **8**, 1995, 283–292.

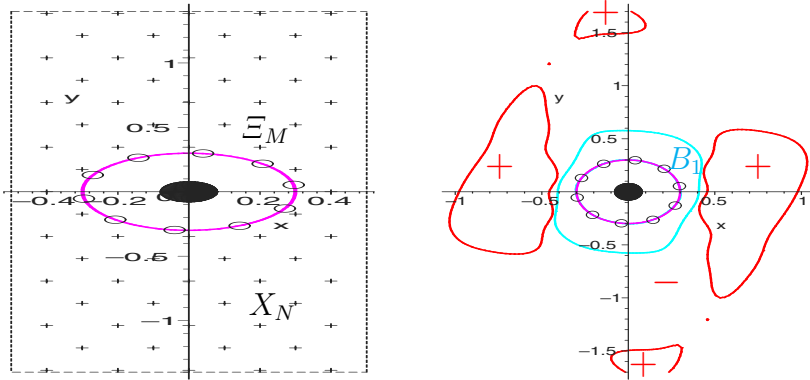


Fig. 2. The first step with the Lyapunov function s_1 . Left: the grid X_N (+) in the set \tilde{K} bounded by the rectangle (dotted line), the grid Ξ_M (o) in the set Γ_1 , which is the boundary of B_0 . Right: the set B_0 , the grid Ξ_M (o) which is on ∂B_0 by construction, the sign of $s_1'(x)$ and the set $B_1 = \{x \in \mathbb{R}^2 \mid s_1(x) < R_1\}$ with $R_1 = 1.7$ as well as the level set $s_1(x) = 1$. Note that the sign of $s_1'(x)$ is negative in $B_1 \setminus \{0\}$.

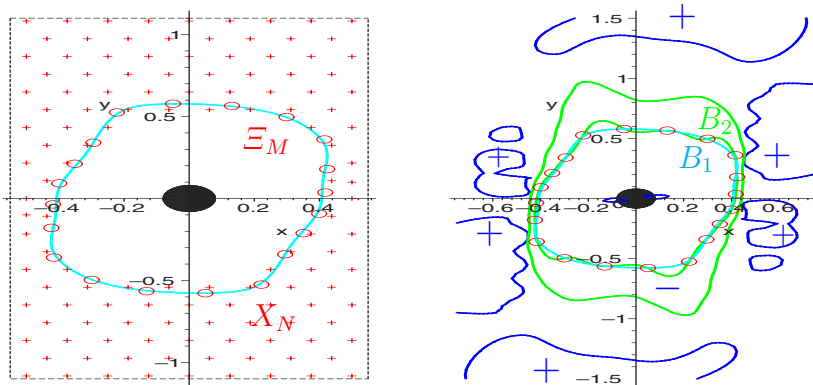


Fig. 3. The second step with the Lyapunov function s_2 . Left: the new grid X_N (+) in the new set \tilde{K} bounded by the rectangle (dotted line), the new grid Ξ_M (o) in the set Γ_2 , which is the boundary of B_1 . Right: the set B_1 , the grid Ξ_M (o) which is on ∂B_1 by construction, the sign of $s_2'(x)$ and the set $B_2 = \{x \in \mathbb{R}^2 \mid \tilde{s}_2(x) < R_2\}$ with $R_2 = 1.5$ as well as the level set $\tilde{s}_2(x) = 1$. Note that the sign of $\tilde{s}_2'(x)$ is positive near the origin. Hence, in this case we use the continuation s_2 of \tilde{s}_2 . However, since the signs of $s_2'(x)$ and $\tilde{s}_2'(x)$ are equal outside B_1 and the level sets of \tilde{s}_2 and s_2 coincide, $B_2 = \{x \in \mathbb{R}^2 \mid \tilde{s}_2(x) < R_2\} = \{x \in \mathbb{R}^2 \mid s_2(x) < R_2^*\}$ with a suitable constant R_2^* .

Integro-Differential Equation Models and Numerical Methods for Cell Motility and Alignment

Athena Makroglou

Department of Mathematics, University of Portsmouth, Portsmouth, Hampshire
PO1 3HF, UK, athena.makroglou@port.ac.uk

Summary. Integro-differential equations have been used in a number of areas of cell biology, like cells cycles, cell growth, cell motility. This article is a short review of some of the models which have been used in the literature to model cell motility and cell orientation and alignment with the emphasis on integro-differential equation models. It presents several such models in the form of ordinary or partial integro-differential equations, together with some information about the numerical methods used in the original papers. It also describes a numerical method which was used in this paper for obtaining some computational results for an alignment model.

1 Introduction

Cell motility (ability of cells to move) models are important to study since the development of cells, tissues and organs depends on cell motility. Examples include [26] movement of cells to the ‘right’ place during embryonic development, movement of white cells (neutrophils, leukocytes) to the site of infection (immune response to bacterial invasion), wound healing (epidermal cells (fibroblasts, keratocytes) move where the wound is). Movement of cells happens for the wrong causes too. Examples include angiogenesis and cancer metastasis.

Different types of cells move in a number of ways. Ionides et al. [29] and also Dickinson [17] give a classification of cell motion with respect to length and time scales (scales of locomotion, translocation, migration).

The individual cell movement along a substrate on a locomotion scale is in general a four step process (cf. [11, 26, 31]). A figure that demonstrates nicely the four stages of cell movement may be found at

www.ncbi.nlm.nih.gov/books/bv.fcgi?rid=mcb.figgrp.5251.

Alignment is found in actin filaments (rod-like polymers, the main building part of the cyto-skeleton), in fibroblasts (part of the connective tissue involved in wound healing), in mycobacteria (they form streets in which all cells have

the same orientation and move forward or backwards). We also speak of speed alignment (adaptation with respect to speed).

A wealth of information about the cytoskeleton and cell motility and much more can be found at the Biochemistry and Cell Biology Virtual Library web page www.biochemweb.org/cytoskeleton.shtml. An illuminating article for crawling cell mechanisms written for a more general audience, is for example one by Thomas P. Stossel [53]. For more information about cell movement we refer for example to the books [5, 7] and to the articles [3, 28].

The types of equations which have been used in modeling cell motility and alignment and related problems, include partial differential equations (PDEs), cf. [1, 6, 9, 10, 26, 27, 30, 35, 38, 39, 42, 43, 45, 47, 51, 52, 55], integro-differential equations (IDEs) and partial integro-differential equations (PIDEs), and systems of such equations, cf. [4] and the references of Sections 2 and 3. Some ordinary differential equations (ODEs), cf. [19, 22, 32, 41, 46, 54] and stochastic differential equations (SDEs), cf. [29] – on the scale of translocation and migration – have also been used.

This article presents cell motility and alignment models in the form of integro-differential equations. The form of the equations of the models is given, together with some details about the numerical methods used in the original papers (Sections 2-3 for cell motility and cell alignment models respectively). In addition, one numerical method is implemented to obtain some computational results for a cell alignment model by [24], in the form of a PIDE (Section 4). Some directions for further work may be found in Section 5.

The notation of the original papers is kept for easy reference to the corresponding equations there. The notation $K * L$, (unless otherwise defined in subsequent sections) denotes the convolution integral

$$K * L = \int_{-\pi}^{\pi} K(\theta - \theta')L(\theta', t) d\theta'.$$

2 Cell Motility Integro-Differential Equation Models

Papers that have presented integro-differential equations include: [40] (for modeling force-velocity relation for growing microtubules using a PIDE), [20] (it models the length distribution of the actin-filament in a lamellipod, using PIDEs and IDEs). We also mention the paper by Novak, Slepchenko, Mogilner and Loew ([44]) which presents a model in the form of two PDEs and one integro-differential equation which is used for explaining why the focal adhesions tend to high-curvature regions at the cell periphery for stationary cells, since it can be extended to more complex processes in moving cells, too [44].

In Subsection 2.1 we present the Mogilner and Oster model [40] and in Subsection 2.2 the Edelstein-Keshet and Ermentrout model [20].

2.1 The Mogilner and Oster Model

The authors [40] present a model for $n(x, t)$ the continuous special density of microtubule filament tips at position x and at time t in the form of the following integral-differential-difference equations.

$$\begin{aligned} \frac{\partial n}{\partial t} = & k_{\text{on}}(n(x + \delta) - n(x)) + k_{\text{off}}(n(x - \delta) - n(x)) \\ & + \int_0^\delta p(f, y)n(y)n(x + y - \delta)dy - n(x) \int_0^\delta p(f, y)n(y)dy, \quad x \geq \delta, \end{aligned} \quad (1)$$

$$\begin{aligned} \frac{\partial n}{\partial t} = & k_{\text{on}}n(x + \delta) - k_{\text{off}}n(x) + \int_{\delta-x}^\delta p(f, y)n(y)n(x + y - \delta)dy \\ & - n(x) \int_0^\delta p(f, y)n(y)dy, \quad 0 \leq x < \delta, \end{aligned} \quad (2)$$

where $p(f, y) = k_{\text{on}} \exp[f(y - \delta)/k_B T]$, f is the load force, k_B is the Boltzmann constant, T is the absolute temperature, $\delta = 8nm$ is the size of a tubulin dimer. $k_{\text{on}}, k_{\text{off}}$ are rate parameters indicating assembly and disassembly of tubulin dimers onto the protofilament tips respectively.

Numerical Methods.

See [40, p. 241]. Equations (1)-(2) were transformed to dimensionless form and solved on the interval $0 < x < 6\delta$. $78 = 6 \times 13$ mesh points were used. The integrals were evaluated by the trapezoidal method. The equations were integrated using the forward Euler method with Matlab. Uniform initial conditions and no flux boundary conditions were used.

2.2 The Edelstein-Keshet and Ermentrout Model

The authors [20] give PIDEs for $b_a(x, l, t)$, $b_c(x, l, t)$, the density of active barbed ends and capped barbed ends respectively, at position x and time t with filament of length l attached to them. a stands for $a(x, t)$ the concentration of the actin monomers at position x and time t . Under several simplifying assumptions and introducing a new variable ξ relating to t and x , they derive an IDE in $b_c(\xi, l, t)$. They set $\frac{\partial b_c}{\partial t} = 0$ to find stationary solutions which obey the following IDE in the stationary density distribution $B(\xi, t)$ [20],

$$\begin{aligned} v_b \frac{\partial B}{\partial \xi} = & v_p \frac{\partial B}{\partial l} + gP(\xi)B_a(\xi + l) + g \int_0^\xi B(y, \xi - y + l)P(\xi - y)dy \\ & + gP(l) \int_l^\infty B(\xi, l')dl' - gB(\xi, l) \int_0^l P(l')dl', \quad B(0, l) = 0, \end{aligned}$$

where g is the concentration of actin filament ‘chopper’, and $P(t)$ is the filament cutting probability at distance l from an active barbed end, v_b, v_p are

the apparent rates of motion of the barbed, respectively the capped, end of a filament. The suggested choices of $P(l)$ are: $P(l) = p = \text{const}$, $P(l) = pl$, $P(l) = 1 - \exp(-rl)$, where $r = 0.011$.

Numerical Methods.

See [20, p. 345]. The above PIDE was solved numerically in the paper by discretizing first w.r.t. l and then solving a system of IDEs in ξ . The software package XPPAUT with Euler's method and stepsize 0.1. XPPAUT is available from www.math.pitt.edu/~bard/xpp/xpp.html. It is suitable for solving differential equations, difference equations, delay equations, functional equations, boundary value problems, and stochastic equations. Its use is now explained in the book Ermentrout [21].

3 Cell Alignment Integro-Differential Models

The equations are in the form of PIDEs usually. Related papers include: [12, 24, 25, 34, 37, 48, 50] (for actin structures), [18, 36, 37] (whole cell (fibroblasts) structure), [15] (fibroblast and collagen orientation), [13] (endothelial cells), [16] (extracellular matrix alignment of skin and connective tissue), [33] (alignment and movement combined). In Subsections 3.1-3.6 the equations of the models used and some numerical details are given for at least one paper from the above categories; the presentation is in chronological order.

3.1 The Civelekoglu, Edelstein-Keshet Model

One of the models introduced in [12], is concerned with the dynamics of actin filaments in the cell. It has the form of two PIDEs for $L(\theta, t)$ and $B(\theta, t)$, the concentration of free and bound actin filaments respectively, at orientation θ and at time t .

The Model Equations.

See [12, p. 595].

$$\begin{aligned} \frac{\partial L}{\partial t}(\theta, t) &= \mu \frac{\partial L^2}{\partial \theta^2} - \gamma L + \alpha A L + \delta B - \beta \rho L(K * B) - \beta \rho L(K * L), \\ \frac{\partial B}{\partial t}(\theta, t) &= -\gamma B + \alpha A B - \delta B + \beta \rho B(K * L) + \beta \rho L(K * L), \quad -\pi \leq \theta \leq \pi, \end{aligned}$$

where $A(t)$ denotes the density of actin monomers at time t , μ denotes the rotational diffusion constant of F-actin, $\rho(t)$ is the unbound actin binding protein concentration, δ is the dissociation rate of the binding proteins and β is the affinity of the binding. $K(\phi)$ is the probability that a filament contacting another filament at a relative angle ϕ binds to it in the presence of actin binding proteins. Two different types of kernels $K(\phi)$ were considered, see [12, pp. 593-594] for their form. All functions of θ are assumed to be periodic.

Numerical Methods.

See [12, p. 598]. The equations were discretized with respect to the θ variable on a grid of 30-36 points with $\Delta\theta = 360/30 = 12^\circ$ or $\Delta\theta = 360/36 = 10^\circ$ and then a forward finite difference scheme was used with respect to time with $\Delta t = 0.01$.

Initial functions (see [12, p. 598]). A variety of initial densities which included random, or sinusoidal deviations from the steady-state, or from a random homogeneous density. The magnitude of these deviations was reported to be equal to about 10% of the initial homogeneous densities.

3.2 The Spiros, Edelstein-Keshet Model

The paper [50] presents a PIDE model for actin filament interactions which is based on models presented in [12, 18] and [37] and it is concerned also with estimation of parameters.

The Model Equations.

See [50, p. 278].

$$\begin{aligned}\frac{\partial N}{\partial t}(x, \theta, t) &= \beta_1 F(K * F) + \beta_2 N(K * F) - \gamma N \\ \frac{\partial F}{\partial t}(x, \theta, t) &= -\beta_1 F(K * F) - \beta_2 F(K * N) + \gamma N + \mu_1 \frac{\partial^2 F}{\partial \theta^2} + \mu_2 \frac{\partial^2 F}{\partial x^2},\end{aligned}$$

where

$$\begin{aligned}K * F &= \int_{-\pi}^{\pi} \int_{\Omega} K(\theta - \theta', x - x') F(x', \theta') d\theta' dx'. \\ K(\theta, x) &= K_1(\theta) K_2(x), \quad K_i(u) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-\frac{u^2}{2\sigma_i^2}\right), i = 1, 2.\end{aligned}$$

L is the average length of an actin filament, $N(x, \theta, t)$ is the number density of network (i.e., bound) filaments at x, θ and at time t , $F(x, \theta, t)$ is the number density of free filaments at x, θ and time t , $\mu_1, \mu_2, \beta_1, \beta_2, \gamma$, are rate constants, see [50, p. 276] for details.

Numerical Methods.

See [50, p. 292]. The evaluation of the convolution integrals was done by using Fourier transforms and then the inverse fast Fourier transform (IFFT). An explicit fourth-order Runge-Kutta method was used to solve the system of partial differential equations. Periodic boundary conditions in both the spatial and the angular variable were used.

Initial functions. The initial actin distribution was taken to be a 10% random deviation from the uniform steady-state situation, see [50, p. 293].

3.3 The Geigant, Ladizhansky, Mogilner Model

The authors [24] consider a PIDE model for the angular order of the actin. The unknown function is $f(\theta, t)$, the mean density function of the filaments with orientation angle θ at time t . It has the form

$$\begin{aligned} \frac{\partial f}{\partial t} = & -f(\theta, t) \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \eta(\theta - \theta_i) \omega(\theta - \theta_n, \theta - \theta_i) f(\theta_i, t) d\theta_i d\theta_n \\ & + \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \omega(\theta_0 - \theta, \theta_0 - \theta_i) \eta(\theta_0 - \theta_i) f(\theta_0, t) f(\theta_i, t) d\theta_i d\theta_0, \end{aligned} \quad (3)$$

with denoting

- $f(\theta, t)$: angular distribution of filaments,
- $\eta(\theta_0 - \theta_i)$: rate per unit time of interaction between two filaments at directions θ_0, θ_i ,
- $\omega(\theta_0 - \theta_n, \theta_0 - \theta_i)$: probability of turning of a filament from direction θ_0 to direction θ_n as a result of interactions with filaments at direction θ_i ,

where the functions are 2π -periodic in all variables. The form of $\omega(\theta_1, \theta_2)$ is $\omega(\theta_1, \theta_2) = g_\sigma(\theta_1 - v(\theta_2))$, where $g_\sigma(\theta)$ is the periodic Gaussian or a step function, given in [24] respectively as

$$g_\sigma(\theta) = \frac{1}{\sqrt{(2\pi)\sigma}} \sum_{z \in \mathbb{Z}} \exp\left(-\frac{1}{2} \left(\frac{\theta + 2\pi z}{\sigma}\right)^2\right), \theta \in (-\pi, \pi) \quad (4)$$

and

$$g_\sigma(\theta) = \begin{cases} \frac{1}{2\sigma}, & |\theta| < \sigma (\leq \pi), \\ 0, & \sigma \leq |\theta| \leq \pi. \end{cases}$$

One choice of η is: $\eta = \frac{1}{2\pi}$. Choices of $v(\theta)$ included

$$v(\theta) = \kappa \sin \theta, v(\theta) = \kappa \theta, v(\theta) = \frac{\kappa}{2} \sin 2\theta.$$

Numerical Methods.

In [24], the equations were discretized with respect to θ to obtain a system of n differential equations which were solved by an Euler scheme (see [24, p. 799]). In [25], the integro-differential equation was solved numerically by use of Fourier transforms which resulted in a system of ordinary differential equations in the Fourier transforms. A standard Runge-Kutta method with variable time steps was applied to solve the ODE system (see [25, Appendix A]).

Initial functions. A randomly chosen periodic continuous distribution ([24, p. 799]), such that $\int_{-\pi}^{\pi} f(\theta, 0) d\theta = 1$.

3.4 The Dallon-Sherratt Models

The authors [15] developed a model for fibroblast and collagen orientation ‘with the ultimate objective of understanding how fibroblasts form and remodel the extracellular matrix, in particular its collagen component’. The paper [16] introduces spatial variation, too.

The Model Equations.

See [15, p. 105], equations in dimensionless form.

$$\begin{aligned}\frac{\partial f}{\partial t} &= \frac{\partial}{\partial \theta} \left(D \frac{\partial f}{\partial \theta} - f \frac{\partial}{\partial \theta} (W_1 * c) \right), \theta \in [0, 2\pi], \\ \frac{\partial c}{\partial t} &= -\alpha \frac{\partial}{\partial \theta} \left(c(\theta) (W_2 * f)(\theta) \frac{\partial}{\partial \theta} (W_3 * f)(\theta) \right), \theta \in [0, \pi],\end{aligned}$$

with boundary conditions periodic in θ , see [15, p. 104] for particular details. $f(\tau, \theta), \theta \in [0, 2\pi]$ and $c(\tau, \theta), \theta \in [0, \pi]$ are the densities of fibroblasts and collagen fibers respectively at time τ , oriented at an angle θ with respect to some arbitrary reference direction. W_2, W_3 are 2π periodic and W_1 is π periodic. They also obey a normalization condition. Choices of $W_i(\theta)$ may be found in [15, pp. 105-106]. In [16] the model was extended to include spatial variation (see [16, p. 509] for more information).

Numerical Methods.

The convolutions were calculated by using a left hand rectangle rule. The partial differential equations in [15] were solved using a Crank-Nikolson method ([15, p. 110]). The spatial flux in [16] was discretized with upwinding; the Lax-Wendroff method was also tried (see [16, p. 510] for more information).

3.5 The Lutscher Model

The Lutscher models allow for both movement and alignment. They are based on reaction transport equations in one and two dimensions. Several PIDEs have been given in the paper, some within proofs of theorems. The ‘full alignment transport equation’ is [33, p. 249], eq. (27),

$$u_t + s \cdot \nabla_x u = -\mu_*(u - K * u)(t, x, s) + A(u)(t, x, s),$$

with periodic boundary conditions, where $u(t, x, s)$ is the density of particles at position $x \in \Omega \subset \mathbb{R}^n$ with velocity $s \in V \subset \mathbb{R}^n$. μ_* is the turning rate, $K(s, s')$ is a kernel function according to which the particles choose a new direction s' . The function $A(u)$ gives the net rate of change in direction s . Ω is assumed either equal to \mathbb{R}^2 , or equal to $[0, 1]^2$. Simulations were performed for some of the models, using an explicit forward time and backward space scheme, cf. [33, p. 244].

3.6 The Civelekoglu-Scholey Model

The authors have introduced a model for the coupled dynamics of (endothelial) cell adhesions, small GTPases Rac and Rho and actin stress fibers (parallel actin filaments) in the form of a system of ordinary differential equations ([13, p. 576]) and a model for stress fiber alignment with each other and with the long axis of the cell ([13, p. 577]) in the form of a system of PIDEs. The PIDEs are:

$$\begin{aligned} \frac{\partial n}{\partial t}(\theta, t) = & -\frac{1}{2}(I * n)n(\theta, t) + (1 - r_1)(n * n)(\theta, t) \\ & + n_0 - r_2 n(\theta, t) + \gamma m(\theta, t) - r_3(I * m)n(\theta, t) \end{aligned} \quad (5)$$

$$\begin{aligned} \frac{\partial m}{\partial t}(\theta, t) = & r_4 \left[-\frac{1}{2}(I * m)m(\theta, t) + (m * m)(\theta, t) \right] - r_5 m(\theta, t) \\ & - \gamma m(\theta, t) + r_1(n * n)(\theta, t) + r_3(I * n)m(\theta, t), \end{aligned} \quad (6)$$

for $\theta \in [-\pi/2, \pi/2]$, where $n(\theta, t)$ and $m(\theta, t)$ are the angular densities of F-actin contained in the nascent and mature fibres respectively,

$$(I * n) = \frac{2}{\pi} \int_{-\pi/2}^{\pi/2} n(\phi) d\phi, \quad (n * n)(\theta) = \frac{2}{\pi} \int_{-\pi/4}^{\pi/4} n(\theta + \phi)n(\theta - \phi) d\phi,$$

and the rest of the convolutions defined similarly. n_0 is the stress fiber nucleation rate, γ is the rate of mature stress fibers fragmentation, r_1, \dots, r_5 are more rate constants (see Table 5 in [13, p. 578] for their meaning). A flat 2-d ellipsoidal cell domain is chosen.

Numerical Methods.

See [13, p. 579]. The authors discretized the interval $-\pi \leq \theta \leq \pi/2$ (in radians) using spatial step equal to 0.05 and solved the corresponding ODE system using an explicit Euler method. For the evaluation of the integrals, a composite midpoint rule was used. Constant initial conditions perturbed ‘weakly and randomly’ were used.

4 Some Computational Results

The integro-differential equation (3) of the Geigant, Ladizhansky and Mogilner model [24] of Subsection 3.3 is considered. Following [24], the $[-\pi, \pi]$ interval is transformed to $[0, 1]$ giving

$$\begin{aligned} \frac{\partial \tilde{f}}{\partial t}(\tilde{\theta}, t) = & -4\pi^2 \tilde{f}(\tilde{\theta}, t) \int_0^1 \int_0^1 \tilde{\eta}(\tilde{\theta} - \tilde{\phi}) \tilde{\omega}(\tilde{\theta} - \tilde{x}, \tilde{\theta} - \tilde{\phi}) \tilde{f}(\tilde{\phi}, t) d\tilde{\phi} d\tilde{x} \\ & + 4\pi^2 \int_0^1 \int_0^1 \tilde{\eta}(\tilde{s} - \tilde{\phi}) \tilde{\omega}(\tilde{s} - \tilde{\theta}, \tilde{s} - \tilde{\phi}) \tilde{f}(\tilde{\phi}, t) \tilde{f}(\tilde{s}, t) d\tilde{\phi} d\tilde{s} \end{aligned} \quad (7)$$

where we have set

$$\tilde{\eta}(x) = \eta(2\pi x), \tilde{\omega}(x, y) = \omega(2\pi x, 2\pi y), \tilde{f}(\tilde{\theta}, t) = f(-\pi + 2\pi\tilde{\theta}, t),$$

and have replaced the $\theta_0, \theta_i, \theta_n$ in (3) by s, ϕ, x , respectively.

A simple way to solve equation (7) numerically is to discretize with respect to $\tilde{\theta}, 0 \leq \tilde{\theta} \leq 1$, replacing $\tilde{\theta}$ by $\tilde{\theta}_\lambda, \lambda = 0, \dots, n$, and approximate the integrals by a quadrature rule (say trapezoidal). Then, a system of $n + 1$ ordinary differential equations is obtained in $y_\lambda(t) = \tilde{f}(\tilde{\theta}_\lambda, t), \lambda = 0, 1, \dots, n$.

$$\begin{aligned} \frac{dy_\lambda(t)}{dt} = & -4\pi^2 h^2 y_\lambda(t) \sum_{i=0}^n \sum_{j=0}^n w_i w_j \tilde{\eta}(\tilde{\theta}_\lambda - \tilde{\theta}_j) \tilde{\omega}(\tilde{\theta}_\lambda - \tilde{\theta}_i, \tilde{\theta}_\lambda - \tilde{\theta}_j) y_j(t) \\ & + 4\pi^2 h^2 \sum_{i=0}^n \sum_{j=0}^n w_i w_j \tilde{\eta}(\tilde{\theta}_i - \tilde{\theta}_j) \tilde{\omega}(\tilde{\theta}_i - \tilde{\theta}_\lambda, \tilde{\theta}_i - \tilde{\theta}_j) y_i(t) y_j(t), \end{aligned} \quad (8)$$

where $w_i, i = 0, 1, \dots, n$ are the weights of the trapezoidal rule.

Some computational results have been obtained using the Matlab function ODE45 for solving the ODE system (8). The form of the function $g_\sigma(\theta)$ used in the definition of $\omega(x_1, x_2)$ is given by (4), $\eta(x) = \frac{1}{2\pi}$. The form of the $v(x)$ functions is given on the graphs. The von Mises distribution (cf. [2, 14, 23]) with values multiplied by 2π was used for $y_\lambda(0)$. The Matlab function `von_mises_pdf(x, a, b)` by John Burkardt, see

www.scs.fsu.edu/~burkardt/m_src/prob/,

with $a = -\frac{2\pi}{3}$ and $b = 0.5$ was used for the simulations (it uses the function `bessel_i0(arg)`, also available from the same web page). The value of n used was $n = 50$. Two graphs are shown in next figure with $\theta \equiv \tilde{\theta}$.

Comparing with graphs of Figure 3 and Figure 4 of [24, pp. 800-801], respectively, we may note that the qualitative behaviour of the corresponding graphs is similar, but the peaks in Figure 1 here occur at larger density values. This might be due to the use of different initial function and to the use of more accurate ODE solver.

5 Further Work

Further work can be directed towards the computational treatment of some of the IDEs of Sections 2 and 3, extending for example [8] and to comparisons with the results of the corresponding papers.

There is also a wealth of software packages addressing Cell Biology problems, some of which are publicly available (The Virtual Cell, www.vcell.org/), see also www.ccbsymposium.org/software.html for more. It will be interesting to try to use some of these for solving IDEs applying to cell motility.

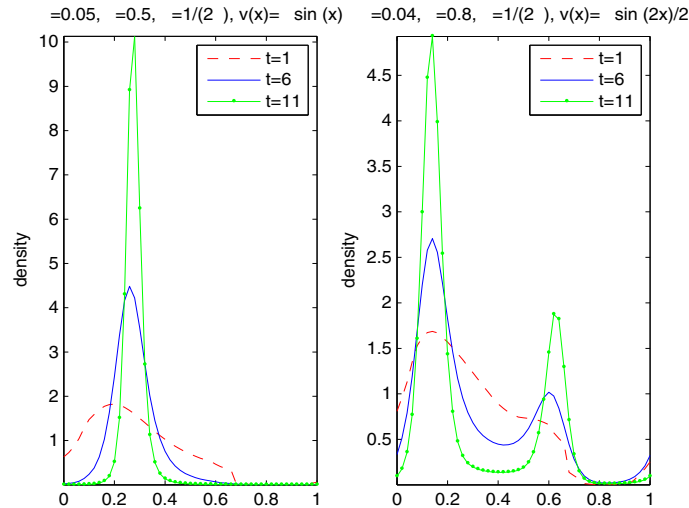


Fig. 1. $\tilde{f}(\theta, t)$, model [24]

References

1. W. Alt and M. Dembo: Cytoplasm dynamics and cell motion: two-phase flow models, *Math. Biosciences* **156**, 1999, 207–228.
2. E. Batschelet: *Circular Statistics in Biology*, Academic Press, 1981.
3. J. Bereiter-Hahn: Mechanics of crawling cells, *Medical Engineering and Physics* **27**, 2005, 743–753.
4. P.C. Bressloff: Euclidean shift-twist symmetry in population models of self-aligning objects, *SIAM J. Appl. Math.* **64**, 2004, 1668–1690.
5. D. Boal: *Mechanics of the Cell*, Cambridge University Press, Cambridge, 2002.
6. D.C. Bottino and L.J. Fauci: A computational model of ameboid deformation and locomotion, *Eur. Biophys. J.* **27**, 1998, 532–539.
7. D. Bray: *Cell Movements: From Molecules to Motility*, Garland Publishing, New York, 2005.
8. H. Brunner, A. Makroglou and R.K. Miller: Mixed interpolation collocation methods for first and second order Volterra integro-differential equations with periodic solution, *Applied Numerical Mathematics* **23**(4), 1997, 381–402.
9. D.C. Bottino, A. Mogilner, T. Roberts, M. Stewart and G. Oster: How nematode sperm crawl, *J. Cell Sci.* **115**, 2002, 367–384.
10. Y.S. Choi, J. Lee and R. Lui: Travelling wave solutions for a one-dimensional crawling nematode sperm cell model, *J. Math. Biol.* **49**, 2004, 310–328.
11. Y.S. Choi and R. Lui: Existence of traveling wave solutions for a one-dimensional cell motility model, *Taiwanese J. Math.* **8**(3), 2004, 399–414.
12. G. Civelekoglu and L. Edestein-Keshet: Modelling the dynamics of F-actin in the cell, *Bull. Math. Biol.* **56**(4), 1994, 587–616.
13. G. Civelekoglu-Scholey, A.W. Orr, I. Novak, J.-J. Meister: Model of coupled transient changes of Rac, Rho, adhesions and stress fibers alignment in endothelial cells responding to shear stress, *J. Theor. Biol.* **232**, 2005, 569–585.

14. E.A. Codling and N.A. Hill: Calculating spatial statistics for velocity jump processes with experimentally observed reorientation parameters, *J. Math. Biol.* **51**, 2005, 527–556.
15. J.C. Dallon and J.A. Sherratt: A mathematical model for fibroblast and collagen orientation, *Bull. Math. Biol.* **60**, 1998, 101–129.
16. J.C. Dallon and J.A. Sherratt: A mathematical model for spatially varying extracellular matrix alignment, *SIAM J. Appl. Math.* **61**, 2000, 506–527.
17. R.B. Dickinson: A generalized transport model for biased cell migration in an anisotropic environment, *J. Math. Biol.* **40**, 2000, 97–135.
18. L. Edelstein-Keshet and G.B. Ermentrout: Models for contact-mediated pattern formation: cells that form parallel arrays, *J. Math. Biol.* **29**, 1990, 33–58.
19. L. Edelstein-Keshet and G.B. Ermentrout: Models for the length distribution of actin filaments: I. Simple polymerization and fragmentation, *Bull. Math. Biol.* **60**, 1998, 449–475.
20. L. Edelstein-Keshet and G.B. Ermentrout: A model for actin-filament length distribution in a lamellipod, *J. Math. Biology* **43**, 2001, 325–355.
21. B. Ermentrout: *Simulating, Analyzing, and Animating Dynamical Systems. A Guide to XPPAUT for Researchers and Students*. SIAM, Philadelphia, 2002.
22. G.B. Ermentrout and L. Edelstein-Keshet: Models for the length distribution of actin filaments: II. polymerization and fragmentation by Gelsolin acting together, *Bull. Math. Biol.* **60**, 1998, 477–503.
23. N.I. Fisher: *Statistical Analysis of Circular Data*, Cambridge University Press, Cambridge, 1993.
24. E. Geigant, K. Ladizhansky and A. Mogilner: An integro-differential model for orientational distributions of F-actin in cells, *SIAM J. Appl. Math.* **59**(3), 1998, 787–809.
25. E. Geigant and M. Stoll: Bifurcation analysis of an orientational aggregation model, *J. Math. Biol.* **46**, 2003, 537–563.
26. M.E. Gracheva and H.G. Othmer: A continuum model of motility in Ameboid cells, *Bull. Math. Biol.* **66**, 2004, 167–193.
27. H.P. Grimm, A.B. Verkhovskiy, A. Mogilner and J.-J. Meister: Analysis of actin dynamics at the leading edge of crawling cells: implications for the shape of keratocyte lamellipodia, *Eur. Biophys. J.* **32**, 2003, 563–577.
28. R. Horwitz and D. Webb: Cell migration, *Curr Biol.* **19**(13), 2003, R756–R759.
29. E.L. Ionides, K.S. Fang, R.R. Isseroff and G.F. Oster: Stochastic models for cell motion and taxis, *J. Math. Biol.* **48**, 2004, 23–37.
30. K.A. Landman, M.J. Simpson, J.L. Slater and D.F. Newgreen: Diffusive and chemotactic cellular migration: smooth and discontinuous travelling wave solutions, *SIAM J. Appl. Math.* **65**(4), 2005, 1420–1442.
31. D.A. Lauffenburger and A.F. Horwitz: Cell migration: a physically integrated molecular process, *Cell* **84**, 1996, 359–369.
32. C.P. Lowe: Dynamics of filaments: modelling the dynamics of driven microfilaments, *Phil. Trans. R. Soc. Lond. B* **358**, 2003, 1543–1550.
33. F. Lutscher: Modeling alignment and movement of animals and cells, *J. Math. Biology* **45**, 2002, 234–260.
34. T.L. Madden and J. Herzfeld: Crowding-induced organization of cytoskeletal elements, *Biophys. J.* **65**, 1993, 1147–1154.
35. B.C. Mazzag, I.B. Zhulin and A. Mogilner: Model of bacterial band formation in aerotaxis, *Biophysical J.* **85**, 2003, 3558–3574.

36. A. Mogilner and L. Edelstein-Keshet: Selecting a common direction, I: How orientational order can arise from simple contact responses between interacting cells, *J. Math. Biol.* **33**, 1995, 619–660.
37. A. Mogilner and L. Edelstein-Keshet: Spatio-angular order in populations of self-aligning objects: formation of oriented patches, *Physica D* **89**, 1996, 346–367.
38. A. Mogilner and L. Edelstein-Keshet: Regulation of actin dynamics in rapidly moving cells: a quantitative analysis, *Biophysical J.* **83**, 2002, 1237–1258.
39. A. Mogilner and G. Oster: The physics of lamellipodial protrusion, *Eur. Biophys. J.* **25**, 1996, 47–53.
40. A. Mogilner and G. Oster: The polymerization ratchet model explains the force-velocity relation for growing microtubules, *Eur. Biophys. J.* **28**, 1999, 235–242.
41. A. Mogilner and G. Oster: Force generation by actin polymerization II. The elastic Ratchet and tethered filaments, *Biophysical J.* **84**, 2003, 1591–1605.
42. A. Mogilner and B. Rubinstein: The physics of filopodial protrusion, *Biophysical J.* **89**, 2005, 1–14.
43. A. Mogilner and D.W. Verzi: A simple 1-d physical model for the crawling nematode sperm cell, *J. of Stat. Physics* **110**(3/6), 2003, 1169–1189.
44. I. L. Novak, B.M. Slepchenko, A. Mogilner and L.M. Loew: Cooperativity between cell contractility and adhesion, *Phys. Rev. Lett.* **93**, 2004, article no. 268109.
45. C.S. Peskin, G.M. Odell and G.F. Oster: Cellular motions and thermal fluctuations: the Brownian ratchet, *Biophysical J.* **65**, 1993, 316–324.
46. S. Portet, J. Vassy, C.W.V. Hogue, J. Arino and O. Arino: Intermediate filament networks: in vitro and in vivo assembly models, *C.R. Biologies* **327**, 2004, 970–976.
47. B. Rubinstein, K. Jacobson and A. Mogilner: Multiscale two-dimensional modeling of a motile simple-shaped cell, *SIAM J. Multiscale Models Simul.* **3**(2), 2005, 413–439.
48. J.A. Sherratt and J. Lewis: Stress-induced alignment of actin filaments and the mechanics of cytogel, *Bull. Math. Biol.* **55**(3), 1993, 637–654.
49. J.V. Small, A. Rohlfis and M. Herzog: Actin and cell movement, in: *Cell Behaviour: Adhesion and Motility*, G. Jones, C. Wigley and R. Warn (eds.), The Company of Biologists Ltd., Cambridge, 1993, 57–71.
50. A. Spiros, L. Edelstein-Keshet: Testing a model for the dynamics of actin structures with biological parameter values, *Bull. Math. Biol.* **60**, 1998, 275–305.
51. A. Stéphanou and Ph. Tracqui: Cytomechanics of cell deformations and migration: from models to experiments, *C. R. Biologies* **325**, 2002, 295–308.
52. A. Stéphanou, M.A. Chaplain and P. Tracqui: A mathematical model for the dynamics of large membrane deformations of isolated fibroblasts, *Bull. Math. Biol.* **66**(5), 2004, 1119–1154.
53. T.P. Stossel: The machinery of cell crawling, *Scientific American*, Sept. 1994, 40–47.
54. C.W. Wolgemuth, L. Miao, O. Vanderlinde, T. Roberts and G. Oster: MSP dynamics drives nematode sperm locomotion, *Biophysical J.* **88**, 2005, 2462–2471.
55. C.W. Wolgemuth, A. Mogilner and G. Oster: The hydration dynamics of polyelectrolyte gels with applications to cell motility and drug delivery, *Eur. Biophys. J.* **33**, 2004, 146–158.

Spectral Galerkin Method Applied to Some Problems in Elasticity

Chris J. Talbot

School of Computing and Engineering, University of Huddersfield, Huddersfield
HD1 3DH, UK, c.j.talbot@hud.ac.uk

Summary. Spectral methods offer an attractive alternative to finite element procedures for the numerical solution of problems in elasticity. Especially for simple domains, in both two and three dimensional elasticity, Navier's Equations or their non-linear generalisations can be solved using either collocation or Galerkin techniques. This paper examines the use of an efficient Galerkin method in linear elasticity and by comparing the numerical results with known analytic solutions demonstrates its validity. It then shows how such methods can be extended to include friction, and in particular shows how a model that involves sliding friction between a steadily rotating shaft and a fixed elastic body gives rise to a standard linear complementarity problem that can be easily solved.

1 Introduction

The vast majority of numerical methods used in elasticity are based on finite element techniques applied to a variational formulation of the problem. Spectral and spectral element methods have more typically been used in areas such as computational fluid dynamics where such techniques can provide efficient solvers in nonlinear time-stepping problems that are very costly in computing time. In the last decade the extension of the spectral approach to spectral elements has made the technique increasingly attractive for a range of applications. There is no reason why such techniques cannot be used in elasticity, especially for nonlinear and contact problems where transient solutions are of importance as well as the usual static and modal analysis.

This paper illustrates the use of a very efficient spectral Galerkin method in elasticity. It is first applied to elastostatic and modal vibration problems and compared with known analytic solutions to demonstrate how the typical accuracy expected from spectral techniques is obtained by relatively low order expansions. Then a problem involving friction is examined to demonstrate the effectiveness of the technique in this area. Even in the apparently simple problem investigated here - a rigid rotating shaft in an elastic collar with sliding friction - the surfaces cannot remain in contact throughout but

must allow separation to occur in what is effectively a type of free boundary problem. The spectral approach, as with finite element techniques, gives rise to a constrained optimisation problem. However the spectral method, involving banded matrices, is much more computationally efficient and should prove effective when applied to transient problems such as those involving frictional vibration.

2 Spectral Galerkin Method

The approach used in this paper follows the Spectral Galerkin method developed by Jie Shen for simple regions in rectangular and polar coordinates [5, 6, 7]. Shen's papers give examples of second and fourth order elliptic problems in dimensions 1,2 or 3. Consider for example the second order differential equation:

$$\frac{d^2u}{dx^2} - \lambda u = f \quad -1 \leq x \leq 1 \quad \text{with} \quad u(\pm 1) = 0 \quad (1)$$

The standard Galerkin method is to approximate u by an expansion in suitable polynomials ϕ_k satisfying the boundary conditions,

$$u_N = \sum_{k=0}^{N-2} c_k \phi_k, \quad (2)$$

and using integration by parts to replace (1) by

$$-((w\phi_k)', u_N') - \lambda(w\phi_k, u_N) = (w\phi_k, f) \quad (3)$$

where

$$(u, v) = \int_{-1}^1 uv dx$$

is the L^2 scalar product, and w is a suitable weight. (The summation from 0 to $N - 2$ follows Shen).

Shen's choice is either $\phi_k = L_k - L_{k+2}$, $k = 0, \dots, N - 2$ where $L_k(x)$ is the k th degree Legendre polynomial and $w = 1$, or $\phi_k = T_k - T_{k+2}$, $k = 0, \dots, N - 2$ where $T_k(x)$ is the k th degree Chebyshev polynomial and $w = (1 - x^2)^{-\frac{1}{2}}$. For these choices (3) gives rise to a matrix equation with banded matrices that can be given by appropriate analytic formulae. For example in the Legendre case:

$$-\sum_{j=0}^{N-2} (a_{ij} - \lambda b_{ij})c_j = f_i \quad \text{where} \quad f_i = (f, \phi_i) \quad (4)$$

$$a_{ij} = (\phi'_i, \phi'_j) = \begin{cases} 4i + 6, & i = j \\ 0, & i \neq j \end{cases} \quad (5)$$

and

$$b_{ij} = (\phi_i, \phi_j) = \begin{cases} \frac{2}{2i+1} + \frac{2}{2i+5} & i = j \\ -\frac{2}{2i+5} & i = j - 2 \\ -\frac{2}{2i+1} & i = j + 2 \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

To demonstrate the effectiveness of the method, (4) was solved for the case $f = 0$, i.e. an eigenvalue problem where the exact eigenvalues of the original differential equation are $-\frac{n^2\pi^2}{4}$ for integer n . For Legendre polynomial expansions with $N = 16$ and $N = 24$, the results for the first eight eigenvalues were as follows:

Table 1. Values of $-\frac{4\lambda}{\pi^2}$ for the Spectral Galerkin solution of $\frac{d^2u}{dx^2} = \lambda u$.

N=16	N=24
0.999999999999998	1.000000000000000
4.000000000000000	4.000000000000001
8.999999999999995	9.000000000000004
16.00000000026940	16.000000000000000
25.00000001567554	25.000000000000045
36.00005234790152	36.000000000000004
49.00044554199490	49.000000000000097
64.08006565457515	64.00000000373487

Extending the method to 2 or 3 dimensions is possible by taking Kronecker products of the Shen banded matrices (see [5]).

The extension of the method to polar coordinates is achieved by using a polynomial expansion in r , the radial coordinate (or r and z , i.e. cylindrical coordinates, in 3 dimensions) and a Fourier series expansion in the angular coordinate θ . For example, consider the eigenvalue equation of the Laplacian in an annular region:

$$\frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} = \lambda u \quad (7)$$

with $a \leq r \leq b$, $0 \leq \theta \leq 2\pi$, $u = 0$ when $r = a$ or b . The polynomial approximation the the n th Fourier term, $u_N^{(n)} = \sum_{k=0}^{N-2} c_k \phi_k(r)$, will satisfy:

$$-\int_a^b r^2 \frac{du_N^{(n)}}{dr} \frac{d(w\phi_k)}{dr} dr - n^2 \int_a^b u_N^{(n)} w\phi_k dr = \lambda \int_a^b r^2 u_N^{(n)} w\phi_k(r) dr \quad (8)$$

obtained by multiplying (7) by $r^2 w\phi_k$ and integrating by parts. Note that although ϕ_k is still a function of x the integration and differentiation in (8) is with respect to r where

$$r = s(x + c) \quad \text{with} \quad s = \left(\frac{b-a}{2}\right) \quad \text{and} \quad x = \left(\frac{b+a}{b-a}\right)$$

Taking the Legendre case ($w = 1$) (8) gives rise to a matrix eigenvalue equation

$$-\sum_{j=0}^{N-2} (p_{ij} + q_{ij} + n^2 r_{ij}) c_j = \lambda \sum_{j=0}^{N-2} s_{ij} c_j \quad (9)$$

where

$$p_{ij} = \int_a^b \frac{d\phi_i}{dr} \frac{d\phi_j}{dr} r^2 dr = s \left(\int_{-1}^1 x^2 \phi'_i \phi'_j dx + 2c \int_{-1}^1 x \phi'_i \phi'_j dx + c^2 \int_{-1}^1 \phi'_i \phi'_j dx \right),$$

$$q_{ij} = \int_a^b \phi_i \frac{d\phi_j}{dr} r dr = s \left(\int_{-1}^1 x \phi_i \phi'_j dx + c \int_{-1}^1 \phi_i \phi'_j dx \right),$$

$$r_{ij} = \int_a^b \phi_i \phi_j dr = s \int_{-1}^1 \phi_i \phi_j dx$$

and

$$s_{ij} = \int_a^b \phi_i \phi_j r^2 dr = s^3 \left(\int_{-1}^1 x^2 \phi_i \phi_j dx + 2c \int_{-1}^1 x \phi_i \phi_j dx + c^2 \int_{-1}^1 \phi_i \phi_j dx \right) \quad (10)$$

All the integrals in x in (10) turn out to be of banded type and can be given by explicit formulae. Solving (9) for the first two eigenvalues for $n = 1$ and $n = 2$ with $a = 1$ and $b = 2$, and using a Legendre polynomial expansion for $N = 8$ and 16 , the results agree well with the analytical solution:

Table 2. Eigenvalues of the Laplacian in an annular region $a \leq r \leq b$, $0 \leq \theta \leq 2\pi$.

Eigenvalue	N=8 Spectral	N=16 Spectral	Analytical
1st n=1	3.19657838080016	3.19657838081064	3.19657838081063
2nd n=1	6.31235023359561	6.31234951037327	6.31234951037326
1st n=2	3.40692142663368	3.40692142656752	3.40692142656753
2nd n=2	6.42776702931196	6.42776592259607	6.42776592259606

3 Linear Elasticity

The basic equations of linear elasticity with no external forces in a body B are as follows [2]:

$$\nabla \sigma = \rho \frac{\partial^2 \mathbf{u}}{\partial t^2} \quad (\text{Newton's Law of Motion}) \quad (11)$$

$$\sigma = \mathbf{C} \mathbf{s} \quad (\text{Generalised Hooke's Law})$$

$$\mathbf{s} = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T) \quad (\text{Strain - Displacement relation})$$

where $\sigma = \sigma(\mathbf{x}, \mathbf{t})$ is the stress tensor at each point \mathbf{x} of the elastic body at time t , $\mathbf{u} = \mathbf{u}(\mathbf{x}, \mathbf{t})$ is the displacement vector, $\mathbf{s} = \mathbf{s}(\mathbf{x}, \mathbf{t})$ is the strain tensor, ρ is the density (assumed constant) and C is the stiffness tensor (also assumed constant) given in terms of E (Young's Modulus), ν (Poisson's ratio), etc. for the elastic material under consideration. The boundary S of the body B can be partitioned into two parts S_u and S_σ , so that the boundary conditions are:

$$\begin{aligned} \mathbf{u} = \bar{\mathbf{u}} \quad \text{on} \quad \mathbf{S}_u \quad (\text{displacement boundary condition}) \\ \text{and} \quad \sigma \mathbf{n} = \bar{\mathbf{t}} \quad \text{on} \quad \mathbf{S}_\sigma \end{aligned}$$

(traction boundary condition where \mathbf{n} is the unit normal)

Taking the scalar product of (11) with weight vectors \mathbf{w} satisfying $\mathbf{w} = \mathbf{0}$ on S_u and using the Divergence Theorem yields the weak or variational form of the initial boundary value problem:

$$\int_B \nabla \mathbf{w} \cdot \mathbf{C} \nabla \mathbf{u} dS + \int_B \rho \mathbf{w} \cdot \frac{\partial^2 \mathbf{u}}{\partial t^2} dS = \int_{S_\sigma} \bar{\mathbf{t}} \cdot \mathbf{n} ds \quad (12)$$

As the traction boundary conditions appear explicitly in (12) they are often termed "natural" boundary conditions.

(12) are often written in a more convenient matrix form (see for example [8]). For example in two dimensions and in polar coordinates:

$$\int_B (LW)^T C (LU) dV + \int_B \rho W^T \frac{\partial^2 U}{\partial t^2} dV = \int_{S_\sigma} W^T \bar{T} dS \quad (13)$$

Here U, W and \bar{T} are column vectors containing the r and θ components of \mathbf{u}, \mathbf{w} and $\bar{\mathbf{t}}$ respectively,

$$L = \begin{bmatrix} \frac{\partial}{\partial r} & 0 \\ \frac{1}{r} \frac{\partial}{\partial \theta} & \frac{\partial}{\partial r} - \frac{1}{r} \end{bmatrix} \quad \text{and} \quad C = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}$$

(assuming the body is isotropic).

Approximate solutions can be obtained by assuming a Fourier series expansion in the angular coordinate θ as in the Laplacian case above and polynomial approximations in the radial coordinate r

Consider again, for example, the case of an annulus $a \leq r \leq b$ and $0 \leq \theta \leq 2\pi$. Again the calculations use Legendre polynomials. Two cases arise where (13) can be solved using the Spectral Galerkin method and compared with analytic solutions: the electrostatic case ($\frac{\partial^2 U}{\partial t^2} = 0$) and the case of vibrations at natural frequencies (assume $T = 0$ and $U = U(r, \theta)e^{i\omega t}$ so that $\frac{\partial^2 U}{\partial t^2} = -\omega^2 U$).

For the electrostatic case in order not to impose zero displacement (fixed) boundary conditions the choice of polynomials $\phi_k(x)$, $k = 0, \dots, N-2$, is extended in this case to include the zero and linear polynomials 1 and x which do not vanish on the $r = a$ and $r = b$ ($x = -1$ and $+1$) boundaries. Thus the n th Fourier terms in the displacement are $U^{(n)} = [u_N^{(n)} \ v_N^{(n)}]^T$ where

$$u_N^{(n)} = \sum_{k=0}^N u_k \phi_k(r) \quad \text{and} \quad v_N^{(n)} = \sum_{k=0}^N v_k \phi_k(r) \quad (14)$$

and in the surface tractions $\bar{T}^{(n)} = [f_N^{(n)} \ g_N^{(n)}]^T$ where

$$f_N^{(n)} = \sum_{k=0}^N f_k \phi_k(r) \quad \text{and} \quad g_N^{(n)} = \sum_{k=0}^N g_k \phi_k(r) \quad (15)$$

(Sums from 0 to N because the extra two polynomials are included).

Substituting (14) and (15) and $W = [r\phi_i(r) \ 0]^T$ and $W = [0 \ r\phi_i(r)]^T$ in turn into (13) yields:

$$A \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} b^2 f_N^{(n)}(b)\Phi(b) - a^2 f_N^{(n)}(a)\Phi(a) \\ b^2 g_N^{(n)}(b)\Phi(b) - a^2 g_N^{(n)}(a)\Phi(a) \end{bmatrix} \quad (16)$$

where

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad (17)$$

and

$$\begin{aligned} A_{11} &= (1-\nu)(P+Q) + \nu(Q+Q^T+R) + \left(\frac{1-2\nu}{2}\right)n^2 R \\ A_{12} &= n(\nu(R+Q^T) + (1-\nu)R) + \left(\frac{1-2\nu}{2}\right)(R-Q) \\ A_{21} &= n(\nu Q + (1-\nu)R - \left(\frac{1-2\nu}{2}\right)Q^T) \\ A_{22} &= n^2(1-\nu)R + \left(\frac{1-2\nu}{2}\right)(P-Q^T) \end{aligned} \quad (18)$$

Here P , Q and R are the banded matrices with elements p_{ij} , q_{ij} and r_{ij} defined in (10), u , v are the vectors with elements u_i and v_i , and

$$\Phi(r) = [\phi_0(r) \ \phi_1(r) \ \dots]^T. \quad (19)$$

For brevity the factor $\frac{E\nu}{(1+\nu)(1-2\nu)}$ has been absorbed into the traction \bar{T} .

A similar calculation can be performed for the case of elastic vibrations of the annulus. The choice of polynomial basis in this case is extended to include $1+x$ as well as $\phi_k(x)$, $k = 0, \dots, N-2$ so that displacement is zero on $r = a$ ($x = -1$) but not on $r = b$ where the traction is zero. Thus the components of U are

$$u_N^{(n)} = \sum_{k=0}^{N-1} u_k \phi_k(r) \quad \text{and} \quad v_N^{(n)} = \sum_{k=0}^{N-1} v_k \phi_k(r) \quad (20)$$

(sum from 0 to $N-1$ to include the extra polynomial in the basis).

Substituting (20) into (13) now yields the following matrix eigenvalue equation for each n :

$$A \begin{bmatrix} u \\ v \end{bmatrix} = w^2 \begin{bmatrix} S & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \quad (21)$$

where A is the matrix defined in (17), S is the banded matrix with elements s_{ij} defined in (10), u and v are the vectors with elements u_i and v_i and the constants. $\frac{E\nu}{(1+\nu)(1-2\nu)}$ and ρ have been absorbed into the angular frequency w .

Results from (20) with $N = 8$ and $N = 16$ are compared to the analytic solution for this case in Table 3

Table 3. Angular frequencies w for an annulus (scaled by a factor $\sqrt{\frac{\rho(1+\nu)(1-2\nu)}{E\nu}}$).

n	Analytic	$N = 8$	$N = 16$
3	2.38405696147517	2.38405701177263	2.38405696147517
4	2.71806186378332	2.71806206511980	2.71806186378331
5	3.07468120525290	3.07468168373889	3.07468120525290
6	3.48033825534612	3.48033907909987	3.48033825534612

4 Friction Contact

The Spectral Galerkin approach can be used to investigate frictional contact between linearly elastic bodies. Friction boundary conditions between two bodies generally assume that two restrictions hold: (1) (Kuhn-Tucker conditions) that the separation between the surfaces is positive, that the normal reaction is positive, and that if either of them is zero the other must be strictly positive; (2) (Coulomb Friction) that the magnitude of the tangential stress vector does not exceed the coefficient of friction μ multiplied by the normal contact force, with equality holding when the relative velocity is not zero [3]. Thus there are three possible cases at each point on the surface: separation;

contact with sliding (non-zero relative velocity); and contact with sticking (zero relative velocity) [4].

We consider here an example where the first two cases only occur, namely a rigid shaft rotating at a constant angular velocity encased in an collar modelled by a two dimensional annulus ($a \leq r \leq b$, $0 \leq \theta \leq 2\pi$ as in the previous section). The coefficient of friction is assumed constant and the relative velocity between the shaft and the collar is assumed to be always positive so that no sticking occurs.

Suppose that the surface of the rotating shaft is given by $r = R(\theta, t)$. Writing $U = [u \ v]^T$ and $\bar{T} = [f \ g]^T$ in (13) conditions (1) and (2) can be written

$$u - R(\theta, t) \geq 0, \quad f \geq 0 \quad \text{and} \quad (u - R(\theta, t))f = 0$$

and

$$g = \mu f \tag{22}$$

on the surface $r = a$ for $0 \leq \theta \leq 2\pi$.

If the angular velocity of the shaft is Ω , the equation for the surface of the shaft can be expanded in a Fourier series:

$$R(\theta, t) = r_0 + \sum_{k=1}^M R^{(n)} \cos(n(\theta - \Omega t)) + \hat{R}^{(n)} \sin(n(\theta - \Omega t))$$

Also, we may put

$$u = u^{(0)} + \sum_{k=1}^M u^{(n)} \cos(n(\theta - \Omega t)) + \hat{u}^{(n)} \sin(n(\theta - \Omega t)) \tag{23}$$

where for each n we may expand $u^{(n)}$ and $\hat{u}^{(n)}$ in terms of a polynomial basis:

$$u^{(n)} = \sum_{k=0}^{N-1} u_k \phi_k(r) \quad \text{and} \quad \hat{u}^{(n)} = \sum_{k=0}^{N-1} \hat{u}_k \phi_k(r)$$

(The summation is from 0 to $N - 1$ as we take the displacement fixed on the outer surface $r = b$ and so a term $1 - x$ is used as well as $\phi_k(x)$, $k = 0 \dots N - 2$). v , f and g can be given similar Fourier series expansions, and with similar expansion of the Fourier coefficients in terms of the polynomial basis. For consistency the cos terms are $u^{(n)}$, $-\hat{v}^{(n)}$, $f^{(n)}$ and $-\hat{g}^{(n)}$ whereas the sin terms are $\hat{u}^{(n)}$, $v^{(n)}$, $\hat{f}^{(n)}$ and $g^{(n)}$.

Substituting into (13) for each n there are now two distinct equations similar to (16) corresponding to the cos and sin terms, as well as an additional acceleration term in each equation with the factor $n^2 \Omega^2$ (as before the constant factors $\frac{E\nu}{(1+\nu)(1-2\nu)}$ and ρ are omitted by absorbing them into Ω and into the traction \bar{T}):

$$(A - n^2 \Omega^2) \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -a^2 f^{(n)}(a) \Phi(a) \\ -a^2 g^{(n)}(a) \Phi(a) \end{bmatrix} \tag{24}$$

and a similar equation containing \hat{u} , \hat{v} , \hat{f} and \hat{g} . A , S and $\Phi(r)$ are as defined as in (19) and (21). Note that all the terms in $\Phi(a)$ are zero apart from $\phi_{N-1}(a) = 2$ ($\phi_{N-1}(x) = 1 + x$) and that (22) implies $\hat{g}^{(n)}(a) = -\mu f^{(n)}(a)$ (cos terms) and $g^{(n)}(a) = \mu \hat{f}^{(n)}(a)$ (sin terms). This means that (24) can be used to find all the elements of u , \hat{u} , v and \hat{v} in terms of the elements of u_{N-1} and \hat{u}_{N-1} , where $u^{(n)}(a) = 2u_{N-1}$ and $\hat{u}^{(n)}(a) = 2\hat{u}_{N-1}$ leaving a single pair of equations of the form

$$P_{\mu}^{(n)} \begin{bmatrix} u^{(n)}(a) \\ \hat{u}^{(n)}(a) \end{bmatrix} = \begin{bmatrix} f^{(n)}(a) \\ \hat{f}^{(n)}(a) \end{bmatrix} \quad (25)$$

where $P_{\mu}^{(n)}$ is a 2×2 matrix. (Note that for $n = 0$ there will only be a single equation).

Using (23) and the similar Fourier expansion for f , the values of u and f on $r = a$ can be calculated in terms of $u^{(n)}(a)$, $\hat{u}^{(n)}(a)$, $f^{(n)}(a)$ and $\hat{f}^{(n)}(a)$ for a given time t , for $n = 0, \dots, M$. If they are calculated at the $2M + 1$ evenly spaced points $\theta_k = \frac{\pi k}{M}$, $k = 0, \dots, 2M$, so that

$$\tilde{u} = [u(a, \theta_0) \ u(a, \theta_1) \ u(a, \theta_2) \ \dots]^T \quad \text{and} \quad \tilde{f} = [\tilde{f}(a, \theta_0) \ \tilde{f}(a, \theta_1) \ \tilde{f}(a, \theta_2) \ \dots]^T \quad (26)$$

the equations (25) can be used to relate the vectors \tilde{u} and \tilde{f} :

$$P\tilde{u} = \tilde{f}$$

where P is a $(2M + 1) \times (2M + 1)$ matrix formed from $P_{\mu}^{(n)}$, $n = 0, \dots, M$. If \tilde{R} is the vector formed from the equation for the shaft surface $r = R(\theta, t)$ at θ_k , $k = 0, \dots, 2M$, and at the given time t , i.e. $\tilde{R} = [R(\theta_0, t) \ R(\theta_1, t) \ R(\theta_2, t) \ \dots]^T$ then using (26) the Coulomb Friction conditions (2) become:

$$\tilde{u} - \tilde{R} \geq 0 \quad \tilde{f}P\tilde{u} \geq 0 \quad \text{and} \quad (\tilde{u} - \tilde{r})^T.P\tilde{u} = 0 \quad (27)$$

which in the area of mathematical programming is known as a linear complementarity problem [1] and can be solved by standard algorithms. Note that although the results depend on the choice of time t , changing t will only effect a rotation $\theta \rightarrow \theta - t$ and give essentially the same steady state problem. Typical solutions of (27) are shown in the Figure 1 showing \tilde{u} , \tilde{R} and \tilde{f} plotted against angle θ .

5 Conclusions

The effectiveness of spectral Galerkin techniques in elasticity has been demonstrated in a number of illustrative cases. Typical spectral accuracy and computational efficiency can be obtained. The author has been able to extend the technique to three dimensions and in two dimensions to join simple regions together using continuity conditions at the boundaries. In the application to

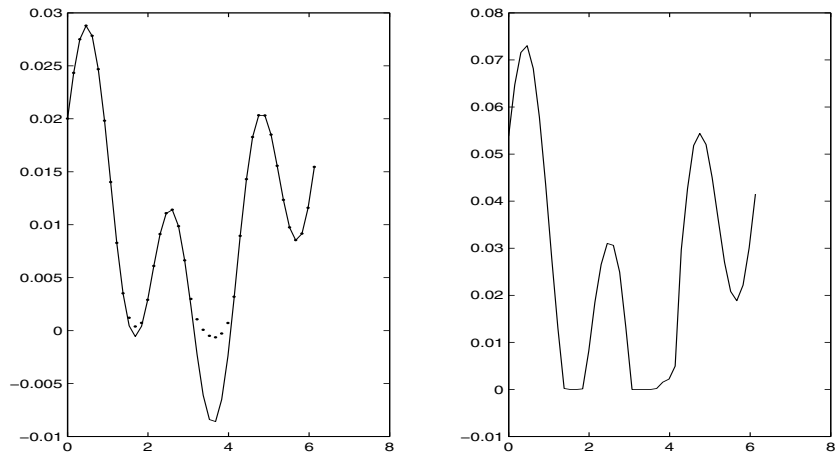


Fig. 1. Left hand graph shows variation of shaft surface (continuous line) and displacement of collar (dotted line) against angle. Right hand graph shows variation of normal reaction against angle, showing that reaction is zero when collar moves away from shaft.

friction problems the possibility of extending the method to studying transient vibration problems – without the very high computing overheads involved in finite element techniques – is now being investigated.

References

1. R.W. Cottle, J.-S. Pang, and R.E. Stone (eds.): *The Linear Complementarity Problem*. Academic Press, 1992.
2. G.A. Holzapfel: *Nonlinear Solid Mechanics*. Wiley, 2000.
3. T.A. Laursen: *Computational Contact and Impact Mechanics*. Springer, 2002.
4. J.A.C. Martinis and M. Raous (eds.): *Friction and Instabilities*. CISM Courses and Lectures, no. 457. Springer, 2002.
5. J. Shen: Efficient spectral-Galerkin method I. direct solvers for the second on fourth order equations using Legendre polynomials. *SIAM J. Comput.* **15**, 1994, 1489–1505.
6. J. Shen: Efficient spectral-Galerkin method II. direct solvers for the second on fourth order equations using Chebyshev polynomials. *SIAM J. Comput.* **16**, 1995, 74–87.
7. J. Shen: Efficient spectral-Galerkin method III. polar and cylindrical geometries. *SIAM J. Comput.* **18**, 1997, 1583–1604.
8. O.C. Zienkiewicz and K. Morgan: *Finite Elements and Approximation*. Wiley, 1983.

Statistical Approximation Methods

Bayesian Field Theory Applied to Scattered Data Interpolation and Inverse Problems

Chris L. Farmer^{1,2}

¹ Schlumberger Abingdon Technology Center, Abingdon OX14 1UJ, UK,

² Oxford Centre for Industrial and Applied Mathematics, University of Oxford, Oxford OX1 3LB, UK, farmer5@s1b.com

Summary. Problems of scattered data interpolation are investigated as problems in Bayesian statistics. When data are sparse and only available on length scales greater than the correlation length, a statistical approach is preferable to one of numerical analysis. However, when data are sparse, but available on length scales below the correlation length it should be possible to recover techniques motivated by more numerical considerations. A statistical framework, using functional integration methods from statistical physics, is constructed for the problem of scattered data interpolation. The theory is applicable to (i) the problem of scattered data interpolation (ii) the regularisation of inverse problems and (iii) the simulation of natural textures. The approaches of Kriging, Radial Basis Functions and least curvature interpolation are related to a method of ‘maximum probability interpolation’. The method of radial basis functions is known to be adjoint to the Universal Kriging method. The correlation functions corresponding to various forms of Tikhonov regularisation are derived and methods for computing some samples from the corresponding probability density functionals are discussed.

1 Introduction

Scattered data interpolation is the process of reconstructing a function given a relatively small number of values at known points. There may be error in the values and the coordinates of the points. The problem is said to be *scattered* when the sampling points do not fill a regular grid. If the function is smooth on the scale of the separation of the data points, the problem is a classical problem in numerical analysis. Although classical, the problem is still an area of active research with much interest in the radial basis function and neural network communities [18, 24]. When the function is not smooth between the data points, the inherent non-uniqueness in the problem becomes obvious. It is then more appropriate to use statistical methods. One aim of this paper is to show that in the statistical case the problem loses none of its appeal to the functional analyst. Indeed the problem becomes even more challenging. We

hasten to add, the problem is of enormous practical importance as well as of great theoretical interest, [13].

A generalisation of the scattered data interpolation problem is obtained by seeking *two* functions where some sample values are available for one or both and where the two functions are related by being, for example, the solution and the coefficient function in an elliptic boundary value problem. One might view this as a problem in the constrained interpolation of functions, or, as is most common, as a problem in the class of *inverse* problems. Inverse problems are normally regarded as conceptually distinct from interpolation. Another view is to regard scattered data interpolation as a special case of an inverse problem. However, in the following we will regard inverse problems as generalised constrained scattered data interpolation problems. The motive for this is that the theory is easier to explain and motivate when the scattered data interpolation problem is considered first.

The general inverse problem is far more difficult than the scattered data interpolation problem. This is primarily due to the nonlinear dependence of the observations upon the properties of the system - nonlinearity that can be present even in physical problems with linear models. For example the solution of a linear diffusion equation is a nonlinear functional of the conductivity function. Another difficulty is the inconsistency that can be present in the data, through measurement error or through modelling error. However, such inconsistency can be removed using a least squares approach. Least squares does not remove under-determination. This needs a regularisation procedure, a statistical formulation or systematic construction of all possible (or at least very many) solutions explaining the observations.

The main aim in the following is to review various approaches to solving inverse problems and show how they all fit into a common, Bayesian framework. Much of the material is already known, but spread through a large literature appearing in many different disciplines. We do however prove some new results that help build intuition regarding the properties of the various methods.

2 Scattered Data Interpolation

Spatial statistics, often called *geostatistics*, is concerned with problems of *interpolation under conditions of uncertainty*.

Consider, for example, interpolating a scalar valued function $\varphi = \varphi(x)$, in some region, Ω , of D -dimensional space, \mathbb{R}^D , where the values of φ , $\{\varphi_i\}$, at the points $\{x_i\}$ have been measured with only small errors. Further data are abstracted from some ‘prototype’ or analogue that could be said to ‘look like’ or ‘have the same texture’ as the property that φ is to model. To be specific; given detailed information about a function regarded as of the same ‘type’ as the one to be interpolated, construct an interpolant of the actual measurements that is qualitatively the same as the prototype. Where there

are manifest differences between the prototype and the system to be modelled it is necessary to devise methods of transforming the data relating to the prototype in response to expert judgement. The prototype is used to assign realistic estimates of statistical measures such as correlation functions (this is defined later on). It is a mistake to use *only* the measured data available from the target system to ascertain the correlation structure, unless the data are sampled on a scale smaller than the correlation length.

There are obviously many possible interpolants of the data that look like the prototype. Uncertainty quantification is the characterisation of the variation between these different, but data consistent, interpolants. Sometimes only one of these interpolants is selected. For example, the one that is, in some sense, the ‘smoothest’ or the ‘most probable’. Some methods, such as kriging, allow an estimate of uncertainty to be assigned to these single estimates.

There are several approaches to this interpolation problem; approaches that are not always equivalent. It is, however, generally agreed that some probabilistic element is required. Having said that, it is also the case that deterministic interpolation procedures are in widespread use. Thus, before reviewing statistical and stochastic methods, a paragraph on deterministic methods is provided. Later sections show these methods to be closely related to kriging. This is not a new result [16] but does not seem to be widely known. For a conventional exposition of geostatistics see the books [8, 9].

3 Deterministic Scattered Data Interpolation

There are two main classes of deterministic interpolation method. In both classes an interpolant, dependent upon a fixed number of unknown parameters is proposed. Then, when the number of parameters is the same as the number of data points, in the first class of method the scattered data are used to provide a system of algebraic equations for the parameters. Often the equations are linear and so the scattered data interpolation problem reduces to an algebraic problem. In the second class of deterministic method, where there are more parameters than data points, an objective function (in addition to the interpolant) is also proposed and is then minimised over the set of proposed interpolants. In this second class the data can either be imposed as a set of constraints or they can be incorporated into the objective function as known parameters. As the main deterministic methods used for practical problems are special cases of statistical methods (as reviewed later on) we do not give a separate review here. For further detail and references to the literature see [13]. In the limit as the number of parameters tends to infinity, somewhat amusingly, this is called a *non-parametric method*, [20].

4 Statistical Scattered Data Interpolation

Two classes of probabilistic approach are possible. One class is the direct probability density functional approach, often generalising the multivariate Gaussian (normal) distribution. The other class consists of models defining a stochastic process. In this second class of method it is not usually possible to state an explicit probability density functional for the interpolants; the process must be studied via its sample realisations and their properties. The derivation of standard geostatistical results often appears to be model based but, as shown in the next few pages, can be derived from an explicit probability density functional. More research using explicit probability density functionals could lead to new results and insights into the methods of spatial statistics. (The following two sections reproduce similar material from [13].)

4.1 Random Fields

Review of Some Basic Theory

This subsection reviews some basic properties of Gaussian random fields in D -dimensions. First the idea of the *functional derivative*,

$$\frac{\delta F}{\delta \varphi(x)}$$

of a functional $F[\varphi]$, is introduced. To accomplish this, define the ‘first functional differential’

$$DF[\varphi : \delta\varphi] = \frac{d}{d\epsilon} F[\varphi + \epsilon\delta\varphi]|_{\epsilon=0}$$

for arbitrary functions $\delta\varphi$. If the differential can be written as an integral over the domain of interest, Ω ,

$$DF[\varphi : \delta\varphi] = \int_{\Omega} \xi(x)\delta\varphi(x)d^Dx$$

then the function valued functional, $\xi(x)$, is called the ‘functional derivative’ of F and the notation

$$\xi(x) = \frac{\delta F}{\delta \varphi(x)}$$

is used. Higher order functional derivatives are then defined by applying functional differentiation to the lower order functional derivatives, as all functional derivatives are themselves functionals. For more information concerning the functional differential calculus see [4].

A later theorem needs the well known result that

$$\int_{-\infty}^{\infty} e^{-y^2/2} dy = \sqrt{2\pi} \quad (1)$$

and the further expression, obtained by completing the square that

$$\int_{-\infty}^{\infty} e^{-\frac{\lambda}{2}\gamma^2 + j\gamma} d\gamma = \sqrt{\frac{2\pi}{\lambda}} e^{\frac{j^2}{2\lambda}} \quad (2)$$

for real λ , γ and j . We note that the last result holds also for complex j but this is not used in the following.

General Gaussian Random Fields

The functional probability density of a general Gaussian random field, $\gamma(x)$ with zero mean is of the form

$$\pi(\gamma) = C \exp(-H[\gamma]), \quad (3)$$

where

$$H[\gamma] = \frac{1}{2} \int_{\Omega \times \Omega} \gamma(x) a(x, y) \gamma(y) d^D x d^D y, \quad (4)$$

and the integral is over Ω , the volume, or area, of interest. C is a normalisation constant such that

$$\int_S \pi(\gamma) D[\gamma] = 1 \quad (5)$$

where $D[\gamma]$ denotes integration over some suitable space of functions, S . A general Gaussian random field with non-zero mean is written as $\varphi(x) = h(x) + \gamma(x)$, where $h(x)$ is the expectation value, or mean of φ and γ has an average of zero.

One way to make sense of functional integrals such as (5) is to discretise on a finite grid of N cells, with γ_i a uniform value in the i -th cell. Then, using the same symbol for the approximate γ function,

$$\pi(\gamma) = C_N \exp\left(-\frac{1}{2} \sum_{i,j} \gamma_i a_{i,j} \gamma_j\right) \quad (6)$$

and $a_{i,j} = \int_{x \in \Omega_i, y \in \Omega_j} a(x, y) d^D x d^D y$ is the integral over the cells, Ω_i and Ω_j . (6) is just the usual expression for the zero-mean multivariate Gaussian distribution. The coefficient C_N is chosen so that the integral of the distribution over all N variables is unity.

Introducing Green's function, $g(x, y)$, defined as the solution of the integral equation

$$\int_{\Omega} a(x, y) g(y, z) d^D y = \delta(x - z) \quad (7)$$

where $\delta(x - z)$ is the usual Dirac δ -function, the following result holds:

$$\langle \gamma(x) \gamma(y) \rangle = g(x, y) \quad (8)$$

That is, the *Green's function is the correlation function*, where the angular brackets denote the average obtained by integrating over all functions in the space, S , with the probability measure, $\pi(\gamma)$.

To prove this result, first define the *moment generating functional*

$$Z[J] = \int_S \exp\left(-H[\gamma] + \int_{\Omega} \gamma(x)J(x)d^Dx\right) D[\gamma]$$

where $J(x)$ is an arbitrary function. Before giving meaning to this last formal expression note that the correlation functions can be derived via functional derivatives of Z with respect to J evaluated at $J = 0$. Thus

$$\langle \gamma(x)\gamma(y) \rangle = \frac{1}{Z[0]} \frac{\delta^2 Z[J]}{\delta J(x)\delta J(y)}.$$

To define the functional integral and to prove the result (8), expand all functions as infinite superpositions of eigenfunctions $\psi_n(x)$ with eigenvalues λ_n , defined by the equations

$$\int_{\Omega} a(x,y)\psi_n(y)d^Dy = \lambda_n\psi_n(x).$$

Then set

$$\gamma(x) = \sum_n \gamma_n \psi_n(x), \quad J(x) = \sum_n J_n \psi_n(x)$$

assuming the eigenfunctions are normalised so that $\int_{\Omega} \psi_n(y)\psi_m(y)d^Dy = \delta_{nm}$.

First note the standard result that

$$\delta(x-y) = \sum_n \psi_n(x)\psi_n(y).$$

(For arbitrary $f(x)$, $f(x) = \sum f_n \psi_n$, $\int f(x) \sum_n \psi_n(x)\psi_n(y)d^Dx = \sum f_n \psi_n(y)$.)

Then by substitution of

$$g(x,y) = \sum_n \frac{\psi_n(x)\psi_n(y)}{\lambda_n} \tag{9}$$

into the integral equation (7), it follows that (9) is a representation of Green's function.

Substitution into the generating functional gives

$$Z[J] = \int_{-\infty}^{\infty} \prod_n d\gamma_n e^{-\frac{1}{2}\lambda_n\gamma_n^2 + J_n\gamma_n}.$$

Exchanging the order of the product and the integral leads to

$$Z[J] = \prod_n \int_{-\infty}^{\infty} d\gamma_n e^{-\frac{1}{2}\lambda_n\gamma_n^2 + J_n\gamma_n}$$

and using (1) and (2)

$$Z[J] = \prod_n \sqrt{\frac{2\pi}{\lambda_n}} e^{\frac{J_n^2}{2\lambda_n}}.$$

Finally using the expression (9), gives

$$Z[J] = Z[0] \exp\left(\frac{1}{2} \int_{\Omega \times \Omega} J(x)g(x, y)J(y)d^D x d^D y\right)$$

where

$$Z[0] = \prod_n \sqrt{\frac{2\pi}{\lambda_n}}.$$

It then follows that

$$\frac{1}{Z[0]} \left(\frac{\delta^2 Z[J]}{\delta J(x) \delta J(y)} \right)_{J=0} = g(x, y),$$

and thus the correlation function is a Green's function.

4.2 Local Gaussian Random Fields

In the following, energy functionals of the form,

$$H[\varphi] = \frac{1}{2} \int_{\Omega} [a_2(\nabla^2(\varphi - h))^2 + a_1(\nabla(\varphi - h))^2 + a_0(\varphi - h)^2] d^D x \quad (10)$$

are studied. Using Gauss' theorem and assuming suitable vanishing boundary conditions this can be written in the form,

$$H[\varphi] = \frac{1}{2} \int_{\Omega} (\varphi - h)L(\varphi - h)d^D x \quad (11)$$

where the linear partial differential expression, $L(\varphi - h)$ is

$$L(\varphi - h) = a_2 \nabla^2(\nabla^2(\varphi - h)) - a_1 \nabla^2(\varphi - h) + a_0(\varphi - h). \quad (12)$$

To understand the correlations of the random field φ with mean field h the generating functional

$$Z[J] = \int_S \exp\left(-H[\varphi] + \int_{\Omega} \varphi(x)J(x)d^D x\right) D[\varphi]$$

is evaluated. To do this, introduce ψ_n , the n -th eigenfunction, and λ_n the n -th eigenvalue, of L so that

$$L\psi_n = \lambda_n \psi_n$$

and it is assumed that the eigenfunctions are normalised to unity. Noting that the eigenfunctions satisfy a condition of orthonormality the following expansions, $\varphi = \sum_n \varphi_n \psi_n$, $h = \sum_n h_n \psi_n$ and $J = \sum_n J_n \psi_n$ are inserted

into the generating functional, and following a similar argument as for the non-local zero-mean case earlier, one calculates that

$$\frac{Z[J]}{Z[0]} = \exp\left(\frac{1}{2} \int_{\Omega \times \Omega} J(x)g(x,y)J(y)d^D x d^D y + \int_{\Omega} h(x)J(x)d^D x\right)$$

where g is the Green's function satisfying $Lg(x,y) = \delta(x-y)$. It can be seen, as before, that the Green's function is the correlation function and that h is, indeed, the mean (as follows from evaluating the first and second functional derivatives).

Examples of Local Gaussian Random Fields

The Biharmonic-Helmholtz Functional. For later convenience the functional considered in the previous section is re-written in the form

$$H[\varphi] = \frac{a}{2} \int_{\Omega} [(\nabla^2(\varphi - h))^2 + 2b^2 \cos(2t)(\nabla(\varphi - h))^2 + b^4(\varphi - h)^2] d^D x \quad (13)$$

where a , b and t are real parameters with $a > 0$. Since $\cos(2t)$ can be negative it is interesting to observe that by completing the square it can be shown that this functional is positive for all real values of the parameter t . Thus, using Gauss' theorem and assuming vanishing boundary conditions,

$$\begin{aligned} \int_{\Omega} [(\nabla^2\psi)^2 + 2b^2 \cos(2t)(\nabla\psi)^2 + b^4\psi^2] d^D x = \\ \int_{\Omega} [(\nabla^2\psi)^2 - 2b^2 \cos(2t)\psi\nabla^2\psi + b^4\psi^2] d^D x. \end{aligned}$$

Then, by completing the square,

$$H[\varphi] = \int_{\Omega} [(\nabla^2\psi - b^2 \cos(2t)\psi)^2 + b^4(1 - \cos^2(2t))\psi^2] d^D x.$$

This last expression is positive since $1 - \cos^2(2t) \geq 0$ for all t .

The correlation function, $g(x,y)$ is the Green's function that satisfies the equation

$$a\nabla^2(\nabla^2 g) - 2ab^2 \cos(2t)\nabla^2 g + ab^4 g = \delta(x-z). \quad (14)$$

Using Fourier transform techniques, [2] one can show that, in 3-D, where radial symmetry is exploited in infinite space and $r = |x-y|$,

$$g(x,y) = g(r) = \frac{1}{2a\pi^2 r} \int_0^\infty dk \frac{k \sin(kr)}{(k^4 + 2b^2 \cos(2t)k^2 + b^4)}.$$

This integral can be evaluated using the calculus of residues or, more easily, by referring to the tabulated integrals in Gradshteyn and Ryzhik [15]

$$\langle \varphi(x)\varphi(y) \rangle = g(x, y) = g(r) = \frac{e^{-|x-y|b \cos t} \sin(|x-y|b \sin t)}{4\pi a|x-y|b^2 \sin(2t)}. \quad (15)$$

The validity of this result requires that $a > 0$, $b > 0$, $|t| < \frac{\pi}{2}$. It is quite clear that in general this Green's function is a decaying and oscillatory function. There are values of the parameters that give a simple decay.

An example of an oscillatory Green's function is shown in Figure 1, and Figure 2 shows one that simply decays. The parameters written on the figures correspond to the parameters in equation (15).

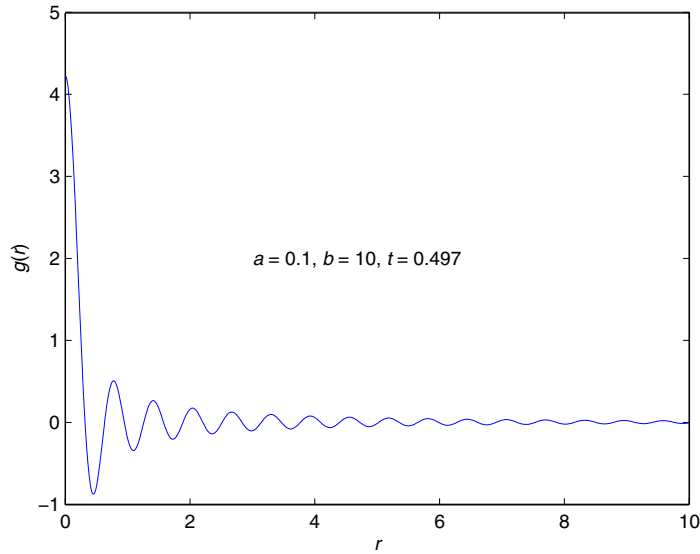


Fig. 1. Oscillatory Green's function for the 3D Biharmonic-Helmholtz equation

The limiting case of the previous equation, when $t = 0$ is of interest, in which case the Green's function reduces to

$$\langle \varphi(x)\varphi(y) \rangle = g(x, y) = \frac{e^{-|x-y|b}}{8\pi ab}.$$

In 2-D it does not appear possible to obtain the Green's function for the Biharmonic-Helmholtz equation in closed form. However, it can be reduced to the integral,

$$\frac{1}{2\pi a} \int_0^\infty dk \frac{k J_0(kr)}{k^4 + 2b^2 \cos(2t)k^2 + b^4} \quad (16)$$

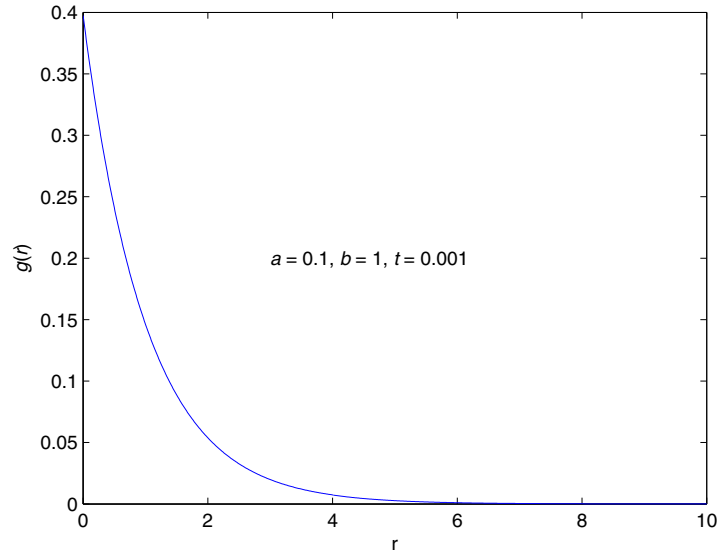


Fig. 2. Monotonic Green's function for the 3D Biharmonic-Helmholtz equation

where J_0 is the zeroth-order Bessel function. Evaluation of this integral using the trapezoidal rule shows that the Green's function has the same qualitative form as the 3-D version.

In 1-D the Green's function is,

$$g(r) = \frac{e^{-rb \cos t}}{2ab^3 \sin 2t} \sin(t + rb \sin t) \quad (17)$$

which is again of the same qualitative form as the 3D version.

It is apparent from the figures, and the limiting cases, that the Biharmonic-Helmholtz, Gaussian random field has a wide range of qualitative behaviour. The correlation function can mimic the general form of many of the correlation functions in general use [8]. Yet this particular correlation function is the result of a local model. The property of locality means that (i) it is easier to sample from the distribution (ii) the model can easily be generalised to curvilinear coordinates and (iii) the probability density function can be defined in cases where there is no one, global, coordinate system. This latter circumstance is common in the geosciences, where several local coordinate systems must be used together.

A more rigorous treatment of Gaussian local random fields can be found in the paper [21] where the notion of 'Markov Random Field' is used, rather than that of locality.

The Biharmonic-Laplace Functional. An interesting example is provided by the functional,

$$H[\varphi] = \frac{a}{2} \int_{\Omega} [(\nabla^2(\varphi - h))^2 + b^2(\nabla(\varphi - h))^2] d^D x.$$

The correlation function relating to the Biharmonic-Laplace functional is not derivable as a limiting case of the Biharmonic-Helmholtz example.

In 3-D the Biharmonic-Laplace functional leads to the correlation function,

$$g(r) = \frac{1}{4\pi ab^2 r} (1 - e^{-rb}). \quad (18)$$

The Damped Biharmonic Functional. Leaving out the gradient term in equation (13) the equation

$$a\nabla^2(\nabla^2 g) + ab^4 g = \delta(x - z) \quad (19)$$

for the Green's function is obtained. In 3-D it is found that

$$g(r) = \frac{e^{-\frac{rb}{\sqrt{2}}}}{4\pi arb^2} \sin\left(\frac{rb}{\sqrt{2}}\right).$$

In 1-D the result is,

$$g(r) = \frac{e^{-\frac{rb}{\sqrt{2}}}}{2ab^3} \sin\left(\frac{\pi}{4} + \frac{rb}{\sqrt{2}}\right). \quad (20)$$

The Helmholtz Functional. The functional

$$H[\varphi] = \frac{a}{2} \int_{\Omega} [(\nabla\varphi)^2 + b^2\varphi^2] d^D x$$

leads to the Green's function partial differential equation

$$-a\nabla^2 g + ab^2 g = \delta(x - z)$$

which in 3D has the well-known solution

$$g(r) = \frac{e^{-rb}}{4\pi ar},$$

and in the limit that $b = 0$ reduces to

$$g(r) = \frac{1}{4\pi ar}.$$

In 2D the Green's function is

$$g(r) = \frac{1}{2\pi a} K_0(br)$$

where K_0 is the zeroth order modified Bessel function of the third kind, which decays monotonically. In 1D the Green's function is

$$g(r) = \frac{e^{-br}}{2ab}.$$

The Laplace Functional. The functional

$$H[\varphi] = \frac{a}{2} \int_{\Omega} (\nabla \varphi)^2 d^D x$$

leads to the Green's function partial differential equation,

$$-a \nabla^2 g = \delta(x - z).$$

We have already seen the solution in 3D. In 2D and 1D the solution cannot be found as a simple limit as $b \rightarrow 0$ from, say, the Helmholtz functional. It can be found directly in 2D, and is

$$g(r) = -\frac{1}{2\pi a} \ln r.$$

This is not a very useful correlation function (because it is unbounded as $r \rightarrow \infty$), and so we must be wary of the use of a pure Laplacian probability density functional in 2D on an infinite region.

The White Noise Functional. The functional

$$H[\varphi] = \frac{a}{2} \int_{\Omega} \varphi^2 d^D x$$

is included for completeness, and leads to the Green's function or correlation function,

$$ag = \delta(x - z).$$

Discretisation of the white-noise functional on a rectangular grid leads to the strange properties enjoyed by the white-noise stochastic process.

It will be noticed that the above list of examples is not complete. This is due to the fact that in some cases the Green's functions do not decay suitably at infinity. As far as we understand, this is related to the phenomenon of 'boundary layers at infinity' [22]. This phenomenon occurs when an apparently small term with a derivative of lower order than the highest in the equation, or even a term just involving the Green's function, enables us to satisfy the decay condition at infinity. This happens more often in 1D than in 2D or 3D because of the presence (when in radial coordinates) of first order derivative terms with a decaying coefficient. One can, nevertheless, find the Green's functions in finite geometries. These, however, are not of a homogeneous form (i.e. functions of $|x - y|$) and are not easy to interpret. Another feature is that Green's functions in finite geometries, when the corresponding infinite geometry Green's functions do not decay at infinity, display sensitivity to the size of the domain. Examples of Green's functions in finite geometries can be found in [10].

4.3 Bayesian Statistics and Random Fields

For a comprehensive introduction to the theory and practice of Bayesian statistics see [17]. For a short introduction see [29].

Bayes' Theorem

Before outlining the Bayesian formulation of spatial statistics, let us review Bayes' theorem. First a definition of the *conditional probability density* is given. Let $\pi(x, y)$ be a probability density functional where x and y can be real valued parameters, finite or infinite vectors of real parameters or functions of a scalar or vector real variable. Then the conditional probability density functional $\pi(x|y)$ is defined by

$$\pi(x|y) = \frac{\pi(x, y)}{\pi(y)}$$

where the *marginal* distribution $\pi(y)$ is defined by

$$\pi(y) = \int_{\Omega_x} \pi(x, y) dx$$

where $\Omega = \Omega_x \times \Omega_y$ is the region (which could be a function space) over which the probability density is defined. When the arguments are functions, these expressions are formal, and great care needs to be taken in practice. A similar definition is given for $\pi(y|x)$.

Bayes' theorem then states that,

$$\pi(x|y)\pi(y) = \pi(y|x)\pi(x)$$

which follows directly from the definitions of the conditional probability density and marginal density functions.

Strictly speaking the probability density functional $\pi(x, y)$ should be written as $\pi(x, y|I)$ where I denotes the totality of the relevant information that is available before any observations are made. Some authors do include such a symbol in all their equations. However, as there are many other symbols to be used in the description of inverse problems, the convention is adopted in the following that the background information is implicitly present, and not included in the expressions.

Bayes' Rule

The essential idea in Bayesian statistics is, before observations are analysed, all prior knowledge about possible values of the observations is encoded in a joint probability density. Suppose that x represents some observations, and y some functions or parameters to be inferred from the observations. Then, before the observations are made but having modelled the prior information, one can state Bayes' theorem as trivially true of the prior. Bayes' *rule* is then to use the actual values of the observations x^* , say, to compute the *posterior* probability density functional, $\pi(y|x^*)$ using the formula

$$\pi(y|x^*)\pi(x^*) = \pi(x^*|y)\pi(y).$$

The function $\pi(x^*|y)$, considered as a function of y is known as the *likelihood function*.

Bayes' rule, although published posthumously in 1763, is still a cause of considerable controversy. See [26] and [3] for some philosophical and historical background. Our view is that Bayes' rule is a useful approach, that links together most other approaches. However, further philosophical analysis is needed, particularly in the context of inverse problems.

Bayesian Formulation of Spatial Statistics

Now the general Bayesian formalism is applied to the specific problem of spatial statistics. Suppose that φ is an unknown scalar field. Suppose that observations of a functional, $A[\varphi]$, of the field, $\alpha = A[\varphi]$ are available and suppose further that the observations are made with independent errors with variance σ . The joint probability density functional of the field and the observations (before they are analysed) is then

$$\pi(\varphi, \alpha, c) = \delta_\sigma(\alpha - A[\varphi])\pi(\varphi|c)\pi(c),$$

where δ_σ is a Gaussian distribution with variance σ (the δ symbol is used to emphasise that the Gaussian is close to a delta-function). The probability distribution for φ depends on a finite vector of parameters c - known as 'hyperparameters' - which themselves have a probability density functional, $\pi(c)$. This can be generalised to the case where c is a 'hyperfunction' [20]. Note that where the symbol π is used with different arguments it is generally a different function (a standard notation used in the statistics literature).

Bayes' rule then provides the *posterior* probability density given by

$$\pi(\varphi, c|\alpha^*) = \frac{\delta_\sigma(\alpha^* - A[\varphi])\pi(\varphi|c)\pi(c)}{\int_S D[\varphi] dc \delta_\sigma(\alpha^* - A[\varphi])\pi(\varphi|c)\pi(c)}, \quad (21)$$

where α^* are the actual *values* of the measurements.

The core ingredients of Bayesian statistics are: (i) every function and parameter that is not known exactly (or very nearly exactly) is described by probability densities that quantify the background data available - analogue data, opinions and previous studies (ii) a model of the physical system under consideration, including a model of the way errors or noise corrupt the measurement process - the likelihood function (iii) the data from the observations (iv) Bayes' rule for calculating the posterior density from the product of the likelihood and the prior probability density functional (v) a technique for sampling from the posterior distribution (vi) techniques for visualising the posterior distribution and (vii) a technique for summarising the posterior distribution. 'Summarising the distribution' implies, for example, calculating the mean and correlation functions. See [17] for a clear account of Bayesian statistics and the role of summarising the posterior distribution.

4.4 Maximum Probability Interpolation

Let us suppose that the prior probability density functional for a particular scattered data interpolation problem is given as a Gaussian random field. Suppose also that the errors in the observations of each diagnostic functional are small, Gaussian and independent from one another. It follows that the posterior probability density functional is also Gaussian. The mean of a Gaussian distribution is determined by its maximum value, thus a very useful summary of the posterior probability density functional in this case is to compute the maximum value. This leads to the method of *maximum a posteriori probability* estimation, or the maximum mode method. In the following it will be called *maximum probability interpolation*.

There are various forms of this problem. One could assume that the mean was known exactly, or one could assume that this, too, was uncertain and so was described using a probability density functional. Further, one could assume the correlation function was known or described via a probability density functional. The case considered here is where the correlation parameters, a , b and t are *known* and the mean is given by $h(x) = \sum_k b_k \psi^k(x)$ where the basis functions ψ^k are orthonormal with $\int_{\Omega} \psi^k \psi^l d^D x = \delta_{kl}$. A uniform prior is assumed for the coefficients, b_k with a large negative minimum and large maximum. The maximum probability interpolant is then obtained by maximising the posterior probability density functional. An interesting special case is the minimum curvature method of [5]. A longer discussion of these techniques can be found in [13].

When the prior is a Gaussian probability density functional and the observations are modelled as the values of linear functionals, it follows that the posterior distribution is also Gaussian. Explicit formulae for the posterior mean and the posterior correlation functions can be found in [20] and [27].

4.5 Radial Basis Functions, Kriging, Minimum Curvature and Maximum Probability Interpolation

The details of the maximum probability interpolation method, for the general Gaussian case are provided in [13]. In [13] it is shown that the maximum probability interpolant is the same as the method of Universal Kriging which is, itself, adjoint to general forms of radial basis function interpolation. It has been known for many years that a *dual formulation* of kriging is far more efficient [25]. Although known for a long time, the dual formulation is not widely known, or used. The equivalence of kriging to radial basis functions as a means of interpolation is more widely known, [9].

4.6 Stochastic Sampling Techniques

As stated by Tarantola in [27] it is always worth sampling the probability distributions to increase our intuitive appreciation of the assumptions made

in the prior density through visualisation of realisations. There are many ways of generating realisations, such as the Hastings-Metropolis or the Gibbs sampling methods. See [13] for further references on these classical methods. When the probability density functional is of the local form a particularly convenient method of sampling the distribution is via the *partial differential Langevin equation*

$$\frac{\partial \varphi(x, t)}{\partial \tau} = -\frac{1}{2}L[\varphi] + w$$

where L is the operator defined by equation (12) and $w = w(x, t)$ is a realisation of the white noise process, that is a process with the density

$$\pi(w) = Z \exp\left(-\frac{1}{2} \int_{\Omega \times [0, T]} d^D x d\tau w(x, \tau)^2\right)$$

and where $\tau \in [0, T]$ is a ‘pseudo-time’ or a ‘realisation’ label and Z a normalisation constant. In the limit as $\tau \rightarrow \infty$ it can be shown that the equilibrium density function of the Langevin equation is the expression for the local probability density functional. Proofs of this can be found in [4] and [14].

For exploratory purposes it suffices to solve the Langevin equation with periodic boundary conditions. Generation of white noise is easy, using a Gaussian random pseudo-random number generator with zero mean and a variance of $(h^D \tau)^{-1}$, where h is the grid spacing in x -space and τ is the time step in pseudo-time. Space discretisation is straightforward using central difference formulae for the Laplacian and Biharmonic operators. Although a forward Euler method for the τ derivative will work, it is very slow. For numerical experiments of our own we have found that a backward Euler method, and subsequent solution of the resulting linear equations using a pre-conditioned conjugate gradient method, was very satisfactory.

It should be noted that when the random field is Gaussian and the correlation function is known one can make use of spectral methods. See [23] for further information and examples.

5 Inverse Problems

5.1 Example of a Forward Problem

Rather than describe the idea of an inverse problem in abstract generality (as in the paper [13]) a simple example will be used here. Consider the problem of diffusion in a heterogeneous medium with diffusion coefficient $k(x)$ such that $k = \ln(\varphi)$. The equation for the solution will be

$$\nabla \cdot (k \nabla p) = 0, \quad x \in \Omega. \quad (22)$$

The inverse problem requires determination or at least a characterisation of both functions, k and p when their values are only known at a few points.

Even the boundary conditions for p might be incomplete. Thus some procedure must be invoked to deal with the loss of uniqueness.

5.2 Bayesian Formulation of Inverse Problems

It will be supposed that a mix of known Dirichlet and Neumann boundary conditions are provided, and that the value of k is known at a few points throughout the domain Ω . Further, suppose that some observations, modelled as the values of functionals of p and k are available. Write these diagnostic functionals as $\alpha = D[k, p]$. However, the partial differential equation, equation (22) defines the solution as a functional of the coefficient k and the boundary data. This can be written, therefore, as $\alpha = A[\varphi]$ and the problem is seen to be a generalisation of the scattered data interpolation problem, but now with a *nonlinear* functional A instead of a linear functional.

The Bayesian formulation of the inverse problem is then very similar to the spatial interpolation problem as stated in equation (21). In this more general case, the posterior density functional is given by the equation

$$\pi(\varphi, c | \alpha^*) = \frac{\delta_\sigma(\alpha^* - A[\varphi])\pi(\varphi|c)\pi(c)}{\int_{\mathcal{S}} D[\varphi] dc \delta_\sigma(\alpha^* - A[\varphi])\pi(\varphi|c)\pi(c)} \quad (23)$$

where α^* are the actual *values* of the measurements and δ_σ stands for a product of small-variance Gaussian functionals over the different measurements.

5.3 Tikhonov Regularisation and Local Random Fields

Now consider the case where the prior distribution is of the form equation (10). If the field of properties, φ , is computed so that it maximises the probability density functional then this is equivalent to minimising the ‘energy functional’ or ‘misfit functional’

$$H_T[\varphi] = \sum_i \frac{(\alpha_i^* - A_i[\varphi])^2}{2\sigma_i^2} + H[\varphi] \quad (24)$$

where $H[\varphi]$ is defined by equation (11) and the subscript, i , ranges over a finite number of different measurements.

Using the notation of equation (10), for various choices of a_0 , a_1 and a_2 a variety of well known regularisation procedures are derived. In particular the choices $a_0 > 0$, $a_1 = 0$, $a_2 = 0$ corresponds to ‘Tikhonov order-0’, $a_0 = 0$, $a_1 > 0$, $a_2 = 0$ corresponds to ‘Tikhonov order-1’, and $a_0 = 0$, $a_1 = 0$, $a_2 \geq 0$ corresponds to ‘Tikhonov order-2’ regularisation [28]. In some circumstances the correlation functions of the prior probability densities relating to these choices of regularisation can be found in closed form, as was shown in the paragraph on local random fields in section 4.2.

It thus becomes clear that the classical, Tikhonov, regularisation methods are equivalent to maximum probability Bayesian inversion with a Gaussian

prior. When, for the chosen prior, Green's functions with suitable decay properties at infinity do not exist, we suspect that the results from regularisation will display interesting sensitivities to the size of the computational domain. This has not been investigated as part of the research reported in this paper, and as far as we are aware sensitivity to the size of the domain is not usually investigated in the context of scattered data interpolation or inverse problems. It would perhaps be fruitful to perform more work along this direction.

5.4 Discussion - Inverse Problems and Stochastic Sampling

Attention now turns to a brief discussion about generating samples drawn from the posterior probability density. If our task is just to summarise the distribution via the maximum probability inversion, as described in the previous section, then the Bayesian approach that has been described has the same computational cost as standard minimum misfit approaches. All that has been done is provide a theoretical framework for the choice of the objective (misfit) function and the parameters that appear as weights. One approach is to simply sample from the prior and, by brute computational force, calculate the predicted observations. Then one simply rejects realisations that are too far from the observations. This method (sometimes called 'screening') will work, but is very slow and rather inaccurate because only a small number of samples from the posterior can be obtained. Another approach might be to build an emulator of the forward model and then perform the posterior Monte-Carlo sampling using the emulator - while improving the emulator as the Monte-Carlo proceeds. An investigation of this kind, for low dimensional examples, has been reported in [6, 7]. Much work remains to be done in devising practical methods for summarising a posterior density when the prior involves random fields.

6 Concluding Discussion

This paper provides an introduction to the theory of Gaussian random fields. The treatment, though formal, is given in a continuous, functional analytical, setting. Through this setting one sees simple relationships between the theory of random fields, the theory of Kriging, the theory of radial basis functions, the method of Tikhonov regularisation and Bayesian field theory of inverse problems.

The notion of a local random field - where the correlation function is the Green's function of a differential equation - was emphasised. The correlation functions for several examples of local random fields have been derived. In particular, the correlation function for the Biharmonic-Helmholtz functional has been shown to have quite general qualitative behaviour which is essentially independent of the space dimension.

In some cases, although an expression for the probability density functional can be formulated, the correlation function cannot be found if it is required that it should decay as the radial coordinate tends to infinity. This behaviour does not prevent a maximum probability inversion - which is equivalent to a conventional Tikhonov regularised inversion. It does, however, raise doubts about the formulation of the inverse problem. It is, in the view of the author, likely that superior analyses and decisions will follow when the prior model receives due attention - even if a full Bayesian analysis, involving Monte Carlo functional integration is not performed. By examining the statistical properties of the prior, one might become aware of sensitivities, such as sensitivity to the size of the domain which might otherwise not be investigated. It has been conjectured that one should use functional probability density functions that give rise to well behaved Green's functions on infinite domains, and then geometric sensitivity will not occur.

Our motives for studying local random fields are to (i) understand the relationship between Bayesian inversion and Tikhonov regularisation and (ii) to develop a theory of spatial statistics and scattered data interpolation that does not require constructing global rectangular coordinate systems. When dealing with general systems, such as geological formations with complex faulting, global rectangular coordinate systems do not exist. This was discussed more in [12] but has not been fully explored. Generalisations to the non-Gaussian case would be very interesting and useful, and so there is much research to be done on the theory and application of local random fields in the context of scattered data interpolation and inverse problems.

Acknowledgement

I would like to thank the Royal Society for the award of an Industry Fellowship at the University of Oxford and John Ockendon (Oxford) for stimulating discussions during the writing of this paper.

References

1. M. Abramowitz and I.A. Stegun: *Handbook of Mathematical Functions*. Dover, New York, 1972.
2. G. Barton: *Elements of Green's Functions and Propagation*. Oxford University Press, Oxford, 1999.
3. J.M. Bernardo and F.M. Smith: *Bayesian Theory*. John Wiley & Sons, 2000.
4. J.J. Binney, N.J. Dowrick, A.J. Fisher, and M.E.J. Newman: *The Theory of Critical Phenomena*. Oxford University Press, Oxford, 1992.
5. I.C. Briggs: Machine contouring using minimum curvature. *Geophysics* **39**, 1974, 39-48.
6. D. Busby, C.L. Farmer, and A. Iske: Uncertainty evaluation in reservoir forecasting by Bayes linear methodology. This volume.

7. D. Busby, C.L. Farmer, and A. Iske: Hierarchical nonlinear approximation for experimental design and statistical data fitting. Manuscript, 2005.
8. J.-P. Chilés and P. Delfiner: *Geostatistics: Modeling Spatial Uncertainty*. John Wiley, New York, 1999.
9. O. Dubrule: *Geostatistics for Data Integration in Earth Models*. Distinguished Instructor Series Short Course, no. 6. Society of Exploration Geophysicists, 2003.
10. D.G. Duffy: *Green's Functions with Applications*. Chapman & Hall/CRC, Boca Raton, Florida, 2001.
11. H.W. Engl, M. Hanke, and A. Neubauer: *Regularization of Inverse Problems*. Kluwer, Dordrecht, 1996.
12. C.L. Farmer: Local geostatistics. In: *Proceedings of the 9th European Conference on the Mathematics of Oil Recovery Cannes*, paper A005, September 2004.
13. C.L. Farmer: Geological modelling and reservoir simulation. In: *Mathematical Methods and Modeling in Hydrocarbon Exploration and Production*, A. Iske, T. Randen (eds.), Springer, Berlin, 2005, 119–212.
14. N. Goldenfeld: *Lectures on Phase Transitions and the Renormalization Group*. Perseus Books, Reading, Massachusetts, 1992.
15. I.S. Gradshteyn and I.M. Ryzhik: *Table of Integrals, Series, and Products*. Academic Press, San Diego, 2000.
16. F.G. Horowitz, P. Hornby, D. Bone, and M. Craig: Fast multidimensional interpolations. In: *26th Proceedings of the Application of Computers and Operations Research in the Mineral Industry (APCOM26)*, R.V. Ramani (ed.), Soc. Mining, Metall., and Explor. (SME), Littleton, Colorado, 1996, 53–56.
17. A. O'Hagan: *Kendall's Advanced Theory of Statistics, Volume 2B, Bayesian Inference*. Edward Arnold, London, 1994.
18. A. Iske: *Multiresolution Methods in Scattered Data Modelling*. Springer, Berlin, 2004.
19. J.P. Kaipio and E. Somersalo: *Statistical and Computational Inverse Problems*. Springer, Berlin, 2004.
20. J.C. Lemm: *Bayesian Field Theory*. Johns Hopkins, Baltimore, 2003.
21. J.M.F. Moura, S. Goswami: Gauss-Markov random fields (GMrf) with continuous indices. *IEEE Trans. on Information Theory* **43**(5), 1997, 1560–1573.
22. H. Ockendon and J.R. Ockendon: *Viscous Flow*. Cambridge University Press, 1995.
23. S.M. Prigarin: *Spectral Models of Random Fields in Monte Carlo Methods*. VSP, Utrecht, 2001.
24. B.D. Ripley: *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.
25. J.J. Royer and P.C. Vieira: Dual formalism of kriging. *NATO-ASI Series C*, 122, Pt 2, 1984, 691–702.
26. R. Swinburne (ed.): *Bayes's Theorem*. Oxford University Press, Oxford, 2002.
27. A. Tarantola: *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, Philadelphia, 2005.
28. A.N. Tikhonov and V. Arsenin: *Solution of Ill-posed Problems*. Wiley, New York, 1977.
29. A. Zellner: Bayesian inference. In: *The New Palgrave: Time Series and Statistics*, J. Eatwell, M. Milgate, P. Newman (eds.), Macmillan, 1990, 36–61.

Algorithms for Structured Gauss-Markov Regression

Alistair B. Forbes

National Physical Laboratory, Teddington, Middlesex, TW11 0LW, UK,
alistair.forbes@npl.co.uk

Summary. This paper is concerned with fitting model surfaces to data for which the associated uncertainty matrix is full. From maximum likelihood principles, the best estimates of the model parameters are determined by solving a least squares (Gauss-Markov) regression problem in which the observation equations are weighted by the inverse of the uncertainty matrix. We show that for a significant class of problems, constrained optimisation and separation of variables techniques can be applied, leading to an $\mathcal{O}(m)$ algorithm, where m is the number of data points. Moreover, the techniques can be applied even if the uncertainty matrix is rank deficient, since the algorithm works directly with a factorisation of the uncertainty matrix, rather than its inverse.

1 Introduction

In metrology, fitting a model to data must take into account the uncertainty associated with the data [2, 10, 11, 14]. Suppose data $X = \{\mathbf{x}_i\}$, $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,p})^T \in \mathbb{R}^p$, $i = 1, \dots, m$, represent measurements of quantities $X^* = \{\mathbf{x}_i^*\}$. The random effects associated with X can usually be modelled as multivariate Gaussian noise so the difference between X and X^* is regarded as an mp -vector $\boldsymbol{\epsilon}$ sampled from $N(\mathbf{0}, U_X)$. The $mp \times mp$ uncertainty (variance-covariance) matrix U_X is symmetric and positive semi-definite. The diagonal elements of U_X are the variances associated with the measurements and the off-diagonal elements are the associated covariances. We assume that the model is specified in terms of a parametric surface $\mathbf{f}(\mathbf{u}, \mathbf{b}) : \mathbb{R}^{p-1} \times \mathbb{R}^n \rightarrow \mathbb{R}^p$ of co-dimension 1 in \mathbb{R}^p , where \mathbf{b} are the model parameters. This includes the case of a response model of the form $y = f(\mathbf{u}, \mathbf{b})$, $\mathbf{u} \in \mathbb{R}^{p-1}$, (in parametric form $(\mathbf{u}, \mathbf{b}) \mapsto (\mathbf{u}, f(\mathbf{u}, \mathbf{b}))$) but also the case of parametric surfaces such as paraboloids, parametric spline surfaces, etc., in \mathbb{R}^3 . Our main interest is in the case where the matrix U_X is full but has an underlying structure.

This paper is organised as follows. In Section 2, we show how full uncertainty matrices arise in practice, but that these full uncertainty matrices can have an underlying factorisation structure. The problem of finding best

estimates of the model parameters is discussed in Section 3. In Section 4, we describe a separation of variables approach for the case of block-diagonal uncertainty matrices and show that these apply equally well in the case where the uncertainty matrix is rank deficient. In Section 5, we define a sequential quadratic programming approach to solving the footpoint problem, a key step in the separation of variables approach. We show in Section 6 how the separation of variables approach can be extended, straightforwardly, to deal with the full uncertainty matrices considered in Section 2. Our concluding remarks are given in Section 7.

2 Uncertainty Matrix Associated with Data Points

In this section we consider examples of uncertainty structures that arise using coordinate measuring systems [13].

2.1 Example: Scale and Squareness Model for a Conventional Coordinate Measuring Machine

A conventional coordinate measuring machine (CMM) provides estimates of point coordinates from scale measurements made along three nominally orthogonal axes. Non-ideal motion of the probe system along the three axes can be described by a kinematic model relating to scale, squareness, straightness, roll, pitch and yaw. Various calibration strategies can be implemented to determine and correct for these kinematic errors [3, 9, 20, 25]. However, the kinematic errors are determined from measurements and therefore have uncertainties that contribute to the uncertainties associated with the corrected coordinate values. For example, the contribution of scale and squareness errors can be modelled as

$$\mathbf{x}_i = S\mathbf{x}_i^* + \boldsymbol{\epsilon}_i, \quad S = \begin{bmatrix} 1 + \delta_{xx} & \delta_{xy} & \delta_{xz} \\ 0 & 1 + \delta_{yy} & \delta_{yz} \\ 0 & 0 & 1 + \delta_{zz} \end{bmatrix}, \quad (1)$$

where \mathbf{x}_i^* is the “true” data point, \mathbf{x}_i the measured coordinates, $\delta_{xx} \in N(0, \sigma_{xx}^2)$, etc., represent uncertainties associated with the corrected scale and squareness errors, and $\boldsymbol{\epsilon}_i \in N(0, \sigma^2 I)$ represents random effects associated with the sensor measurements for the i th data point. (The symbol “ \in ” in this context means “is a sample from”, in this case, the normal distribution.) Writing $\boldsymbol{\delta} = (\delta_{xx}, \delta_{yy}, \delta_{zz}, \delta_{xy}, \delta_{xz}, \delta_{yz})^T$, (1) defines $\mathbf{x}_i = \mathbf{x}_i(\boldsymbol{\epsilon}_i, \boldsymbol{\delta})$ as a function of $\boldsymbol{\epsilon}_i$ and $\boldsymbol{\delta}$. If G_i and $G_{0,i}$ are, respectively, the matrices of derivatives of \mathbf{x}_i with respect to $\boldsymbol{\epsilon}_i$ and $\boldsymbol{\delta}$, then the uncertainty matrix U_X associated with X is given by

$$U_X = BB^T, \quad B = \begin{bmatrix} B_1 & & B_{0,1} \\ & \ddots & \vdots \\ & & B_m & B_{0,m} \end{bmatrix}, \quad B_i = G_i D_i, \quad B_{0,i} = G_{0,i} D_0, \quad (2)$$

where D_i is the 3×3 diagonal matrix with σ on the diagonal and D_0 is the 6×6 diagonal matrix with diagonal elements $(\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{xz}, \sigma_{yz})$. The fact that each \mathbf{x}_i depends on $\boldsymbol{\delta}$ means that U_X is a full matrix with potentially significant correlation amongst all the coordinate values. In a more comprehensive model, $\boldsymbol{\delta}$ could represent the residual uncertainty associated with a more comprehensive model of the kinematic errors.

2.2 Example: The Uncertainty Matrix Associated with Laser Tracker Measurements

A laser tracker uses laser interferometric transducers to measure radial displacement and angle encoders to measure azimuth and elevation angles, from which the location \mathbf{x} of a target is estimated. Given a point $\mathbf{p} = (r, \theta, \phi)^T$ defined in spherical coordinates by radius r , azimuth angle θ and elevation angle ϕ , the corresponding Cartesian coordinates $\mathbf{x} = (x, y, z)^T$ are given by

$$(x, y, z) = (r \cos \theta \cos \phi, r \sin \theta \cos \phi, r \sin \phi). \quad (3)$$

In addition, an estimate of the bulk refractive index of the air is required to calculate the effective wavelength of the laser light so that the optical distances (specified in terms of numbers of wavelengths) can be converted into geometric distances. Uncertainties associated with the sensor measurements will propagate through to uncertainties associated with the location of the target. Let $\mathbf{p}_i^* = (r_i^*, \theta_i^*, \phi_i^*)^T$ be the true spherical coordinates associated with a target and $\mathbf{p}_i = (r_i, \theta_i, \phi_i)^T$ the estimate of \mathbf{p}_i^* determined from measurements, $i = 1, \dots, m$. The sources of uncertainty associated with \mathbf{p}_i can be modelled as follows. For the radial distance,

$$r_i^* = l_0^* + l_i^*, \quad r_i = (1 + \omega_0)(l_0 + l_i), \quad l_0 = l_0^* + \delta_0, \quad l_i = l_i^* + \delta_i,$$

where l_0^* is the true deadpath, l_i^* the true displacement, and ω_0 , δ_0 and δ_i represent random effects, and are modelled as samples from normal distributions. The inclusion of the term l_0 representing the laser deadpath reflects the fact that the interferometric transducers measure the *change* in distance. The laser deadpath is the distance to the target when the interferometer count is set to zero at the start of the measurement cycle; it has to be estimated through a calibration procedure. The term ω_0 represents the uncertainty contribution arising from the measurement of the refractive index of the air.

For the azimuth and elevation angle measurements,

$$\theta_i = \theta_i^* + \epsilon_0 + \epsilon_i, \quad \phi_i = \phi_i^* + \rho_0 + \rho_i,$$

where ϵ_0 , ρ_0 represent uncertainty in the alignment of the angle encoders and ϵ_i and ρ_i represent random effects associated with the sensor readings. Along with (3), these equations define $\mathbf{x}_i = \mathbf{x}_i(\boldsymbol{\epsilon}_i, \boldsymbol{\delta})$ as functions of $\boldsymbol{\epsilon}_i = (\delta_i, \epsilon_i, \rho_i)^T$, and $\boldsymbol{\delta} = (\omega_0, \delta_0, \epsilon_0, \rho_0)^T$. The uncertainty matrix U_X associated with measurements X is constructed exactly as in (2) using the appropriate derivative and uncertainty matrices associated with $\boldsymbol{\epsilon}_i$ and $\boldsymbol{\delta}$. In practice, the uncertainty associated with the laser deadpath l_0 is often very significant so that the correlation is substantial and inferences based on an assumption of independence are likely to be unreliable. Note that even if $\boldsymbol{\delta}$ is known to be identically zero, the 3×3 uncertainty matrix associated with \mathbf{x}_i is full, since the Cartesian coordinates depend on multiple sensor readings.

2.3 Structural Correlation in Uncertainty Matrices

In the examples above, uncertainty matrices U_X were full through a dependence of all of the measurements \mathbf{x}_i on common systematic effects $\boldsymbol{\delta}$. The examples above are taken from coordinate metrology but the dependence on common effects occurs throughout metrology. For example, measurements of both response and stimulus variables are often temperature-corrected, giving rise to a common dependence on the temperature measurement. If the dependence of the measurements \mathbf{x}_i on stochastic effects can be written as $\mathbf{x}_i = \mathbf{x}_i(\boldsymbol{\epsilon}_i, \boldsymbol{\delta})$, then the associated uncertainty matrix U_X can be factored as in (2). This is one of the most common ways in which full uncertainty matrices arise in regression problems. We note that for this type of uncertainty structure, if there are m data points, U_X is specified by $\mathcal{O}(m)$ elements. Rarely, if at all, do uncertainty matrices require $\mathcal{O}(m^2)$ independent elements.

3 Fitting Parametric Surfaces to Data

Let $\mathbf{f}(\mathbf{u}, \mathbf{b}) : \mathbb{R}^{p-1} \times \mathbb{R}^n \rightarrow \mathbb{R}^p$ define a parametric surface in \mathbb{R}^p . We refer to the parameters \mathbf{u} as the *footpoint parameters* and \mathbf{b} as the *surface* (shape) parameters. We assume the parameterization is regular so that the $p \times (p-1)$ matrix $F_{\mathbf{u}}$ of partial derivatives $\partial \mathbf{f} / \partial u_k$ has full column rank. If \mathbf{n} is the orthogonal complement to $F_{\mathbf{u}}$ in \mathbb{R}^p , then \mathbf{n} is orthogonal to the surface at \mathbf{u} . Let X be measurements of X^* , the coordinates of points $\{\mathbf{x}_i^*\}_{i=1}^m$ lying on the surface, and let U_X be the uncertainty matrix associated with X .

If U_X is nonsingular, setting \mathbf{a} to be the $(p-1)m+n$ vector of parameters $\{\mathbf{u}_i\}$ and \mathbf{b} , an estimate of the parameters is given by the solution of

$$\min_{\mathbf{a}} \mathbf{e}^T(\mathbf{a})U_X^{-1}\mathbf{e}(\mathbf{a}), \quad (4)$$

where $\mathbf{e}(\mathbf{a})$ is the pm -vector of residuals $\mathbf{e}_i(\mathbf{a}) = \mathbf{x}_i - \mathbf{f}(\mathbf{u}_i, \mathbf{b})$. If the random effects in the data are modelled as multivariate Gaussian noise with variance matrix U_X , the solution of (4) is the maximum likelihood estimate, i.e., the

value of the parameters that gives the most probable explanation of the measurement data. Each data point is weighted in relation to the degree of belief, as represented by U_X , we have in the measurements. In the case of linear regression, the Gauss-Markov theorem states that the solution of (4) is the best linear unbiased estimate [21].

3.1 Full Matrix Approaches

If U_X is the identity matrix, then the Gauss-Newton algorithm can be applied directly to solve (4) [15]. If J is the matrix of partial derivatives of \mathbf{e} with respect to \mathbf{a} , then an updated estimate of \mathbf{a} is given by $\mathbf{a} := \mathbf{a} + \mathbf{p}$ where \mathbf{p} solves the linear least squares problem

$$\min_{\mathbf{p}} \|\mathbf{e} + J\mathbf{p}\|_2^2.$$

If J has QR factorisation $J = QR$ [16], where Q is an orthogonal matrix of the same dimension as J and R is upper triangular, then $R\mathbf{p} = -Q^T\mathbf{e}$.

For general U_X , if U_X has Cholesky factorisation [16] $U_X = L_X L_X^T$, then the solution of (4) solves the modified nonlinear least squares problem

$$\min_{\mathbf{a}} \tilde{\mathbf{e}}^T(\mathbf{a})\tilde{\mathbf{e}}(\mathbf{a}), \quad \tilde{\mathbf{e}}(\mathbf{a}) = L_X^{-1}\mathbf{e}(\mathbf{a}), \quad (5)$$

which can again be solved using the Gauss-Newton algorithm. If J is the Jacobian matrix associated with \mathbf{e} then $\tilde{J} = L_X^{-1}J$ is that associated with $\tilde{\mathbf{e}}$. The presence of L_X^{-1} in the formulation can lead to numerical stability issues if U_X is poorly conditioned. If U_X is singular, then another approach to determining appropriate estimates of the model parameters is necessary.

Suppose U_X has factorisation $U_X = BB^T$. Then (4) can be reformulated as

$$\min_{\mathbf{a}} \boldsymbol{\alpha}^T \boldsymbol{\alpha} \quad \text{subject to} \quad \mathbf{e}(\mathbf{a}) = B\boldsymbol{\alpha}. \quad (6)$$

If B is the Cholesky factor of U_X then $\boldsymbol{\alpha} = B^{-1}\mathbf{e}$ and the equivalence of (6) with (5) is clear. However, formulation (6) still makes sense if U_X is singular, a case that arises in practice, or if B is non-square, as in the examples in Section 2. In either case, the solution of (6) provides maximum likelihood estimates of the parameters.

The Gauss-Newton algorithm can be adapted to solve (6). If J is the Jacobian matrix associated with $\mathbf{e} = \mathbf{e}(\mathbf{a})$ then the update step \mathbf{p} for \mathbf{a} solves

$$\min_{\mathbf{a}} \boldsymbol{\alpha}^T \boldsymbol{\alpha} \quad \text{subject to} \quad \mathbf{e} = -J\mathbf{p} + B\boldsymbol{\alpha}. \quad (7)$$

The generalised QR factorisation [17, 22] can be used to determine \mathbf{p} and involves the QR factorisation of $J = QR$ and the RQ factorisation of $Q^T B = TP$ where R and T are upper-triangular and Q and P are orthogonal [23]. We note again that (7) can be solved even if B is singular.

These full matrix approaches are problematic if m is large. There are $\mathcal{O}(m)$ observations and $\mathcal{O}(m)$ parameters which leads to an $\mathcal{O}(m^3)$ algorithm, since the various factorisations require $\mathcal{O}(m^3)$ steps.

4 Generalised Distance Regression

If there is no statistical correlation between the the i th and q th measurements \mathbf{x}_i and \mathbf{x}_q , $i \neq q$, then U_X is a block diagonal matrix with $p \times p$ matrices $U_i = B_i B_i^T$ along its diagonal:

$$U_X = \begin{bmatrix} U_1 & & & & \\ & \ddots & & & \\ & & U_i & & \\ & & & \ddots & \\ & & & & U_m \end{bmatrix}, \quad U_i = B_i B_i^T.$$

This case is important in that many problems have this uncertainty structure (at least to a good approximation) but also because the techniques developed for its efficient solution can also be applied to the more general uncertainty matrices considered in Section 2. In the block-diagonal case, (4) decomposes as

$$\min_{\mathbf{a}} \sum_{i=1}^m \mathbf{e}_i^T(\mathbf{a}) U_i^{-1} \mathbf{e}_i(\mathbf{a}), \quad \mathbf{e}_i(\mathbf{a}) = \mathbf{x}_i - \mathbf{f}(\mathbf{u}_i, \mathbf{b}).$$

If U_i has Cholesky factorisation $U_i = L_i L_i^T$, then corresponding to (5), we solve

$$\min_{\mathbf{a}} \sum_{i=1}^m \tilde{\mathbf{e}}_i^T(\mathbf{a}) \tilde{\mathbf{e}}_i(\mathbf{a}), \quad \tilde{\mathbf{e}}_i(\mathbf{a}) = L_i^{-1} \mathbf{e}_i(\mathbf{u}_i, \mathbf{b}). \quad (8)$$

4.1 Exploiting Block Angular Structure of Jacobian Matrix

Since each $\tilde{\mathbf{e}}_i$ in (8) involves only one set of footpoint parameters, the Jacobian matrix and its upper-triangular factor R have a block-angular structure:

$$J = \begin{bmatrix} J_1 & & J_{0,1} \\ & \ddots & \vdots \\ & & J_m \ J_{0,m} \end{bmatrix}, \quad R = \begin{bmatrix} R_1 & & R_{0,1} \\ & \ddots & \vdots \\ & & R_m \ R_{0,m} \\ & & & R_0 \end{bmatrix}.$$

(Many data analysis problems in metrology have this structure [8, 11, 12].) An efficient ($\mathcal{O}(m)$) algorithm [4, 6, 7, 24] can be designed to perform the QR factorisation operating on only $p+n$ rows and $p+n-1$ columns at a time:

$$Q_i^T \begin{bmatrix} J_i & J_{0,i} \\ & R_0 \end{bmatrix} =: \begin{bmatrix} R_i & R_{0,i} \\ & R_0 \\ & & 0 \end{bmatrix};$$

here, R_0 in the righthand side is the update of R_0 in the lefthand side.

4.2 Separation of Variables Approach: Full Rank Case

As an alternative to using structured matrix factorisation techniques, a separation of variables approach can be used [1, 5, 18, 19]. Let $M_i = U_i^{-1}$ and suppose \mathbf{u}_i^* solves the i th footpoint problem

$$\min_{\mathbf{u}_i} (\mathbf{x}_i - \mathbf{f}(\mathbf{u}_i, \mathbf{b}))^T M_i (\mathbf{x}_i - \mathbf{f}(\mathbf{u}_i, \mathbf{b})), \quad (9)$$

defining $\mathbf{u}_i^* = \mathbf{u}_i^*(\mathbf{b})$ as a function of \mathbf{b} . Setting

$$d_i^2(\mathbf{b}) = (\mathbf{x}_i - \mathbf{f}_i^*(\mathbf{b}))^T M_i (\mathbf{x}_i - \mathbf{f}_i^*(\mathbf{b})), \quad \mathbf{f}_i^*(\mathbf{b}) = \mathbf{f}(\mathbf{u}_i^*(\mathbf{b}), \mathbf{b}), \quad (10)$$

also a function of \mathbf{b} , the values of \mathbf{b} which solve (8) are the same as those that solve

$$\min_{\mathbf{b}} \sum_{i=1}^m d_i^2(\mathbf{b}). \quad (11)$$

This means that (8) can be solved as a standard nonlinear least squares problem. The quantity $d_i(\mathbf{b})$ is the *generalised distance* of the data point \mathbf{x}_i from the surface $\mathbf{f}(\mathbf{u}, \mathbf{b})$ defined using the metric matrix M_i .

To use the Gauss-Newton algorithm to solve (11), we need to be able to calculate the partial derivatives $\partial d_i / \partial b_j$ which, at first sight, involves the calculation of $\partial \mathbf{u}_i^* / \partial b_j$. However, the conditions that \mathbf{u}_i^* is a solution of (9) imply that

$$\left(\frac{\partial \mathbf{f}}{\partial u_k} \right)^T M_i (\mathbf{x}_i - \mathbf{f}_i^*) = 0, \quad k = 1, \dots, p-1,$$

showing that $M_i(\mathbf{x}_i - \mathbf{f}_i^*)$ is orthogonal to the surface at $\mathbf{f}_i^* = \mathbf{f}_i^*(\mathbf{b})$ (since it is orthogonal to the $p-1$ tangent vectors $\partial \mathbf{f} / \partial u_k$ which are assumed to be linearly independent). Differentiating $d_i^2(\mathbf{b})$ in (10) with respect to b_j we have

$$2d_i \frac{\partial d_i}{\partial b_j} = -2 \left\{ \frac{\partial \mathbf{f}}{\partial b_j} + \left(\sum_{k=1}^{p-1} \frac{\partial u_k^*}{\partial b_j} \frac{\partial \mathbf{f}}{\partial u_k} \right) \right\}^T M_i (\mathbf{x}_i - \mathbf{f}_i^*),$$

and, since $\partial \mathbf{f} / \partial u_k$ are orthogonal to $M_i(\mathbf{x}_i - \mathbf{f}_i^*)$, we see that

$$\frac{\partial d_i}{\partial b_j} = -\frac{1}{d_i} \left(\frac{\partial \mathbf{f}}{\partial b_j} \right)^T M_i (\mathbf{x}_i - \mathbf{f}_i^*), \quad j = 1, \dots, n,$$

and involves only the partial derivatives of \mathbf{f} with respect to b_j . This formula for the derivatives is not well defined if $d_i = 0$. To cover this case, let \mathbf{n}_i be any non-zero vector orthogonal to the surface at \mathbf{f}_i^* , for example, the null vector of the $p \times (p-1)$ matrix $F_{\mathbf{u}} = \nabla_{\mathbf{u}^T} \mathbf{f}$. It is straightforward to check that if

$$w_i = (\mathbf{n}_i^T U_i \mathbf{n}_i)^{1/2}, \quad (12)$$

then

$$d_i(\mathbf{b}) = \frac{1}{w_i} \mathbf{n}_i^T (\mathbf{x}_i - \mathbf{f}_i^*), \quad \frac{\partial d_i}{\partial b_j} = -\frac{1}{w_i} \mathbf{n}_i^T \left(\frac{\partial \mathbf{f}}{\partial b_j} \right). \quad (13)$$

To summarise, using the separation of variables approach, (8) can be solved as a standard nonlinear least squares problem (11) in the n parameters \mathbf{b} . Since each iteration takes $\mathcal{O}(mn^2)$ and convergence is expected to be linear, the algorithm is $\mathcal{O}(m)$. The only complication is that the evaluation of the functions $d_i(\mathbf{b})$ involve the calculation of the footpoint parameters. We describe a compact and quadratically converging algorithm for solving the footpoint problem in Section 5.

4.3 Separation of Variables Approach: Rank Deficient Case

The formulae (13) for calculating d_i and its derivatives involve U_i (to calculate w_i in (12)), not its inverse. Using the factorisation $U_i = B_i B_i^T$, the optimal footpoint parameters \mathbf{u}_i^* can be determined by solving

$$\min_{\mathbf{u}_i} \boldsymbol{\alpha}_i^T \boldsymbol{\alpha}_i \quad \text{subject to} \quad \mathbf{x}_i = \mathbf{f}(\mathbf{u}_i, \mathbf{b}) + B_i \boldsymbol{\alpha}_i, \quad (14)$$

again, avoiding the calculation of the inverse of U_i . (We refer to (9) as the *direct footpoint problem* and (14) above as the *generalised footpoint problem*.) In fact, formulae (13) hold even if U_i is singular, as we will now show.

Dropping subscript i in (14), suppose U has rank r , $1 \leq r < p$, and eigenvalue decomposition

$$U = PS^2P^T = (PS)(PS)^T,$$

where S is a diagonal matrix with nonzero values in the first r diagonal elements and zeros everywhere else and P is a $p \times p$ orthogonal matrix. We partition S , P and $\boldsymbol{\alpha}$ as

$$S = \begin{bmatrix} S_1 & 0 \\ 0 & 0 \end{bmatrix}, \quad P = [P_1 \ P_2], \quad \boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{\alpha}_1 \\ \boldsymbol{\alpha}_2 \end{bmatrix}.$$

If we multiply the equation $\mathbf{x} = \mathbf{f} + PS\boldsymbol{\alpha}$ by P^T , it partitions as

$$P^T \mathbf{x} = P^T \mathbf{f} + \begin{bmatrix} S_1 \boldsymbol{\alpha}_1 \\ 0 \end{bmatrix}.$$

The rank deficient case of the generalised footpoint problem can therefore be presented as

$$\min_{\mathbf{u}, \mathbf{v}} \boldsymbol{\alpha}^T \boldsymbol{\alpha} \quad \text{subject to} \quad \mathbf{x} = \mathbf{f}(\mathbf{u}, \mathbf{v}, \mathbf{b}) + B\boldsymbol{\alpha}, \quad \mathbf{y} = \mathbf{g}(\mathbf{u}, \mathbf{v}, \mathbf{b}), \quad (15)$$

where B is an $r \times r$ invertible matrix and \mathbf{u} and \mathbf{v} are footpoint components with $r-1$ and t parameters, respectively.

The second set of t constraints defines the t parameters $\tilde{\mathbf{v}} = \tilde{\mathbf{v}}(\mathbf{u}, \mathbf{b})$ as functions of \mathbf{u} and \mathbf{b} . Let $\tilde{\mathbf{f}}(\mathbf{u}, \mathbf{b}) = \mathbf{f}(\mathbf{u}, \tilde{\mathbf{v}}(\mathbf{u}, \mathbf{b}), \mathbf{b})$. Then (15) can be reformulated as

$$\min_{\mathbf{u}} \boldsymbol{\alpha}^T \boldsymbol{\alpha} \quad \text{subject to} \quad \mathbf{x} = \tilde{\mathbf{f}}(\mathbf{u}, \mathbf{b}) + B\boldsymbol{\alpha}, \quad (16)$$

i.e., as a full rank footpoint problem (in \mathbb{R}^r) already considered. Thus, if \mathbf{u}^* solves (16) and $\tilde{\mathbf{n}}$ is orthogonal to the surface $\mathbf{x} = \tilde{\mathbf{f}}$ at \mathbf{u}^* then

$$d(\mathbf{b}) = \frac{1}{\tilde{w}} \tilde{\mathbf{n}}^T (\mathbf{x} - \tilde{\mathbf{f}}), \quad \frac{\partial d}{\partial b_j} = -\frac{1}{\tilde{w}} \tilde{\mathbf{n}}^T \left(\frac{\partial \tilde{\mathbf{f}}}{\partial b_j} \right), \quad \tilde{w} = (\tilde{\mathbf{n}}^T B B^T \tilde{\mathbf{n}})^{1/2}. \quad (17)$$

We now wish to show that the formula (13) applied to (15), a generalised footpoint problem in \mathbb{R}^p corresponding to the $p \times p$ uncertainty matrix

$$U = \begin{bmatrix} B B^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

gives the same results as (17). That is, if $\begin{bmatrix} \mathbf{n} \\ \mathbf{m} \end{bmatrix}$ is orthogonal to the surface

$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{f}(\mathbf{u}, \mathbf{v}, \mathbf{a}) \\ \mathbf{g}(\mathbf{u}, \mathbf{v}, \mathbf{a}) \end{bmatrix}$ at the solution footpoint $(\mathbf{u}^*, \tilde{\mathbf{v}}(\mathbf{u}^*, \mathbf{a}))$ then $d(\mathbf{b})$ and $\partial d / \partial b_j$ can also be calculated from

$$d(\mathbf{b}) = \frac{1}{w} \begin{bmatrix} \mathbf{n} \\ \mathbf{m} \end{bmatrix}^T \begin{bmatrix} \mathbf{x} - \mathbf{f} \\ \mathbf{y} - \mathbf{g} \end{bmatrix}, \quad w = \left(\begin{bmatrix} \mathbf{n} \\ \mathbf{m} \end{bmatrix}^T U \begin{bmatrix} \mathbf{n} \\ \mathbf{m} \end{bmatrix} \right)^{1/2},$$

and

$$\frac{\partial d}{\partial b_j} = -\frac{1}{w} \begin{bmatrix} \mathbf{n} \\ \mathbf{m} \end{bmatrix}^T \begin{bmatrix} \partial \mathbf{f} / \partial b_j \\ \partial \mathbf{g} / \partial b_j \end{bmatrix},$$

respectively. We show first that \mathbf{n} is orthogonal to the surface $\mathbf{x} = \tilde{\mathbf{f}}(\mathbf{u}, \mathbf{b})$ at \mathbf{u}^* . Regarding $\tilde{\mathbf{v}}$ as a function of \mathbf{u} and \mathbf{b} , let $F_{\mathbf{u}}$ be the $r \times (r-1)$ matrix of partial derivatives of \mathbf{f} with respect to the parameters \mathbf{u} , similarly $F_{\mathbf{v}}$ and $F_{\mathbf{b}}$. Let $\tilde{F}_{\mathbf{u}}$ and $\tilde{F}_{\mathbf{b}}$ be the corresponding matrices for $\tilde{\mathbf{f}}$. Likewise, let $G_{\mathbf{u}}$ be the $t \times (r-1)$ matrix of partial derivatives of \mathbf{g} with respect to the parameters \mathbf{u} with $G_{\mathbf{v}}$ and $G_{\mathbf{b}}$ defined similarly. Finally, let $V_{\mathbf{u}}$ be the $t \times (r-1)$ matrix of partial derivatives of $\tilde{\mathbf{v}}$ with respect to \mathbf{u} and define $V_{\mathbf{b}}$ similarly. Then

$$\begin{aligned} V_{\mathbf{u}} &= -G_{\mathbf{v}}^{-1} G_{\mathbf{u}}, & V_{\mathbf{b}} &= -G_{\mathbf{v}}^{-1} G_{\mathbf{b}}, \\ \tilde{F}_{\mathbf{b}} &= F_{\mathbf{b}} + F_{\mathbf{v}} V_{\mathbf{u}} = F_{\mathbf{b}} - F_{\mathbf{v}} G_{\mathbf{v}}^{-1} G_{\mathbf{b}}, \end{aligned}$$

and

$$\tilde{F}_{\mathbf{u}} = F_{\mathbf{u}} + F_{\mathbf{v}} V_{\mathbf{u}} = F_{\mathbf{u}} - F_{\mathbf{v}} G_{\mathbf{v}}^{-1} G_{\mathbf{u}}.$$

The fact that $(\mathbf{n}^T, \mathbf{m}^T)^T$ is orthogonal to the surface can be stated as

$$\mathbf{n}^T F_{\mathbf{u}} = -\mathbf{m}^T G_{\mathbf{u}} \quad \text{and} \quad \mathbf{n}^T F_{\mathbf{v}} = -\mathbf{m}^T G_{\mathbf{v}}.$$

From these relationships, we have

$$\begin{aligned}\mathbf{n}^T \tilde{F}_{\mathbf{u}} &= \mathbf{n}^T F_{\mathbf{u}} - \mathbf{n}^T F_{\mathbf{v}} G_{\mathbf{v}}^{-1} G_{\mathbf{u}}, \\ &= \mathbf{n}^T F_{\mathbf{u}} + \mathbf{m}^T G_{\mathbf{v}} G_{\mathbf{v}}^{-1} G_{\mathbf{u}}, \\ &= \mathbf{n}^T F_{\mathbf{u}} + \mathbf{m}^T G_{\mathbf{u}} = 0,\end{aligned}$$

showing that \mathbf{n} must be a multiple of $\tilde{\mathbf{n}}$. Furthermore, up to sign, $\tilde{w}\mathbf{n} = w\tilde{\mathbf{n}}$. Since at the solution of the footpoint problem (15), $\mathbf{y} = \mathbf{g}$,

$$\begin{bmatrix} \mathbf{n} \\ \mathbf{m} \end{bmatrix}^T \begin{bmatrix} \mathbf{x} - \mathbf{f} \\ \mathbf{y} - \mathbf{g} \end{bmatrix} = \mathbf{n}^T (\mathbf{x} - \mathbf{f}) = \mathbf{n}^T (\mathbf{x} - \tilde{\mathbf{f}}),$$

and

$$\begin{bmatrix} \mathbf{n} \\ \mathbf{m} \end{bmatrix}^T U \begin{bmatrix} \mathbf{n} \\ \mathbf{m} \end{bmatrix} = \mathbf{n}^T B B^T \mathbf{n}.$$

Therefore,

$$\frac{1}{w} \begin{bmatrix} \mathbf{n} \\ \mathbf{m} \end{bmatrix}^T \begin{bmatrix} \mathbf{x} - \mathbf{f} \\ \mathbf{y} - \mathbf{g} \end{bmatrix} = \frac{1}{\tilde{w}} \tilde{\mathbf{n}}^T (\mathbf{x} - \tilde{\mathbf{f}}),$$

confirming the equivalence of the function evaluations, up to sign. Similarly,

$$\begin{aligned}\mathbf{n}^T \tilde{F}_{\mathbf{b}} &= \mathbf{n}^T F_{\mathbf{b}} - \mathbf{n}^T F_{\mathbf{v}} G_{\mathbf{v}}^{-1} G_{\mathbf{b}}, \\ &= \mathbf{n}^T F_{\mathbf{b}} + \mathbf{m}^T G_{\mathbf{v}} G_{\mathbf{v}}^{-1} G_{\mathbf{b}}, \\ &= \begin{bmatrix} \mathbf{n} \\ \mathbf{m} \end{bmatrix}^T \begin{bmatrix} F_{\mathbf{b}} \\ G_{\mathbf{b}} \end{bmatrix},\end{aligned}$$

from which we can confirm the equivalence of the derivative calculations, up to sign.

Example: Surface Fit in \mathbb{R}^3

Consider the generalised footpoint problem,

$$\min_{u,v} \{\alpha^2 + \beta^2\}$$

subject to the constraints

$$x = f(u, v, \mathbf{b}) + \alpha, \quad y = g(u, v, \mathbf{b}) + \beta, \quad z = h(u, v, \mathbf{b}).$$

Let $\mathbf{n}^T = (g_u h_v - g_v h_u, f_v h_u - f_u h_v, f_u g_v - f_v g_u)$, the vector cross-product of $(f_u, g_u, h_u)^T$ with $(f_v, g_v, h_v)^T$, where $f_u = \partial f / \partial u$, etc. The vector \mathbf{n} is orthogonal to the surface at (u, v) . The formula for $d(\mathbf{b})$ for a surface in \mathbb{R}^3 is

$$d(\mathbf{b}) = \frac{(g_u h_v - g_v h_u)(x - f) + (f_v h_u - f_u h_v)(y - g)}{[(g_u h_v - g_v h_u)^2 + (f_v h_u - f_u h_v)^2]^{1/2}},$$

evaluated at the solution (u^*, v^*) of the footpoint problem. Alternatively, the equation $z = h(u, v, \mathbf{a})$ defines $\tilde{v} = \tilde{v}(u, \mathbf{a})$ as a function of u and \mathbf{a} , and we consider the equivalent footpoint problem

$$\min_{u,v} \{\alpha^2 + \beta^2\} \quad \text{subject to} \quad x = \tilde{f}(u, \mathbf{b}) + \alpha, \quad y = \tilde{g}(u, \mathbf{b}) + \beta,$$

where $\tilde{f}(u, \mathbf{a}) = f(u, \tilde{v}(u, \mathbf{a}), \mathbf{a})$, etc. Let $\tilde{\mathbf{n}}^T = (-\tilde{g}_u, \tilde{f}_u)$, orthogonal to the curve $(\tilde{f}(u), \tilde{g}(u))$ at u . The resulting formula for $d(\mathbf{b})$ for a curve in \mathbb{R}^2 is

$$d(\mathbf{b}) = \frac{-\tilde{g}_u(x - \tilde{f}) + \tilde{f}_u(y - \tilde{g})}{(\tilde{f}_u^2 + \tilde{g}_u^2)^{1/2}}.$$

The equivalence of the two formulae follows from the fact that

$$\tilde{f}_u = f_u + f_v \frac{\partial \tilde{v}}{\partial u} = f_u - f_v h_u / h_v, \quad \tilde{g}_u = g_u + g_v \frac{\partial \tilde{v}}{\partial u} = g_u - g_v h_u / h_v.$$

5 Solution of the Generalised Footpoint Problem

In this section we describe a sequential quadratic programming algorithm to solve the generalised footpoint problem (14), treating it as a nonlinearly constrained optimisation problem. We first review the relevant optimisation techniques.

5.1 Quadratic Programming

Let A be an $n \times n$ positive definite, symmetric matrix, C a $p \times n$ matrix, $p < n$ and \mathbf{b} and \mathbf{d} n - and p -vectors, respectively. The quadratic programming problem is

$$\min_{\boldsymbol{\xi}} \frac{1}{2} \boldsymbol{\xi}^T A \boldsymbol{\xi} + \mathbf{b}^T \boldsymbol{\xi} \quad \text{subject to} \quad C \boldsymbol{\xi} = \mathbf{d}. \quad (18)$$

Using a Lagrangian formulation in which we look for a stationary point of

$$\mathcal{L}(\boldsymbol{\xi}, \boldsymbol{\lambda}) = \frac{1}{2} \boldsymbol{\xi}^T A \boldsymbol{\xi} + \mathbf{b}^T \boldsymbol{\xi} - (C \boldsymbol{\xi} - \mathbf{d})^T \boldsymbol{\lambda},$$

we find that $\boldsymbol{\xi}$ and $\boldsymbol{\lambda}$ must solve

$$\begin{bmatrix} A & -C^T \\ -C & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi} \\ \boldsymbol{\lambda} \end{bmatrix} = - \begin{bmatrix} \mathbf{b} \\ \mathbf{d} \end{bmatrix}, \quad (19)$$

involving the *Lagrangian matrix* on the lefthand side. Therefore one approach to solving (18) is to solve the $(n+p) \times (n+p)$ system of equations. We note that although the Lagrangian matrix is symmetric, generally it will not be positive definite and so a Cholesky factorisation approach cannot be applied.

Using generalised constraint elimination, the linear constraints are used to redefine the problem in terms of an unconstrained quadratic problem in $n - p$ variables. One approach is as follows. Let

$$C^T = [Q_1 \ Q_2] \begin{bmatrix} R \\ \mathbf{0} \end{bmatrix} = Q_1 R,$$

be the QR factorisation of C^T , where R is $p \times p$ upper triangular and $Q = [Q_1 \ Q_2]$ is an $n \times n$ orthogonal matrix. We look for a solution of (18) of the form $\boldsymbol{\xi} = Q_1 \boldsymbol{\xi}_1 + Q_2 \boldsymbol{\xi}_2$. From the constraint equation we have

$$\mathbf{d} = C(Q_1 \boldsymbol{\xi}_1 + Q_2 \boldsymbol{\xi}_2) = R^T Q_1^T Q_1 \boldsymbol{\xi}_1 + R^T Q_1^T Q_2 \boldsymbol{\xi}_2 = R^T \boldsymbol{\xi}_1,$$

since $Q_1^T Q_1 = I$ and $Q_1^T Q_2 = \mathbf{0}$. This shows that $\boldsymbol{\xi}_1$ must satisfy $R^T \boldsymbol{\xi}_1 = \mathbf{d}$. These constraints fix $\boldsymbol{\xi}_1$ and we must choose $\boldsymbol{\xi}_2$ to minimise the quadratic expression which amounts to minimising

$$\frac{1}{2} \boldsymbol{\xi}_2^T Q_2^T A Q_2 \boldsymbol{\xi}_2 + \boldsymbol{\xi}_2^T Q_2^T (\mathbf{b} + A Q_1 \boldsymbol{\xi}_1)$$

with respect to $\boldsymbol{\xi}_2$. The conditions for a minimum dictate that $\boldsymbol{\xi}_2$ solves the system

$$Q_2^T A Q_2 \boldsymbol{\xi}_2 = -Q_2^T (\mathbf{b} + A Q_1 \boldsymbol{\xi}_1),$$

where $Q_2^T A Q_2$ is a $(p - n) \times (p - n)$ symmetric, positive definite matrix. This system can be solved using a Cholesky factorisation approach. If required, the Lagrange multipliers $\boldsymbol{\lambda}$ can be determined as the solution of

$$C^T \boldsymbol{\lambda} = \mathbf{b} + A \boldsymbol{\xi},$$

or, using the factorisation of C^T , $R \boldsymbol{\lambda} = Q_1^T (\mathbf{b} + A \boldsymbol{\xi})$.

5.2 Sequential Quadratic Programming

Now consider the nonlinearly constrained optimisation problem

$$\min_{\boldsymbol{\xi}} F(\boldsymbol{\xi}) \quad \text{subject to} \quad c_k(\boldsymbol{\xi}) = 0, \quad k = 1, \dots, p.$$

The solution $\boldsymbol{\xi}^*$ defines a stationary point of the Lagrangian

$$\mathcal{L}(\boldsymbol{\xi}, \boldsymbol{\lambda}) = F(\boldsymbol{\xi}) - \sum_{k=1}^p \lambda_k c_k(\boldsymbol{\xi}).$$

Suppose $\boldsymbol{\lambda}^*$ are the solution Lagrange multipliers and that $\boldsymbol{\xi}$ is an approximation to the solution $\boldsymbol{\xi}^*$. Linearising the conditions

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\xi}} = \mathbf{0}, \quad c_k(\boldsymbol{\xi}) = 0, \quad k = 1, \dots, p,$$

about $\boldsymbol{\xi}$ yields

$$\nabla F + \nabla^2 F \mathbf{p} - \sum_{k=1}^p \lambda_k^* \{ \nabla c_k + \nabla^2 c_k \mathbf{p} \} = \mathbf{0}, \quad c_k(\boldsymbol{\xi}) + \nabla c_k \mathbf{p} = 0.$$

Setting

$$A = \nabla^2 F - \sum_{k=1}^p \lambda_k^* \nabla^2 c_k, \quad C_{kj} = \frac{\partial c_k}{\partial \xi_j},$$

these equations can be written as $A\mathbf{p} - C^T \boldsymbol{\lambda}^* = -\mathbf{g}$, $C\mathbf{p} = -\mathbf{c}$, where $\mathbf{g} = \nabla F$. Comparing these equations with (19), we see that the update step \mathbf{p} for $\boldsymbol{\xi}$ is the solution of the quadratic programming problem

$$\min_{\mathbf{p}} \frac{1}{2} \mathbf{p}^T A \mathbf{p} + \mathbf{g}^T \mathbf{p} \quad \text{subject to} \quad C\mathbf{p} = -\mathbf{c}.$$

The solution of the quadratic programming problem also provides updated estimates of the Lagrange multipliers $\boldsymbol{\lambda}$.

5.3 Sequential Quadratic Programming for the Footpoint Parameters

The sequential quadratic programming (SQP) approach can be applied to solve the generalised footpoint problem (14) as follows. Given estimates $\boldsymbol{\alpha}_q$, $\boldsymbol{\lambda}_q$ and \mathbf{u}_q ,

1. Evaluate the surface function and gradient: $\mathbf{f} = \mathbf{f}(\mathbf{u}_q, \mathbf{b})$, $F_{\mathbf{u}} = \nabla_{\mathbf{u}}^T \mathbf{f}$.
2. Evaluate the objective function gradient: $\mathbf{g} = \begin{bmatrix} \boldsymbol{\alpha}_q \\ \mathbf{0} \end{bmatrix}$.
3. Evaluate the constraint function and gradient: $\mathbf{c} = B\boldsymbol{\alpha}_q + \mathbf{f} - \mathbf{x}$, $C = \begin{bmatrix} B & F_{\mathbf{u}} \end{bmatrix}$.
4. Evaluate the Hessian matrix:

$$A_{22} = - \sum_{k=1}^p \lambda_{k,q} F_{k,\mathbf{u}\mathbf{u}}, \quad F_{k,\mathbf{u}\mathbf{u}} = \nabla_{\mathbf{u}}^2 f_k, \quad A = \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & A_{22} \end{bmatrix}$$

5. Solve, for \mathbf{p} and $\boldsymbol{\lambda}_{q+1}$, the quadratic programming problem

$$\min_{\mathbf{p}} \frac{1}{2} \mathbf{p}^T A \mathbf{p} + \mathbf{g}^T \mathbf{p} \quad \text{subject to} \quad C\mathbf{p} = -\mathbf{c}.$$

6. Update $\begin{bmatrix} \boldsymbol{\alpha}_{q+1} \\ \mathbf{u}_{q+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\alpha}_q \\ \mathbf{u}_q \end{bmatrix} + t\mathbf{p}$ for a suitable step length t . (Near the solution we expect t to be close to 1.)

Given an initial estimate of the footpoint parameters \mathbf{u} , estimates of $\boldsymbol{\alpha}$ and $\boldsymbol{\lambda}$ can be estimated as follows. The Lagrangian function for the generalised footpoint problem is

$$\mathcal{L}(\boldsymbol{\alpha}, \mathbf{u}, \boldsymbol{\lambda}) = \frac{1}{2} \boldsymbol{\alpha}^T \boldsymbol{\alpha} - (\mathbf{x} - \mathbf{f} - B\boldsymbol{\alpha})^T \boldsymbol{\lambda},$$

and the solution footpoint parameters necessarily are associated with a critical point $\nabla \mathcal{L} = \mathbf{0}$, i.e., solve the equations

$$\begin{bmatrix} \boldsymbol{\alpha} - B^T \boldsymbol{\lambda} \\ F_{\mathbf{u}}^T \boldsymbol{\lambda} \\ \mathbf{f} + B\boldsymbol{\alpha} - \mathbf{x} \end{bmatrix} = \mathbf{0},$$

where $F_{\mathbf{u}}$ is the $p \times (p-1)$ matrix of partial derivatives of \mathbf{f} with respect to \mathbf{u} . For \mathbf{u} fixed, these equations are linear in $\boldsymbol{\alpha}$ and $\boldsymbol{\lambda}$ and their estimates can be determined by solving a linear least squares problem.

For functionally defined surfaces $y = f(\mathbf{u}, \mathbf{b})$ the generalised footpoint problem involves constraints $\mathbf{x} = \begin{bmatrix} \mathbf{u} \\ f(\mathbf{u}, \mathbf{b}) \end{bmatrix} + B\boldsymbol{\alpha}$, the first $p-1$ of which are linear. These can be eliminated using generalised constraint elimination, so that the footpoint problem is reduced to minimising a quadratic function of p parameters subject to a single nonlinear constraint.

5.4 Numerical Example: Elliptic Hyperboloid

We give an example of the behaviour of the SQP footpoint algorithm for an elliptic hyperboloid defined parametrically by

$$x = a \cos u \cosh v, \quad y = b \sin u \cosh v, \quad z = c \sinh v.$$

We consider two covariance matrices $U_1 = I$, corresponding to orthogonal distance regression, and the rank 1 matrix

$$U_2 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

For U_1 , the solution footpoint \mathbf{f}^* should satisfy $\mathbf{f}^* - \mathbf{x} = t\mathbf{n}$ for some $t \in \mathbb{R}$, where \mathbf{n} is the normal at \mathbf{f}^* . For U_2 , the footpoint \mathbf{f}^* should satisfy $\mathbf{f}^* - \mathbf{x} = t(1, 1, 0)^T$.

We generated test data points

$$\mathbf{x}_q = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \epsilon_{x,q} \\ \epsilon_{y,q} \\ \epsilon_{z,q} \end{bmatrix}, \quad \epsilon_{x,q}, \epsilon_{y,q}, \epsilon_{z,q} \in N(0, \sigma^2).$$

The footpoint algorithm was then employed to find estimates \mathbf{f}_q of the footpoints starting from $u = v = 0$. Second derivative information for the surfaces

was calculated using finite differences. With 1000 data points generated with $\sigma = 0.2$ and convergence tolerances set at 10^{-15} , the algorithm was able to converge in six or fewer iterations in all cases. The rank of the uncertainty matrix had no significant effect on the rate of convergence. Table 1 indicates typical convergence behaviour in terms of $\|\mathbf{p}\|$ and $\|\mathbf{c}\|$, the norms of the update step and the constraint functions.

Table 1. Typical convergence behaviour for the SQP footpoint algorithm in terms of $\|\mathbf{p}\|$ and $\|\mathbf{c}\|$, the norms of the update step and the constraint functions.

Iteration	$\ \mathbf{p}\ $	$\ \mathbf{c}\ $
1	3.96e-01	5.76e-01
2	4.72e-02	1.07e-01
3	6.39e-05	1.22e-04
4	1.06e-11	2.04e-11
5	2.79e-18	0

5.5 Example Application: Calibration Curves

In this section, we discuss a generalised distance regression problem associated with an instrument calibration in which the uncertainty matrices U_i are naturally rank deficient. We suppose that the instrument’s response y depends approximately linearly (or at least monotonically) on a variable x and that for a sequence of calibrated values $x_i, i = 1, \dots, m$, of x , measurements of the responses y_i are made. Given a model of the form

$$y_i^* = \phi(x_i^*, \mathbf{b}), \quad x_i = x_i^* + \delta_i, \quad y_i = y_i^* + \epsilon_i, \quad \delta_i \in N(0, \rho^2), \quad \epsilon_i \in N(0, \sigma^2),$$

the *response calibration curve* is found by solving the generalised distance regression problem

$$\min_{\mathbf{x}^*, \mathbf{b}} \sum_{i=1}^m \{\alpha_i^2 + \beta_i^2\} \quad \text{subject to} \quad \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} x_i^* \\ \phi(x_i^*, \mathbf{b}) \end{bmatrix} + \begin{bmatrix} \rho & 0 \\ 0 & \sigma \end{bmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix},$$

$i = 1, \dots, m$. If $\rho = 0$, as in the case where the uncertainty in the calibrated values of x is much smaller than those associated with the response measurements, this problem reduces to a standard least squares problem

$$\min_{\mathbf{b}} \sum_{i=1}^m (y_i - \phi(x_i, \mathbf{b}))^2.$$

Given a calibrated value of x , the response curve $\phi(x, \mathbf{b})$ predicts the response of the system. In using the instrument, we are interested in estimating the

value of the stimulus variable x given a measurement of the response y . If the instrument is calibrated in terms of the response curve ϕ , then every time we measure with the instrument, recording an uncalibrated response y , we have to use iterative techniques to solve $\phi(x) = y$ in order to output the calibrated value x . A more attractive proposition is to model the instrument behaviour as $x^* = \psi(y^*, \mathbf{b})$ (so that $\phi = \psi^{-1}$) and the *evaluation calibration curve* is found by solving

$$\min_{\mathbf{y}^*, \mathbf{b}} \sum_{i=1}^m \{\alpha_i^2 + \beta_i^2\} \quad \text{subject to} \quad \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} \psi(y_i^*, \mathbf{b}) \\ y_i^* \end{bmatrix} + \begin{bmatrix} \rho & 0 \\ 0 & \sigma \end{bmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix}.$$

Using a separation of variables approach, the case of $\rho = 0$ (or near zero) introduces no complications (nor numerical stability concerns). The output x can be determined from a direct evaluation of $\psi(y, \mathbf{b})$, given a measured response y .

Regarding the response and evaluation calibration curves as parametric curves,

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} u \\ \phi(u, \mathbf{b}) \end{bmatrix}, \quad \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \psi(u, \mathbf{b}) \\ u \end{bmatrix},$$

respectively, both problems can be solved as generalised distance regression problems using the same software.

6 Surface Fitting for Structured Uncertainty Matrices

We have seen in Section 2 that uncertainty matrices U_X are often full with significant correlation amongst all data elements so that generalised distance regression cannot be applied directly. However, if the uncertainty matrix has the factored structure as in (2), then (6) can be written as

$$\min_{\mathbf{a}, \boldsymbol{\alpha}_0} \sum_{i=0}^m \boldsymbol{\alpha}_i^T \boldsymbol{\alpha}_i \quad \text{subject to} \quad \mathbf{x}_i = \mathbf{f}(\mathbf{u}_i, \mathbf{b}) + B_i \boldsymbol{\alpha}_i + B_{0,i} \boldsymbol{\alpha}_0, \quad i = 1, \dots, m. \quad (20)$$

Holding \mathbf{b} and $\boldsymbol{\alpha}_0$ fixed, it is seen that optimal $\boldsymbol{\alpha}_i$ must solve the footpoint problem (14) but for the surface $\bar{\mathbf{f}}_i(\mathbf{u}, \mathbf{b}, \boldsymbol{\alpha}_0) = \mathbf{f}(\mathbf{u}, \mathbf{b}) + B_{0,i} \boldsymbol{\alpha}_0$. Following the same approach as described in Section 4, we define the generalised distance $d_i(\mathbf{b}, \boldsymbol{\alpha}_0)$ as a function of \mathbf{b} and $\boldsymbol{\alpha}_0$ evaluated at the solution of the i th footpoint. Then (20) is equivalent to

$$\min_{\mathbf{b}, \boldsymbol{\alpha}_0} \left\{ \boldsymbol{\alpha}_0^T \boldsymbol{\alpha}_0 + \sum_{i=1}^m d_i^2(\mathbf{b}, \boldsymbol{\alpha}_0) \right\}, \quad (21)$$

and can be solved using standard nonlinear least squares algorithms. This results in an $\mathcal{O}(m)$ algorithm.

By introducing the parameters $\boldsymbol{\alpha}_0$ explicitly into the optimisation problem to explain the correlation in the point coordinates, a much more efficient algorithm is made possible. The first main element of the approach is to exploit the structure in the uncertainty matrix by using the factorisation (2) which arises naturally in the problem formulation, rather than the Cholesky factorisation in which all the structure is irretrievably lost. The second main element is to pose the problem as a constrained optimisation problem (6) rather than the unconstrained problem (5).

6.1 Parametric Surface Fitting in \mathbb{R}^3

To illustrate the separation of variables approach for structured uncertainty matrices, we show how it can be applied in the case of fitting a parametric surface in \mathbb{R}^3 to data gathered by a laser tracker system, for example.

We assume we are given an $m \times 3$ matrix X of data points \mathbf{x}_i , and that the associated uncertainty matrix is specified in terms of a $3m \times k$ matrix B with $B(3i - 2 : 3i, :) = B_i$ and a $3m \times k_0$ matrix B_0 with $B_0(3i - 2 : 3i, :) = B_{0,i}$. We also assume that an $m \times 2$ matrix U of starting estimates $\mathbf{u}_i = (u_i, v_i)^T$ for the footpoint point parameters are provided (or can be estimated from X).

The following steps calculate $m + k_0$ function values $\mathbf{e}^T(\mathbf{a}) = (\mathbf{d}^T, \boldsymbol{\alpha}_0^T)$ and $(m + k_0) \times (n + k_0)$ Jacobian matrix J associated with (21).

- A For $i = 1, \dots, m$,
 - I Extract \mathbf{x}_i , \mathbf{u}_i , B_i and $B_{0,i}$ from X , U , B and B_0 , respectively, and set $\tilde{\mathbf{x}}_i = \mathbf{x}_i - B_{0,i}\boldsymbol{\alpha}_0$.
 - II Solve the footpoint problem for $\tilde{\mathbf{x}}_i$, B_i and $\mathbf{f}(\mathbf{u}, \mathbf{b})$, with starting estimate \mathbf{u}_i . Store updated estimate \mathbf{u}_i in U .
 - III Calculate $\mathbf{f}_i(\mathbf{u}_i, \mathbf{b})$, vectors \mathbf{f}_u , \mathbf{f}_v , the partial derivatives of \mathbf{f} with respect to u , v , and $3 \times n$ matrix $F_{\mathbf{b}}$ of partial derivatives of \mathbf{f} with respect to b_j , $j = 1, \dots, n$.
 - IV Calculate normal vector $\mathbf{n}_i = \mathbf{f}_u \times \mathbf{f}_v$ (vector cross-product) and weight $w_i = \|B_i^T \mathbf{n}_i\|$.
 - V Set $e_i = \mathbf{n}_i^T(\tilde{\mathbf{x}}_i - \mathbf{f}_i)/w_i$ and $J(i, :) = -\mathbf{n}_i^T [F_{\mathbf{b}} B_{0,i}]/w_i$.
- B Augment \mathbf{e} and J : $\mathbf{e}(m + 1 : m + k_0) = \mathbf{0}$, $J(m + 1 : m + k_0, :) = [\mathbf{0} \ I]$.

This algorithm represents only a minor modification over that required for generalised distance regression with a parametric surface in \mathbb{R}^3 .

7 Concluding Remarks

This paper has been concerned with fitting model surfaces $\mathbf{f}(\mathbf{u}, \mathbf{b})$ to measurement data $X = \{\mathbf{x}_i\}$, taking into account uncertainty in the measurement data as summarised by an uncertainty matrix U_X . For the case where the

measurements \mathbf{x}_i and \mathbf{x}_q are statistically independent, $i \neq q$, the uncertainty matrix U_X is block-diagonal and a separation of variables approach is possible. In Section 4 we showed that this approach applies equally well in the case where the uncertainty matrix is rank deficient and in Section 5 we described a compact, sequential quadratic programming algorithm that gives accurate estimates of the footpoint parameters, a key computation in the separation of variables approach. By posing the regression problem as a constrained optimisation problem, we showed that the separation of variables approach can be extended to full uncertainty matrices U_X provided they arise in a factored form that corresponds to a dependence of the measurements $\mathbf{x}_i = \mathbf{x}(\boldsymbol{\epsilon}_i, \boldsymbol{\delta})$ on common factors $\boldsymbol{\delta}$. In Section 2, we saw that this form of structured uncertainty matrix appeared often in practice. Thus, using the techniques described here, the separation of variables approach (usually applied to orthogonal regression problems) can be extended to the case of rank deficient uncertainty matrices and also to a wide class of full uncertainty matrices. This enables the regression problem to be solved in $\mathcal{O}(m)$ steps rather than $\mathcal{O}(m^3)$, where m is the number of data points. Applications include fitting response surfaces to data and fitting geometric surfaces to coordinate data.

Acknowledgement

This work was supported by the Department of Trade and Industry's Software Support for Metrology Programme. The paper has benefited from peer review comments and those from Dr Ian Smith, NPL. The author gratefully acknowledges the kind support provided by the Algorithms for Approximation V International Committee.

References

1. I.J. Anderson, M.G. Cox, A.B. Forbes, J.C. Mason, and D.A. Turner: An efficient and robust algorithm for solving the footpoint problem. In *Mathematical Methods for Curves and Surfaces II*, M. Dæhlen, T. Lyche, and L.L. Schumaker (eds.), Vanderbilt University Press, Nashville, 1998, 9–16.
2. R.M. Barker, M.G. Cox, A.B. Forbes, and P.M. Harris: *Software Support for Metrology Best Practice Guide 4: Modelling Discrete Data and Experimental Data Analysis*. Technical report, National Physical Lab, Teddington, 2004.
3. G. Belforte, B. Bona, E. Canuto, F. Donati, F. Ferraris, I. Gorini, S. Morei, M. Peisino, and S. Sartori: Coordinate measuring machines and machine tools selfcalibration and error correction. *Ann. CIRP* **36**(1), 1987, 359–364.
4. P.T. Boggs, R.H. Byrd, and R.B. Schnabel: A stable and efficient algorithm for nonlinear orthogonal distance regression. *SIAM J. Sci. Stat. Comp.* **8**(6), 1987, 1052–1078.
5. B.P. Butler, A.B. Forbes, and P.M. Harris: Geometric tolerance assessment problems. In: *Advanced Mathematical Tools in Metrology*, P. Ciarlini, M.G. Cox, R. Monaco, and F. Pavese (eds.), World Scientific, Singapore, 1994, 95–104.

6. M.G. Cox: The least squares solution of overdetermined linear equations having band or augmented band structure. *IMA J. Numer. Anal.* **1**, 1981, 3–22.
7. M.G. Cox: Linear algebra support modules for approximation and other software. In: *Scientific Software Systems*, J.C. Mason and M.G. Cox (eds.), Chapman & Hall, London, 1990, 21–29.
8. M.G. Cox, A.B. Forbes, P.M. Fossati, P.M. Harris, and I.M. Smith: *Techniques for the Efficient Solution of Large Scale Calibration Problems*. Technical report, National Physical Laboratory, Teddington, May 2003.
9. M.G. Cox, A.B. Forbes, P.M. Harris, and G.N. Peggs: Experimental design in determining the parametric errors of CMMs. In: *Laser Metrology and Machine Performance IV*, V. Chiles and D. Jenkinson (eds.), WIT Press, Southampton, 1999, 13–22.
10. M.G. Cox and P.M. Harris: *Software Support for Metrology Best Practice Guide No. 6: Uncertainty Evaluation*. Technical report, National Physical Laboratory, Teddington, 2004.
11. A.B. Forbes: Generalised regression problems in metrology. *Numerical Algorithms* **5**, 1993, 523–533.
12. A.B. Forbes: Efficient algorithms for structured self-calibration problems. In: *Algorithms for Approximation IV*, J. Levesley, I. Anderson, and J.C. Mason (eds.), University of Huddersfield, 2002, 146–153.
13. A.B. Forbes: Surface fitting taking into account uncertainty structure in coordinate data. *Measurement Science and Technology* **17**, 2006, 553–558.
14. A.B. Forbes, P.M. Harris, and I.M. Smith: Generalised Gauss-Markov regression. In: *Algorithms for Approximation IV*, J. Levesley, I. Anderson, and J.C. Mason (eds.), University of Huddersfield, 2002, 270–277.
15. P.E. Gill, W. Murray, and M.H. Wright: *Practical Optimization*. Academic Press, London, 1981.
16. G.H. Golub and C.F. Van Loan: *Matrix Computations*. 3rd edition, John Hopkins University Press, Baltimore, 1996.
17. S. Hammarling: *The Numerical Solution of the General Gauss-Markov Linear Model*. Technical Report TR2/85, Numerical Algorithms Group, Oxford, 1985.
18. H.-P. Helfrich and D. Zwick: A trust region method for implicit orthogonal distance regression. *Numerical Algorithms* **5**, 1993, 535–544.
19. H.-P. Helfrich and D. Zwick: A trust region method for parametric curve and surface fitting. *J. Comput. Appl. Math.* **73**, 1996, 119–134.
20. J.P. Kruth, P. Vanherk, and L. de Jonge: Self-calibration method and software error correction for three dimensional co-ordinate measuring machines using artefact measurements. *Measurement*, 1994, 1–11.
21. K.V. Mardia, J.T. Kent, and J.M. Bibby: *Multivariate Analysis*. Academic Press, London, 1979.
22. C.C. Paige: Fast numerically stable computations for generalized least squares problems. *SIAM J. Numer. Anal.* **16**, 1979, 165–171.
23. SIAM, Philadelphia: *The LAPACK User's Guide*. 3rd edition, 1999.
24. D. Sourlier and W. Gander: A new method and software tool for the exact solution of complex dimensional measurement problems. In: *Advanced Mathematical Tools in Metrology, II*, P. Ciarlini, M.G. Cox, F. Pavese, and D. Richter (eds.), World Scientific, Singapore, 1996, 224–237.
25. G. Zhang, R. Ouyang, B. Lu, R. Hocken, R. Veale, and A. Donmez: A displacement method for machine geometry calibration. *Annals of the CIRP* **37**, 1988, 515–518.

Uncertainty Evaluation in Reservoir Forecasting by Bayes Linear Methodology

Daniel Busby¹, Chris L. Farmer^{1,2}, and Armin Iske³

¹ Schlumberger Abingdon Technology Center, Abingdon OX14 1UJ, UK,
{`dbusby4`, `farmer5`}@`s1b.com`

² Oxford Centre for Industrial and Applied Mathematics, University of Oxford,
Oxford OX1 3LB, UK, `farmer5@s1b.com`

³ Department of Mathematics, University of Hamburg, D-20146 Hamburg,
Germany, `iske@math.uni-hamburg.de`

Summary. We propose application of Bayes linear methodology to uncertainty evaluation in reservoir forecasting. On the basis of this statistical model, effective emulators are constructed. The resulting statistical method is illustrated by application to a commonly used test case scenario, called PUNQS [11]. A statistical data analysis of different output responses is performed. Responses obtained from our emulator are compared with both true responses and with responses obtained using the response surface methodology (RSM), the basic method used by leading commercial software packages.

1 Introduction

A reservoir simulator is a large computer code which requires solving a system of nonlinear partial differential equations from complex geological model data. The reservoir geology is typically characterized by a huge number of input parameters to the simulator. As these input parameters are usually uncertain, so is the output of the simulator uncertain. Thus, uncertainty evaluation of large simulation codes has become a major task in reservoir forecasting.

In this paper Bayes linear methodology is applied to reservoir forecasting using a sequential experimental design [9] for the construction of effective emulators. We remark that the application of the Bayes linear approach to comparable applications was recently discussed in related works [3, 7]. Moreover, our sequential experimental design is similar to that one in [13].

The performance of Bayes linear methodology is evaluated by comparison with true responses for different outputs of the reservoir simulator. Moreover, response surfaces from reservoir forecasting are analyzed, and our results are also compared with the *response surface methodology* (RSM) [6], which is the basic method of the commercial software package COUGAR [2].

The outline of this paper is as follows. In Section 2, the methodology of Bayes linear estimation is reviewed. In Section 3, a model for the construction of effective emulators, based on the Bayes linear estimator, is proposed. Numerical results are in Section 4, where numerical comparisons with the response surface methodology (RSM) are performed.

2 Bayes Linear Methodology

Simulator output $s(\mathbf{x})$ is a function of n , $n \geq 1$, uncertain input parameters $\mathbf{x} \in \chi \subset \mathbb{R}^n$. Uncertainty evaluation requires the *probability density*

$$p(y) = p(s(\mathbf{x}) = y) = \int_{\chi} \delta(s(\mathbf{x}) - y) \rho(\mathbf{x}) d\mathbf{x},$$

where $\rho(\mathbf{x})$ is a given density function of $\mathbf{x} \in \chi$ and δ is the Dirac δ -functional. Statistical quantities, such as expectation, $E[s(\mathbf{x})]$, or variance, $\text{Var}[s(\mathbf{x})]$, are also of particular interest,

$$E[s(\mathbf{x})] = \int_{\chi} s(\mathbf{x}) \rho(\mathbf{x}) d\mathbf{x},$$

$$\text{Var}[s(\mathbf{x})] = \int_{\chi} |s(\mathbf{x}) - E[s(\mathbf{x})]|^2 \rho(\mathbf{x}) d\mathbf{x}.$$

For these tasks, Monte Carlo methods are computationally too expensive, as too many simulation runs are required. As shown in [3, 6, 8], more sophisticated statistical approaches, such as response surface methodology (RSM) or Bayesian approaches, are more appropriate than Monte Carlo methods.

When $s(\mathbf{x})$ is a smooth function, one can use multiple regression techniques to approximate $s(\mathbf{x})$ from a few simulation runs. In the RSM, a linear model is used, i.e., a linear combination of q fixed basis functions; usually low order polynomials. The coefficients of the linear model are calculated using a standard least squares technique.

RSM was originally introduced in physical experiments, where each observation of a physical process is subject to measurement error. In contrast, a simulator is deterministic, i.e., rerunning the code with the same inputs gives identical observations. In this case, an interpolatory estimator rather than an approximation is usually preferred. A Bayesian approach yields, unlike RSM, an *interpolatory* (posterior) estimator, see the appendix of [3] for details.

Application of a Bayesian approach results in updating a prior distribution of a statistical model s_B by *Bayes' rule*,

$$P_{\text{Post}}(s_B(\mathbf{x})|s_{\mathbf{X}}) \propto P_{\text{Prior}}(s_B(\mathbf{x})) P_{\text{Likelihood}}(s_{\mathbf{X}}|s_B(\mathbf{x})),$$

where $s_{\mathbf{X}} = (s(\mathbf{x}_1), \dots, s(\mathbf{x}_m))^T \in \mathbb{R}^m$ denotes a response vector containing m simulation outputs taken at a design set $X = \{\mathbf{x}_1, \dots, \mathbf{x}_m\} \subset \mathbb{R}^n$ of m pairwise distinct input configurations, and P is the (conditional) probability.

We prefer to work with a Bayes *linear* estimator, as suggested in [3]. This is mainly for computational reasons, as the Bayes linear estimator s_{BL} considers *only* the first two moments of the prior and posterior distribution, which are related by

$$\begin{aligned} \mathbb{E}[s_{\text{BL}}(\mathbf{x})|s_{\mathbf{X}}] &= \mathbb{E}[s_{\text{BL}}(\mathbf{x})] + \text{Cov}[s_{\text{BL}}(\mathbf{x}), s_{\mathbf{X}}]\text{Var}[s_{\mathbf{X}}]^{-1}(s_{\mathbf{X}} - \mathbb{E}[s_{\mathbf{X}}]), \\ \text{Var}[s_{\text{BL}}(\mathbf{x})|s_{\mathbf{X}}] &= \text{Var}[s_{\text{BL}}(\mathbf{x})] + \text{Cov}[s_{\text{BL}}(\mathbf{x}), s_{\mathbf{X}}]\text{Var}[s_{\mathbf{X}}]^{-1}\text{Cov}[s_{\mathbf{X}}, s_{\text{BL}}(\mathbf{x})]. \end{aligned}$$

Therefore, Bayes linear estimation can be viewed as an approximation to a full Bayesian approach. Moreover, we remark that in the absence of any prior information on model parameters for mean and autocovariance, the Bayes linear methodology is equivalent to (universal) kriging, see [5] for details.

Now the random process $s_{\text{BL}}(\mathbf{x})$ with posterior mean $\mathbb{E}[s_{\text{BL}}(\mathbf{x})|s_{\mathbf{X}}]$ and variance $\text{Var}[s_{\text{BL}}(\mathbf{x})|s_{\mathbf{X}}]$ is referred to as an *emulator*. An emulator is a cheap surrogate for a (costly) simulator.

3 Construction of the Emulator

3.1 Model Description

Similarly to [3], we work with a (prior) emulator of the form

$$s_{\text{BL}}(\mathbf{x}) = \beta^T g(\mathbf{x}_*) + \epsilon(\mathbf{x}_*), \quad (1)$$

with unknown coefficients $\beta \in \mathbb{R}^q$, $q < m$, regression functions $g = (g_1, \dots, g_q)$, and where \mathbf{x}_* are the active variables of $\mathbf{x} \in \chi$. Loosely speaking, the active variables are those which account for most of the output variation. The discrepancy between the linear regression $\beta^T g(\mathbf{x}_*)$ and the simulator $s(\mathbf{x})$ is modelled by a stationary Gaussian process $\epsilon(\mathbf{x}_*)$ with zero mean and an autocovariance function

$$\text{Cov}[\epsilon(\mathbf{x}_*), \epsilon(\mathbf{y}_*)] = \sigma_\epsilon^2 r(\mathbf{x}_* - \mathbf{y}_*),$$

where $r(\mathbf{z})$ denotes a correlation function to be specified. The selection of active variables \mathbf{x}_* , of the regression functions g and of the correlation function $r(\mathbf{z})$ are based on prior knowledge about the process. This is discussed in the following subsection.

3.2 The Prior Summaries

Prior knowledge about the random process is usually built by expert elicitation [4]. In our case, an initial set of simulator runs is used to support the elicitation process. This initial data is not analyzed statistically. The data is rather interpreted by reservoir engineers who provide estimates of the prior

mean $E[s_{\text{BL}}(\mathbf{x})]$ and variance $\text{Var}[s_{\text{BL}}(\mathbf{x})]$. The required selection of the active variables \mathbf{x}_* and of the regression functions g in (1), usually low order polynomials, is done through sensitivity analysis, as described in [10, 14].

We decided to work with the autocovariance function

$$\text{Cov}[\epsilon(\mathbf{x}_*), \epsilon(\mathbf{y}_*)] = \sigma_\epsilon^2 \exp(-\theta \|\mathbf{x}_* - \mathbf{y}_*\|), \quad (2)$$

which leads to continuous but non-smooth response surfaces, as desired in the situation of our particular application, see Section 4. In a more general situation, the selection of the autocovariance function in (2) should be made on the basis of previous observations in similar problems.

The parameters θ and σ_ϵ in (2) can be determined by *maximum likelihood estimation* (MLE), see [13]. This gives

$$\hat{\sigma}_\epsilon^2 = \frac{1}{m} (s_{\mathbf{X}} - G\beta)^T R^{-1} (s_{\mathbf{X}} - G\beta),$$

for the estimation of σ_ϵ^2 , where m is the number of simulations, and where

$$R = (r(\mathbf{x}_i, \mathbf{x}_j))_{1 \leq i, j \leq m} \in \mathbb{R}^{m \times m}, \quad G = (g_j(\mathbf{x}_i))_{1 \leq i \leq m; 1 \leq j \leq q} \in \mathbb{R}^{m \times q}.$$

Estimation of θ by MLE requires global optimization and is generally sensitive to the number of simulations. Therefore, in our case we prefer to use data visualization techniques which yields a more robust estimate of $\hat{\theta} = 2$ for θ . For more details on the estimation of the autocovariance function in (2) we refer to our previous paper [1].

3.3 Experimental Design

In computer simulations, the goal of experimental design is to determine suitable input configurations for effective data analysis. The required data analysis is specific to the objectives of the experiment. Possible objectives include uncertainty propagation, optimization of certain response functionals (e.g. oil production), and tuning the simulator to physical data, *history matching*.

In reservoir forecasting, experimental design is of primary importance, especially since each simulation run is computationally very expensive. In view of uncertainty evaluation, we are aiming at the construction of a sufficiently accurate emulator to predict responses at untried input. But we wish to keep the number of required simulation runs as small as possible.

Possible experimental designs can be split in two different categories: single stage methods, such as fractional factorial designs (FFD) or Latin hypercube designs (LHC), and sequential designs which aim at minimizing uncertainty measures of the emulator. In the approach proposed in this paper, a number of initial simulator runs are first performed by using FFD. Then, a number of subsequent simulator runs are done by using a sequential design. But this requires a specific design criterion.

The design criterion we work with relies on the *maximum mean square error* (MMSE). In this case, design points, \mathbf{x}^* , are sequentially added, one at a time, where the posterior variance $\text{Var}[s_{\text{BL}}(\mathbf{x})|s_{\mathbf{X}}]$ of the current Bayes linear emulator $s_{\text{BL}} \equiv s_{\text{BL}}^{(m)}$ is maximal among all $\mathbf{x} \in \chi$. In this way, the prediction error of the subsequent (posterior) emulator $s_{\text{BL}}^{(m+1)}$ vanishes at \mathbf{x}^* . A similar design criterion is proposed in [13], but for kriging.

In summary, each step of the sequential design is performed as follows.

- (1) Compute an input configuration \mathbf{x}^* which maximizes $\text{Var}[s_{\text{BL}}(\mathbf{x})|s_{\mathbf{X}}]$;
- (2) Run the simulator at the selected configuration \mathbf{x}^* to obtain $s(\mathbf{x}^*)$;
- (3) Rebuild the emulator by including the new simulator output $s(\mathbf{x}^*)$.

As regards a stopping criterion, we chose a customized diagnostic measure which relies on the prediction error

$$\eta(m) = |s_{\text{BL}}^{(m-1)}(\mathbf{x}_m) - s(\mathbf{x}_m)|,$$

where $\mathbf{x}_m = \mathbf{x}^*$ denotes the design point which was added at step m , and $s(\mathbf{x}_m)$ is the simulator response at \mathbf{x}_m . Note that $s_{\text{BL}}^{(m)}(\mathbf{x}_m) = s(\mathbf{x}_m)$. When the sequence $\eta(m)$ of prediction errors *stabilizes*, i.e., $|\eta(m) - \eta(m-1)| < \text{TOL}$ for some tolerance TOL, we take $s_{\text{BL}}^{(m)}$ as an a sufficiently accurate emulator.

4 Numerical Results for the PUNQS Test Case

4.1 Reservoir Model Description

The PUNQS test case relies on a synthetic reservoir model taken from the North Sea Brent reservoir, a real-world oilfield. The PUNQS test case is frequently used as an industrial reservoir engineering model since its use in the European research project PUNQ [11] as a benchmark test for comparative inversion studies and for stochastic reservoir modelling.

A top structure map of the PUNQS reservoir field is shown in Figure 1. The geological model contains $19 \times 28 \times 5 = 2660$ grid blocks, 1761 of which are active. The reservoir is surrounded by a strong aquifer in the North and in the West, and it is bounded by a fault to the East and to the South. A small gas cap is located in the centre of this dome-shaped structure. The geological model consists of five independent layers, where the porosity distribution in each layer was modelled by geostatistical simulation. Initially, the field contains six production wells located around the gas-oil contact. Due to the strong aquifer, no injection wells are required.

As suggested by reservoir engineers, we consider the following seven main sources of uncertainty: (i) the analytical coefficient of the aquifer strength, **AQU**, (ii) the residual gas oil saturation, **GOS**, (iii) the residual water oil saturation, **WOS**, (iv) the vertical permeability multiplier in low quality sands, **VPML**, (v) the vertical permeability multiplier in high quality

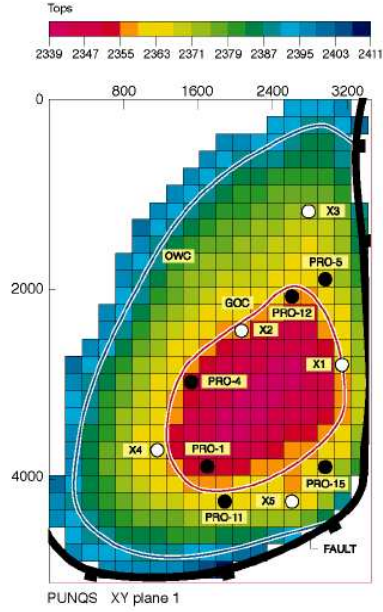


Fig. 1. PUNQS test case. Top structure map of the reservoir field.

sands, **VPMH**, (vi) the horizontal permeability multiplier in low quality sands, **HPML**, (vii) the horizontal permeability multiplier in high quality sands, **HPMH**. For each of the seven input variables, a uniform distribution in the parameter interval $[-1, 1]$ is assumed.

To evaluate and compare different methods by their emulator accuracy, we decided to work with three different error measures when recording the resulting prediction errors for an emulator s_E . The error measures are the *mean absolute error*

$$\eta_1 = \|s - s_E\|_1 / |\Xi| = \frac{1}{|\Xi|} \sum_{\mathbf{x} \in \Xi} |s(\mathbf{x}) - s_E(\mathbf{x})|,$$

mean square error,

$$\eta_2^2 = \|s - s_E\|_2^2 / |\Xi| = \frac{1}{|\Xi|} \sum_{\mathbf{x} \in \Xi} |s(\mathbf{x}) - s_E(\mathbf{x})|^2,$$

and *maximum error*,

$$\eta_\infty = \|s - s_E\|_\infty = \max_{\mathbf{x} \in \Xi} |s(\mathbf{x}) - s_E(\mathbf{x})|,$$

where Ξ denotes a fine uniform grid contained in the computational domain χ . We have implemented the proposed approach in the language R [12].

4.2 Numerical Results from Two-Dimensional Input

In this subsection, we present numerical results for two different responses in the PUNQS model from 2D input. The small size of the PUNQS reservoir model, containing only less than 20,000 grid cells, allows us to perform several thousand simulation runs, which are included in the two numerical tests. The responses from these simulations are taken to visualize the *real* response surface, whose graph is then compared with both the graph of the Bayes linear emulator s_{BL} and the graph of the emulator s_{RSM} obtained by the response surface methodology (RSM).

To demonstrate the good performance of the proposed Bayes linear approach, we selected two rather challenging test cases involving rough response surfaces $s(\mathbf{x})$ of high variation.

The first test case is concerning the oil production rate at well **PRO15** (see Figure 1 bottom right) after 13 years, response surface **P15OPR**, as a function of its two main active variables, **HPMH** and **HPML**. The design set X was constructed by applying FFD to obtain an initial set of 7 points, followed by a sequential design for further 5 points, yielding $m = 12$ design points in total.

Figure 2 displays the response surface of the Bayes linear emulator, s_{BL} , and the response surface obtained by RSM, emulator s_{RSM} . For comparison, Figure 2 displays 10×10 grid points of the true response surface.

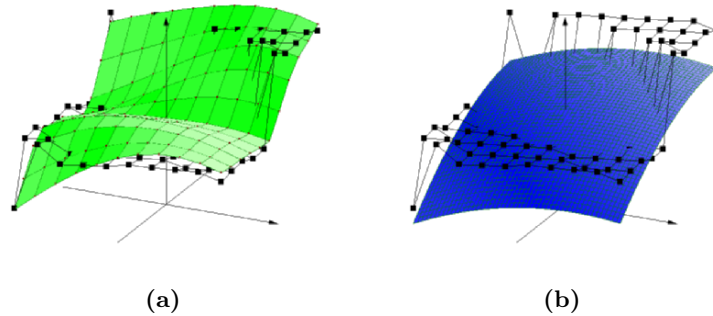


Fig. 2. PUNQS test case **P15OPR(HPMH,HPML)**. Response surface of (a) Bayes linear emulator s_{BL} , (b) s_{RSM} , each constructed by using 12 design points. A 10×10 mesh grid of the true response surface **P15OPR** is shown for comparison.

Note that the response surface s_{BL} obtained from the Bayes linear estimator (Figure 2 (a)) is, in comparison with s_{RSM} of RSM (Figure 2 (b)), much closer to the true response surface **P15OPR**, and so the Bayes linear esti-

mator is superior. This is also confirmed by our numerical results in Table 1, where their prediction errors η_1 , η_2 , and η_∞ are shown.

Table 1. PUNQS test case **P15OPR(HPMH,HPML)**. Prediction errors from emulators s_{BL} and s_{RSM} , each constructed by using $m = 12$ design points.

Method	η_1	η_2	η_∞
BL	3.0	4.6	17.6
RSM	6.2	7.1	16.2

In our second test case, we consider the bottom hole pressure at well **PRO15** after 13 years, response surface **P15BHP**, as a function of **HPMH** and **GOS**. The design set X was constructed by applying FFD to obtain an initial set of 7 points, followed by a sequential design for further 2 points, yielding $m = 9$ design points in total.

Figure 3 displays the response surface of the Bayes linear emulator, s_{BL} , and the response surface obtained by RSM, emulator s_{RSM} , each of which was constructed by using $m = 9$ design points. For comparison, Figure 3 displays 9×9 grid points of the true response surface. Our numerical results are shown in Table 2.

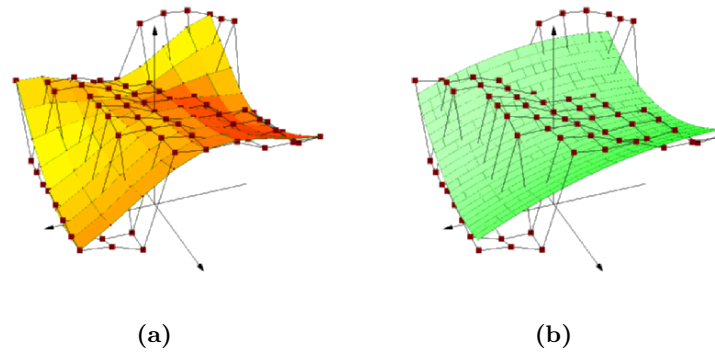


Fig. 3. PUNQS test case **P15BHP(HPMH,GOS)**. Response surface of (a) Bayes linear emulator s_{BL} , (b) s_{RSM} , each constructed by using $m = 9$ design points. A 9×9 mesh grid of the true response surface **P15OPR** is shown for comparison.

Note that the Bayes linear estimator continues to be superior to RSM in terms of its better reconstruction quality. This is supported by both the response surface graphs in Figure 3 and the numerical results in Table 2. Table 2 shows the prediction errors η_1 , η_2 and η_∞ obtained from the two

Table 2. PUNQS test case **P15BHP(HPMH,GOS)**. Prediction errors from emulators s_{BL} and s_{RSM} , constructed by using $m = 7$ and $m = 9$ design points each.

Method	m	η_1	η_2	η_∞	m	η_1	η_2	η_∞
BL	7	3.7	5.3	13.4	9	2.7	4.3	12.6
RSM	7	4.2	5.5	12.4	9	3.6	4.8	11.3

different emulators, s_{BL} and s_{RSM} . Note that Table 2 involves two different comparisons, one using the initial set of $m = 7$ design points, the other using all $m = 9$ design points. Note that the accuracy of the emulator s_{BL} is, unlike that of s_{RSM} , significantly improved by the adaptive insertion of only two design points, \mathbf{x}_8 and \mathbf{x}_9 . Moreover, the prediction quality of the Bayes linear emulator s_{BL} is superior to that of s_{RSM} not only in smooth regions of the true surface **P15BHP**, but also in regions where **P15BHP** is highly nonlinear. However, the emulator s_{BL} exhibits small overshoots near discontinuities of **P15BHP**, which explains the somewhat inferior prediction error η_∞ of s_{BL} . The same comment applies to our first test case, see Table 1.

4.3 Numerical Results from High-Dimensional Input

Let us finally present numerical results obtained from high-dimensional input configurations. To this end, we have analyzed responses from output concerning the oil production rate at production well **PRO15** after 13 years, response **P15OPR**, as a function of all seven input variables which were listed at the outset of this section, **AQU**, **GOS**, **WOS**, **VPML**, **VPMH**, **HPML**, and **HPMH**.

We have performed an initial fractional factorial design (FFD) of 79 simulations. To reduce computational complexity, a sequential design is performed in the restricted input space of the three dominating active variables, **HPMH**, **HPML**, and **WOS**. These three main active variables were determined by a sensitivity analysis (using a Pareto plot [9]), on the basis of the 79 initial simulator runs. Further 30 design points were added by sequential design, yielding $m = 109$ design points in total.

Given the high dimension of this test case, $n = 7$, in combination with the small number of design points, $m = 109$, Bayes linear estimation performs remarkably well in terms of prediction quality obtained from its emulator s_{BL} . Indeed, we found $\eta_1 = 4.3$, $\eta_2 = 5.0$, and $\eta_\infty = 13.1$.

5 Conclusion

We have shown the utility of Bayes linear methodology, in combination with sequential adaptive design, for uncertainty evaluation in reservoir forecasting. The resulting Bayes linear estimation has been applied to the PUNQS

test case, a rather simple but fairly realistic and frequently used model problem from reservoir engineering. The performance of the resulting emulator has been compared with that obtained from the response surface methodology (RSM), the basic method of commercial reservoir software, such as COUGAR [2]. We found that the Bayes linear methodology is superior to RSM, especially for highly nonlinear responses. For high-dimensional input data a significant number of more simulator runs need to be included in the initial sequential design. This is illustrated in our previous paper [1].

Acknowledgement

The work of Daniel Busby and Armin Iske was supported by Schlumberger and by the European Union through the project FAMOUS, contract no. ENK6-CT-2002-50528. Chris L. Farmer thanks the Royal Society for support through an Industry Fellowship at the University of Oxford.

References

1. D. Busby, C.L. Farmer, and A. Iske: Hierarchical nonlinear approximation for experimental design and statistical data fitting. Manuscript, 2005.
2. The COUGAR project, <http://consortium.ifp.fr/cougar/>.
3. P.S. Craig, M. Goldstein, J.C. Rougier, and A.H. Seheult: Bayesian forecasting for complex systems using computer simulators. *Journal of the American Statistical Association* **96**, 2001, 717–729.
4. P.S. Craig, M. Goldstein, J.A. Smith, and A.H. Seheult: Constructing partial prior specifications for models of complex physical systems. *The Statistician* **47**, 1998, 37–53.
5. C. Currin, T. Mitchell, M. Morris, and D. Ylvisaker: A Bayesian approach to the design and analysis of computer experiments. ORNL Technical Report 6498, National Technical Information Service, Springfield, VA 22161, 1988.
6. J.P. Dejean and G. Blanc: Managing uncertainties on production predictions using integrated statistical methods. *SPE Journal*, SPE 56696, 1999.
7. M. Goldstein: Bayes linear analysis. *Encyclopaedia of Statistical Sciences*, 1998.
8. M.C. Kennedy and A. O’Hagan: Bayesian calibration of computer models. *Journal of the Royal Statistical Society B*. **63**, 2000, 425–464.
9. R.H. Myers and D.C. Montgomery: *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. Wiley, 2002.
10. J. Oakley, A. O’Hagan: Probabilistic sensitivity analysis of complex models: a Bayesian approach. *Journal of the Royal Statistical Society B* **63**, 2004, 425–464.
11. The European project PUNQ (Production Forecasting with UNcertainty Quantification), <http://www.nitg.tno.nl/punq/>.
12. R Development Core Team: *R Foundation for Statistical Computing*. <http://www.R-project.org>.
13. J. Sacks, W.J. Welch, T.J. Mitchell, and H.P. Wynn: Design and analysis of computer experiments. *Statistical Science* **4**(4), 1989, 409–435.
14. A. Saltelli, K. Chan, and M. Scott: *Sensitivity Analysis*, Wiley, New York, 2000.

Part IV

Data Fitting and Modelling

Integral Interpolation

Rick K. Beatson and Michael K. Langton

Department of Mathematics and Statistics, University of Canterbury, Christchurch
8020, New Zealand, R.Beatson@math.canterbury.ac.nz

Summary. This paper concerns interpolation problems in which the functionals are integrals against signed measures rather than point evaluations. Sufficient conditions for related strict positive definiteness properties to hold, and formulas making such integral interpolation problems computationally practical, are considered.

1 Introduction

This paper concerns interpolation problems in which the data to be interpolated consists of approximate averages of an unknown function over compact sets such as points, balls and line segments in \mathbb{R}^n . Such an *integral interpolation* approach is natural for many datasets, for example for track data arising in geophysics. We will discuss both the underlying mathematical theory and explicit formulas making the techniques practical for large problems.

Let π_{k-1}^n denote the space of polynomials of degree at most $k-1$ in n variables. In this paper various integral sources will be derived from parent basic functions Φ which are strictly integrally conditionally positive definite in the sense defined below. This definition echoes that of Cheney and Light [5, p. 133].

Definition 1. A continuous real valued kernel $\Phi(\cdot, \cdot)$ will be called *integrally conditionally positive definite of order k on \mathbb{R}^n* if

(i) $\Phi(x, y) = \Phi(y, x)$ for all x, y in \mathbb{R}^n .

(ii) $E(\mu, \mu) = \iint \Phi(x, y) d\mu(x) d\mu(y) \geq 0$

for every compactly supported regular Borel (signed) measure μ on \mathbb{R}^n , such that

$$\int_{\mathbb{R}^n} q(x) d\mu(x) = 0 \quad \text{for all } q \in \pi_{k-1}^n.$$

The kernel Φ will be called *integrally strictly conditionally positive definite of order k on \mathbb{R}^n* , denoted $\text{ISPD}_k(\mathbb{R}^n)$, if the inequality is strict whenever μ is nonzero.

Several examples of $\text{ISPD}_k(\mathbb{R}^n)$ basic functions are listed in Sections 4 and 5 below.

The definition above is a generalisation of the well known definition of pointwise strict conditional positive definiteness which arises when ordinary pointwise, or Lagrange, interpolation is considered. The ordinary pointwise definition will be recovered if we restrict μ to be a finite weighted sum of point evaluations. That is, if we require

$$\mu = \sum_{j=1}^m c_j \delta_{x_j},$$

so that

$$\mu(q) = \sum_{j=1}^m c_j q(x_j) \text{ and } \iint \Phi(x, y) d\mu(x) d\mu(y) = \sum_{i=1}^m \sum_{j=1}^m c_i c_j \Phi(x_i, x_j).$$

The motivation behind the current definition is that if $D \subset \mathbb{R}^n$ is compact then the dual $C(D)^*$ of $C(D)$ is the set of functionals $\mu(f) = \int_D f(x) d\mu(x)$, with μ a regular Borel measure on D . Hence, if we want a definition of positive definiteness appropriate for interpolation problems which involve a mixture of point values and weighted averages it is natural to require only continuity for Φ and to allow functionals that are regular Borel measures. If we were concerned with Hermite interpolation then a different definition of positive definite, requiring at least greater smoothness, would be appropriate. See Wu [18], Sun [16], and Narcowich [13] for some possibilities.

Given a function f , and m compactly supported regular Borel measures μ_i , we will seek an interpolant s such that

$$\mu_i(s) = \mu_i(f), \quad \text{for all } 1 \leq i \leq m.$$

Often we will not know f but only some observations of it. For example if $\mu_i(f)$ is an average over a ball \mathcal{B} and f_1, \dots, f_N are observations of $f(x)$ at points x_1, \dots, x_N then

$$\mu_i(f) = \int_{\mathcal{B}} f(x) d\mu_i(x) = \text{average value of } f \text{ on } \mathcal{B} \approx \frac{1}{\#\{j : x_j \in \mathcal{B}\}} \sum_{j: x_j \in \mathcal{B}} f_j.$$

Hence it is reasonable to take the experimentally observed average value as an approximation to the unknown continuous average, and interpolate to it rather than the continuous average. A possible configuration of regions over which to average, and observation locations, is shown in Figure 1.

Formulated as in the previous paragraph the integral interpolation approach is very much a direct generalisation of the one dimensional histospline technique of Boneva, Kendall and Stefanov [3]. Several such generalisations have been given previously. In particular Schoenberg [15] discusses tensor product histosplines, Duchon [7, Theorems 2 and 4] has a general theory which

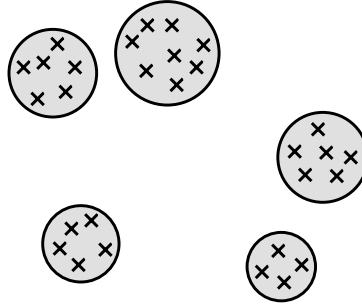


Fig. 1. A possible configuration of data points and regions over which to average.

covers integral interpolation by pseudo splines and polyharmonic splines, and Dyn and Wahba [8] present a theory that covers integral interpolation with polyharmonic splines. Our contribution here covers some different *parent basic functions* and has an emphasis on the practical computational issues. In particular it emphasizes the explicit formulas available for averages over line segments and balls which lower the number of floating point operations required to use the technique dramatically, making it practical for much larger problems.

We will need the following definition.

Definition 2. A set of linear functionals $\mu_i, 1 \leq i \leq m$ will be called *unisolvant* for π_{k-1}^n if

$$q \in \pi_{k-1}^n \text{ and } \mu_j(q) = 0 \text{ for all } 1 \leq j \leq m \implies q \text{ is the zero polynomial.}$$

We consider integral interpolation problems of the following form:

Problem 1 (Integral interpolation). Let Φ be an $\text{ISPD}_k(\mathbb{R}^n)$ kernel. Let μ_1, \dots, μ_m be linearly independent compactly supported linear functionals on $C(\mathbb{R}^n)$ which are unisolvant for π_{k-1}^n . Let b_1, \dots, b_m be m real values. Find a function s of the form

$$s(x) = p(x) + \sum_{j=1}^m c_j \int_{\mathbb{R}^n} \Phi(x, y) d\mu_j(y), \quad p \in \pi_{k-1}^n, \tag{1}$$

such that

$$\int s(x) d\mu_i(x) = b_i, \quad 1 \leq i \leq m,$$

and

$$\sum_{j=1}^m c_j \int q(x) d\mu_j(x) = 0, \quad \text{for all } q \in \pi_{k-1}^n.$$

In the pointwise interpolation case the function s has the form

$$s(x) = p(x) + \sum_{j=1}^m c_j \Phi(x, x_j). \quad (2)$$

The expression (1) for s justifies the name *parent basic function* for Φ used previously, since when interpolating with general functionals, we seek an approximation made up of polynomials plus functions like $\mu_j^y(\Phi(x, y)) = \int_{\mathbb{R}^n} \Phi(x, y) d\mu_j(y)$ derived from Φ . Under weak conditions on the “geometry/independence” of the functionals the derived functions form a *compatible family*. That is they form a family of functions for which the corresponding interpolation matrix has positive definiteness properties making the interpolation problem uniquely solvable.

In order to be more concrete let $\ell = \dim(\pi_{k-1}^n)$ and $\{p_1, \dots, p_\ell\}$ be a basis of π_{k-1}^n . Then the integral interpolation problem above can be rewritten in matrix form as:

Problem 2 (Integral interpolation matrix form). Solve

$$\begin{bmatrix} G & LP \\ (LP)^T & O \end{bmatrix} \begin{bmatrix} c \\ a \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} \quad (3)$$

for vectors c and a where G is $m \times m$ with

$$G_{ij} = \iint \Phi(x, y) d\mu_i(x) d\mu_j(y),$$

LP is $m \times \ell$ with $(LP)_{ij} = \int p_j(x) d\mu_i(x)$, and $p = \sum_{j=1}^{\ell} a_j p_j$.

Considering this problem a slight reworking of standard arguments from the pointwise positive definite case shows:

Theorem 1. *Let Φ be an $\text{ISPD}_k(\mathbb{R}^n)$ kernel. Let $\{\mu_1, \dots, \mu_m\}$ be independent compactly supported regular Borel measures on $C(\mathbb{R}^n)$ which are unisolvent for π_{k-1}^n . Then the integral interpolation problem, Problem 1, has a unique solution. The coefficients of this solution may be found by solving the linear system of Problem 2.*

For the sake of completeness a proof of this theorem is given in Section 2.

The theory above is a direct generalisation of the pointwise, or Lagrange, interpolation case and is very satisfactory. However, integral interpolation would be impractical for large problems if numerical quadrature was required in order to evaluate the fitted function s of equation (1), and if two dimensional or higher quadrature had to be used to form the entries of the matrix G of the fitting problem, Problem 2. Fortunately, usually for averages over line segments no quadrature is needed to evaluate the interpolant s , and only univariate quadrature is needed in finding the entries of the matrix G . For

averages over balls usually all needed quantities can be given in closed form, and no quadrature is needed in either evaluation or fitting.

The layout of the paper is as follows. In Section 2 we recall and enhance some results of Light [11]. Light showed that the well known theorem of Micchelli [12] connecting complete monotonicity and pointwise conditional positive definiteness extends to integral conditional positive definiteness. This provides us with a rich collection of integrally strictly conditionally positive definite functions. The work of Iske [10] provides an alternative criterion for integral positive definiteness relating it to the positivity of the Fourier or generalised Fourier transform. In Section 3 we discuss a sufficient condition for unsolvency and a way of replacing the linear system (3) with a positive definite system. The reader whose primary interest is in applications may wish to skip Sections 2 and 3 on the first reading. In Section 4 we list several integrally strictly conditionally positive definite functions and give line segment sources derived from them in closed form. In Section 5 we describe several ball sources in \mathbb{R}^3 . Finally Section 6 describes a greedy algorithm for fitting track data via integral interpolation.

In the rest of the paper we will assume that the Φ is of the special form $\Phi(x, y) = \psi(|x - y|)$ for some $\psi : \mathbb{R} \rightarrow \mathbb{R}$. We will therefore change notation and write $\Phi(x)$ where Φ is radial. This amounts to replacing $\Phi(x, y)$ by $\Phi(x - y)$ in everything above.

2 Integral Interpolation and Interpolation with General Functionals

In this section we discuss integral interpolation and interpolation with general functionals. We discuss an analogue due to Light [11] of Micchelli's Theorem for completely monotone functions. This provides us with a rich source of strictly integrally conditionally positive definite functions of order k .

Consider Hermite piecewise cubic interpolation in one variable with data at the points $t_0 < t_1 < \dots < t_m$. After some work it is possible to express such an interpolant in the form

$$h(x) = p_1(x) + \sum_{i=0}^m c_i |x - t_i|^3 - \sum_{i=0}^m d_i \mathfrak{Z}(x - t_i) |x - t_i|$$

where

$$\sum_i c_i = 0 = \sum_i (d_i + c_i t_i).$$

In this expression note that the derivative interpolations we wish to make at the points t_i have introduced kernels $\frac{d}{dy} \Phi(x - y)$ into the spline/radial basis function. Here $\Phi(x - y) = |x - y|^3$ is the usual kernel arising when natural cubic spline interpolation is viewed as an example of radial basis function (RBF) interpolation.

The example above is one instance of a much more general pattern. Namely that when interpolating with general functionals μ_i in a symmetric way the kernels $\Phi(x-y)$ appropriate for point evaluations should be replaced by kernels $\mu_i^y(\Phi(x-y))$. The pattern is clear in the papers of Iske [10], Narcowich [13], Franke and Schaback [9], and others. It is this pattern which motivated us to setup the integral interpolation problem as in Problem 1.

In order to use the solution to Problem 1 given in Theorem 1 we need to show that there exist some radial functions Φ which are $\text{ISPD}_k(\mathbb{R}^n)$. Note that it is easy to show that strict pointwise positive definiteness of Φ implies integral positive definiteness of Φ . Unfortunately, this is not enough, the strictness is essential for the poisedness of the integral interpolation problem.

To identify some $\text{ISPD}_k(\mathbb{R}^n)$ functions, one can modify A.L. Brown's elegant *density proof* in [4], or otherwise show:

Lemma 1 (A.L. Brown). *Let $\sigma > 0$. The Gaussian $\Phi(x) = \exp(-\sigma x^2)$ is strictly integrally positive definite on \mathbb{R}^n for every n .*

Then one can generalise the result of Micchelli [12] for the pointwise positive definite case obtaining

Theorem 2 (W.A. Light [11]). *Let $\eta \in C[0, \infty)$ with $(-1)^k \eta^{(k)}$ completely monotonic and not constant on $(0, \infty)$. Then $\Phi(x) = \eta(|x|^2)$ is integrally strictly conditionally positive definite of order k on \mathbb{R}^n , for all n .*

This theorem provides us with a plentiful collection of integrally strictly conditionally positive definite functions. See Section 4 for some examples.

Light actually proved the Theorem for the cases $k = 0$ and $k = 1$. We briefly outline a proof along the lines of Micchelli [12] for general k .

Sketch proof of Theorem 2. Firstly an argument almost identical to the original one in [12, Lemma 3.1] gives

Lemma 2. *Let μ be a compactly supported regular Borel measure such that $\int_{\mathbb{R}^n} q(x) d\mu(x) = 0$ for all $q \in \pi_{k-1}^n$. Then*

$$\iint |x - y|^{2k} d\mu(x) d\mu(y) \geq 0, \tag{4}$$

and equality holds in (4) if and only if

$$\int q(x) d\mu(x) = 0, \quad \text{for all } q \in \pi_k^n. \tag{5}$$

Now consider a function $\eta \in C[0, \infty)$ for which $(-1)^k \eta^{(k)}(t)$ is completely monotone but nonconstant on $(0, \infty)$. Then $(-1)^k \eta^{(k)}(t)$ necessarily tends to a finite nonnegative limit, c , as $t \rightarrow \infty$. Using the Bernstein-Widder theorem there is a finite nonnegative Borel measure ν so that

$$(-1)^k \eta^{(k)}(t) = \int_0^\infty e^{-st} d\nu(s),$$

for all $t > 0$. As noted in [5, p. 135] $c = \lim_{t \rightarrow \infty} (-1)^k \eta^{(k)}(t) = \nu(\{0\})$.

In order to make the proof of Theorem 2 more transparent we want to separate the influence of the point mass at zero and the integral against the measure. Therefore we write

$$(-1)^k \eta^{(k)}(t) = c + \int_{0+}^{\infty} e^{-st} d\nu(s), \quad t > 0,$$

where the integral now definitely does not involve any point mass at zero. This corresponds to splitting η into a polynomial part $q_k(t) = ct^k/k! +$ lower degree terms, and a part $F = \eta - q_k$ which is in $C[0, \infty)$ with $(-1)^k F^{(k)}$ completely monotonic but nonconstant on $(0, \infty)$. By construction $\lim_{t \rightarrow \infty} F^{(k)}(t) = 0$, and the measure occurring in the Bernstein-Widder representation of $(-1)^k F^{(k)}$ has no point mass at zero. That measure is $\nu - \nu(\{0\})\delta_0$.

Consider now a nonzero compactly supported regular Borel measure μ which annihilates π_{k-1}^n . Then applying Lemma 2 to the polynomial q_k which occurs in the splitting of η

$$\iint q_k(|x - y|^2) d\mu(x) d\mu(y) = \frac{c}{k!} \iint |x - y|^{2k} d\mu(x) d\mu(y) \geq 0.$$

That is $q_k(| \cdot |^2)$ and $c| \cdot |^{2k}/k!$ are integrally conditionally positive definite of order k , but not strictly so.

Considering the other part of the splitting, and writing $F_\epsilon(t) = F(t + \epsilon)$, calculations identical to those in [12, p. 17], modulo applying Fubini instead of operating with finite sums, and using Lemma 2 rather than its pointwise analogue, yield

$$\begin{aligned} \iint F(|x - y|^2 + \epsilon) d\mu(x) d\mu(y) \\ = \int_{0+}^{\infty} e^{-\sigma\epsilon} \sigma^{-k} \left\{ \iint e^{-|x-y|^2\sigma} d\mu(x) d\mu(y) \right\} d\nu(\sigma). \end{aligned}$$

Now since $F^{(k)}$ is nonconstant there exists $a > 0$ so that $\int_a^{2a} 1 d\nu(\sigma) > 0$. Also, since $\mu \neq 0$, Lemma 1 implies that the quantity in curly brackets, $\{ \}$, above is a positive and continuous function of $\sigma > 0$. Hence it has a positive lower bound on the compact set $[a, 2a]$. Therefore for all sufficiently small $\epsilon > 0$

$$\begin{aligned} \iint F(|x - y|^2 + \epsilon) d\mu(x) d\mu(y) \\ = \int_{0+}^{\infty} \sigma^{-k} \exp(-\epsilon\sigma) \left\{ \iint \exp(-\sigma|x - y|^2) d\mu(x) d\mu(y) \right\} d\nu(\sigma) \\ > \frac{1}{2}(2a)^{-k} \int_a^{2a} \left\{ \iint \exp(-\sigma|x - y|^2) d\mu(x) d\mu(y) \right\} d\nu(\sigma) \geq C > 0. \end{aligned}$$

Taking the limit as $\epsilon \searrow 0$ shows $\iint F(|x - y|^2) d\mu(x) d\mu(y) > 0$, which implies $F(| \cdot |^2)$ is $\text{ISPD}_k(\mathbb{R}^n)$. It follows that $\eta(| \cdot |^2) = q_k(| \cdot |^2) + F(| \cdot |^2)$ is also $\text{ISPD}_k(\mathbb{R}^n)$, the desired result. \square

For the sake of completeness we now give a proof of Theorem 1.

Proof of Theorem 1. Consider the case when the right hand side of the linear system (3) is zero. Mimicking well known arguments from the pointwise positive definite case multiply the first row of the block system (3) on the left by c^T . This yields

$$0 = c^T Gc + c^T(LP) = c^T Gc \quad \text{since} \quad (LP)^T c = 0.$$

From the strict conditional positive definiteness of Φ this implies $c = 0$. Substituting back the first row of the block system becomes $(LP)a = 0$. But $(LP)a$ is a vector whose i -th component is μ_i applied to the polynomial $q = \sum_{j=1}^{\ell} a_j p_j$. Hence the unisolvency implies $a = 0$. Therefore the only solution to the homogeneous equation is the trivial one and the matrix on the left of equation (3) is invertible. Hence, there is a unique solution for any given right hand side. \square

3 Computational Issues

In this section we address some computational issues.

In the Lagrange interpolation setting it is very useful that the unisolvency condition of the appropriate variant of Theorem 1 can be checked very quickly when only linear polynomials are involved. Specifically, a set of point evaluations is unisolvent for π_1^n if and only if there is no single hyperplane containing all the points.

For integral functionals we have the following related sufficient condition:

Lemma 3. *Let $\mathcal{C} = \{\nu_1, \dots, \nu_m\}$ be a set of $m \geq n + 1$ linearly independent compactly supported regular Borel measures on \mathbb{R}^n . Suppose that there is a subset $\mathcal{B} = \{\mu_1, \dots, \mu_{n+1}\}$ of \mathcal{C} such that each element in \mathcal{B} is a positive measure. Associate with each μ_i a corresponding connected compact set A_i so that $\text{supp}(\mu_i) \subset A_i$. If the sets $\{A_i, 1 \leq i \leq n + 1\}$ can be chosen to be disjoint, and such that no one hyperplane intersects them all, then the set of functionals \mathcal{C} is unisolvent for linears on \mathbb{R}^n .*

Proof. It suffices to prove that a set \mathcal{B} of $n + 1$ measures with the properties listed in the statement of the lemma is unisolvent for linears. We carry out the details in the special case of \mathbb{R}^2 . The generalisation to \mathbb{R}^n is obvious.

Let $\{p_1, p_2, p_3\}$ be a basis for the linears. Then the pointwise interpolation determinant

$$D(x, y, z) = \begin{vmatrix} p_1(x) & p_2(x) & p_3(x) \\ p_1(y) & p_2(y) & p_3(y) \\ p_1(z) & p_2(z) & p_3(z) \end{vmatrix}$$

is nonzero for any $x \in A_1, y \in A_2, z \in A_3$ since these points are not collinear. Therefore, by the Intermediate Value Theorem, this determinant must have constant sign for $x \in A_1, y \in A_2, z \in A_3$. Integrating we find

$$\begin{aligned} & \begin{vmatrix} \int p_1(x) d\mu_1 & \int p_2(x) d\mu_1 & \int p_3(x) d\mu_1 \\ \int p_1(y) d\mu_2 & \int p_2(y) d\mu_2 & \int p_3(y) d\mu_2 \\ \int p_1(z) d\mu_3 & \int p_2(z) d\mu_3 & \int p_3(z) d\mu_3 \end{vmatrix} \\ &= \iiint D(x, y, z) d\mu_1(x) d\mu_2(y) d\mu_3(z) \neq 0. \quad \square \end{aligned}$$

Again in the point evaluation setting it is useful to replace the linear system (3) by a symmetric positive definite one. This allows solution by Cholesky decomposition, or by suitable iterative methods, improving speed and stability. We generalize the construction given in [2] for the pointwise case.

Our construction below assumes that the functionals μ_1, \dots, μ_m have been reordered if necessary so that the first ℓ are unisolvent for π_{k-1}^n . Begin by choosing Q to be any $m \times (m - \ell)$ matrix whose columns span the orthogonal complement of the column space of LP . Then

$$\begin{aligned} GQ\gamma + (LP)a = b &\implies (Q^T GQ)\gamma = Q^T b \\ &\implies Q^T (b - GQ\gamma) = 0 \\ &\implies b - GQ\gamma \text{ is in the column space of } LP. \end{aligned}$$

Therefore the system (3) can be solved as follows:

Procedure for solving the integral interpolation problem

Step 1. Solve the $(m - \ell) \times (m - \ell)$ SPD system $(Q^T GQ)\gamma = Q^T b$ for γ .

Step 2. Set $c = Q\gamma$. Set $\tilde{s} = \sum_j c_j \int \Phi(x - y) d\mu_j(y)$.

Step 3. Find the coefficients of the polynomial part by finding the $p \in \pi_{k-1}^n$ integrally interpolating the residual $(f - \tilde{s})$ with respect to the functionals μ_1, \dots, μ_ℓ . Then $s = p + \tilde{s}$.

It remains to construct a suitable matrix Q . Proceed as follows. Construct $\{p_1, \dots, p_\ell\} \subset \pi_{k-1}^n$ biorthogonal to μ_1, \dots, μ_ℓ , that is satisfying $\mu_i(p_j) = \delta_{ij}$. $(\mathcal{L}g) = \sum_{t=1}^{\ell} (\int g(x) d\mu_t(x)) p_t$ is then a projection onto π_{k-1}^n . \mathcal{L} is the Lagrange polynomial projection for the functionals μ_1, \dots, μ_ℓ . Set the j -th column of Q to

$$\left[-\int p_1 d\mu_{\ell+j}, -\int p_2 d\mu_{\ell+j}, \dots, -\int p_\ell d\mu_{\ell+j}, 0, \dots, 0, 1, 0, \dots, 0 \right]^T,$$

where the 1 is in the $(\ell + j)$ -th position. Then Q clearly has full rank. The i -th row of $(LP)^T$ is

$$\left[\int p_i d\mu_1, \int p_i d\mu_2, \dots, \int p_i d\mu_m \right].$$

Therefore the ij element of $(LP)^T Q$ is

$$\begin{aligned}
& \int p_i d\mu_{\ell+j} - \sum_{t=1}^{\ell} \left(\int p_i d\mu_t \right) \int p_t d\mu_{\ell+j} \\
&= \int \left\{ p_i - \sum_{t=1}^{\ell} \left(\int p_i d\mu_t \right) p_t \right\} d\mu_{\ell+j} \\
&= \int \{p_i - \mathcal{L}(p_i)\} d\mu_{\ell+j} = 0.
\end{aligned}$$

Thus $(LP)^T Q = 0$ as required.

4 Some Explicit Line Sources

In this section we consider interpolation problems in which the data to be fitted is a mixture of point values and averages over line segments. In view of the formulation given in the introduction we will choose a *parent basic function* Φ and interpolate using a combination of a low degree polynomial and line segment sources derived from Φ .

The (uniform weight) line segment source derived from Φ and corresponding to a line segment $\langle \mathbf{a}, \mathbf{b} \rangle \subset \mathbb{R}^n$ has value at \mathbf{x}

$$\Psi(\langle \mathbf{a}, \mathbf{b} \rangle, \mathbf{x}) := \frac{1}{|\mathbf{b} - \mathbf{a}|} \int_{\mathbf{y} \in \langle \mathbf{a}, \mathbf{b} \rangle} \Phi(\mathbf{x} - \mathbf{y}) d\mathbf{y}.$$

Note that the integral is weighted by the inverse of the length of the interval being integrated over. This normalisation ensures that as the segment shrinks to a point the line segment source converges to the corresponding parent basic function. The normalisation also helps the conditioning of the linear systems (3) being used to calculate integral interpolants.

In order to give explicit formulas for some of these line sources we standardise on a geometry as in Figure 2 below. In the diagram d is the perpendicular distance from the evaluation point \mathbf{x} to the line through points \mathbf{a} and \mathbf{b} . \mathbf{p} is the footpoint, the projection of \mathbf{x} on the line through points \mathbf{a} and \mathbf{b} . a and b are the signed distances of \mathbf{a} , respectively \mathbf{b} , from this footpoint with the direction from \mathbf{a} to \mathbf{b} taken as positive. The “coordinates” a , b and d are trivial to calculate. Explicitly the footpoint is given by

$$\mathbf{p} = \mathbf{a} + \left\{ (\mathbf{x} - \mathbf{a})^T \mathbf{u} \right\} \mathbf{u} \quad \text{where} \quad \mathbf{u} = \frac{\mathbf{b} - \mathbf{a}}{|\mathbf{b} - \mathbf{a}|},$$

and then

$$a = (\mathbf{a} - \mathbf{p})^T \mathbf{u}, \quad b = (\mathbf{b} - \mathbf{p})^T \mathbf{u}, \quad \text{and} \quad d^2 = (\mathbf{x} - \mathbf{p})^T (\mathbf{x} - \mathbf{p}).$$

We proceed to give explicit closed forms for various line segment sources. This enables us to evaluate the final fitted function s of (1) without any

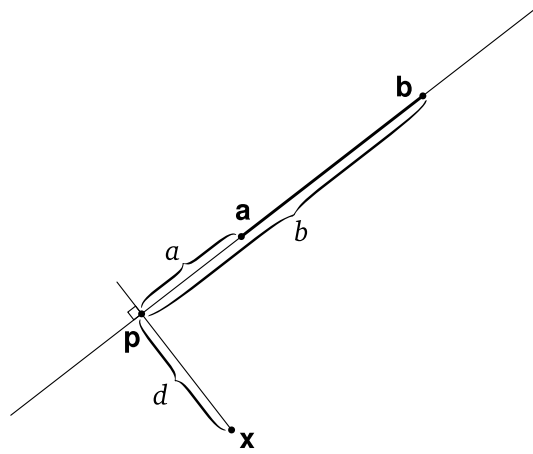


Fig. 2. Line integral parameters.

quadrature, and to form the matrix G of the fitting equations (3) with only univariate quadrature. Contour and surface plots of these line source basic functions are given in Figures 3 and 4 below. In all cases the stated positive definiteness properties follow from Theorem 2.

4.1 Gaussian Line Source

The Gaussian is integrally strictly positive definite on \mathbb{R}^n ($\text{ISPD}_0(\mathbb{R}^n)$) for all n .

$$\Phi(\mathbf{x}) = \exp(-\nu^2 \mathbf{x}^2), \quad \mathbf{x} \in \mathbb{R}^n, \quad \nu > 0.$$

$$|\mathbf{b} - \mathbf{a}| \Psi(\langle \mathbf{a}, \mathbf{b} \rangle, \mathbf{x}) = \frac{\sqrt{\pi}}{2\nu} \exp(-\nu^2 d^2) \{ \text{erf}(\nu b) - \text{erf}(\nu a) \}.$$

4.2 Linear Line Source

The linear basic function is $\text{ISPD}_1(\mathbb{R}^n)$ for all n . RBFs of the form (2) based on this Φ and linear polynomials are biharmonic splines in \mathbb{R}^3 .

$$\Phi(\mathbf{x}) = |\mathbf{x}|, \quad \mathbf{x} \in \mathbb{R}^n.$$

$$|\mathbf{b} - \mathbf{a}| \Psi(\langle \mathbf{a}, \mathbf{b} \rangle, \mathbf{x}) = \frac{1}{2} \left\{ b\sqrt{d^2 + b^2} + d^2 \ln \left(b + \sqrt{d^2 + b^2} \right) \right\} - \frac{1}{2} \left\{ a\sqrt{d^2 + a^2} + d^2 \ln \left(a + \sqrt{d^2 + a^2} \right) \right\}.$$

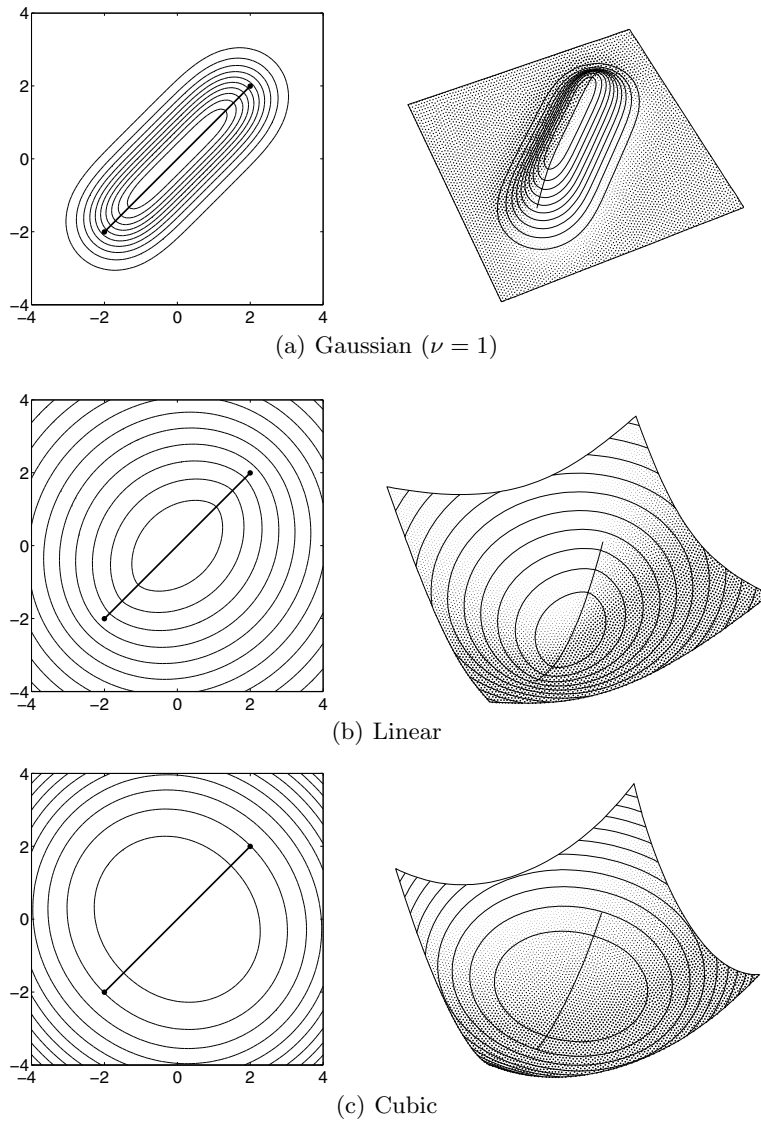


Fig. 3. Line source basic functions.

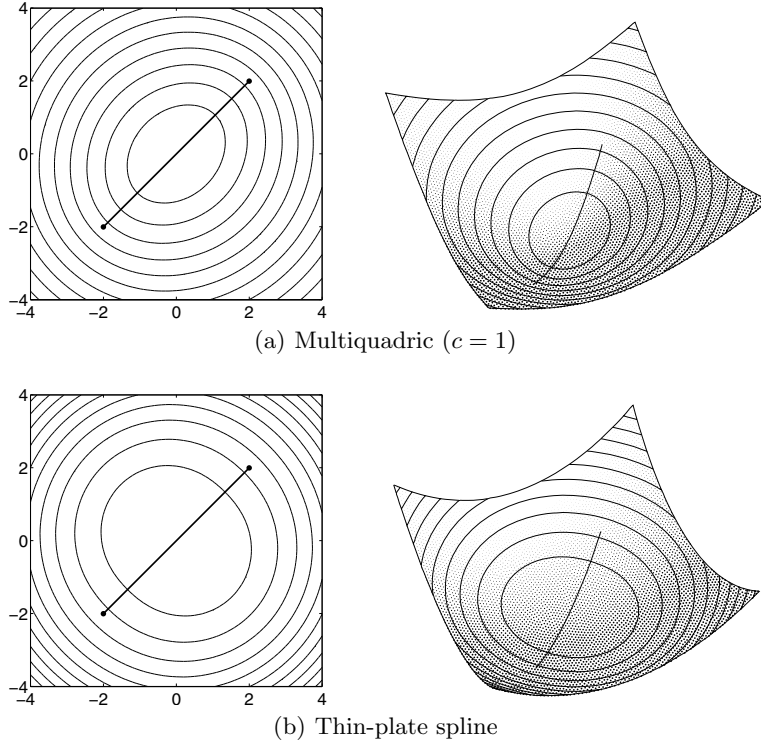


Fig. 4. Line source basic functions continued.

4.3 Cubic Line Source

The cubic basic function is $\text{ISPD}_2(\mathbb{R}^n)$ for all n . RBFs of the form (2) based on this Φ and quadratic polynomials are triharmonic splines in \mathbb{R}^3 .

$$\Phi(\mathbf{x}) = |\mathbf{x}|^3, \quad \mathbf{x} \in \mathbb{R}^n.$$

$$\begin{aligned} & |\mathbf{b} - \mathbf{a}| \Psi(\langle \mathbf{a}, \mathbf{b} \rangle, \mathbf{x}) \\ &= \frac{1}{8} \left\{ 2b (d^2 + b^2)^{3/2} + 3d^2 b \sqrt{d^2 + b^2} + 3d^4 \ln \left(b + \sqrt{d^2 + b^2} \right) \right\} \\ & \quad - \frac{1}{8} \left\{ 2a (d^2 + a^2)^{3/2} + 3d^2 a \sqrt{d^2 + a^2} + 3d^4 \ln \left(a + \sqrt{d^2 + a^2} \right) \right\}. \end{aligned}$$

4.4 Multiquadric Line Source

The multiquadric basic function is $\text{ISPD}_1(\mathbb{R}^n)$ for all n .

$$\begin{aligned}\Phi(\mathbf{x}) &= \sqrt{\mathbf{x}^2 + c^2}, \quad \mathbf{x} \in \mathbb{R}^n, \quad c > 0. \\ |\mathbf{b} - \mathbf{a}| \Psi(\langle \mathbf{a}, \mathbf{b} \rangle, \mathbf{x}) &= \\ & \frac{1}{2} \left\{ b \sqrt{d^2 + b^2 + c^2} + (d^2 + c^2) \ln \left(b + \sqrt{d^2 + b^2 + c^2} \right) \right\} \\ & - \frac{1}{2} \left\{ a \sqrt{d^2 + a^2 + c^2} + (d^2 + c^2) \ln \left(a + \sqrt{d^2 + a^2 + c^2} \right) \right\}.\end{aligned}$$

4.5 Thinplate Spline Line Source

The thinplate basic function is $\text{ISPD}_2(\mathbb{R}^n)$ for all n . RBFs of the form (2) based on this Φ and linear polynomials are biharmonic splines in \mathbb{R}^2 .

$$\begin{aligned}\Phi(\mathbf{x}) &= |\mathbf{x}|^2 \ln |\mathbf{x}|, \quad \mathbf{x} \in \mathbb{R}^n. \\ |\mathbf{b} - \mathbf{a}| \Psi(\langle \mathbf{a}, \mathbf{b} \rangle, \mathbf{x}) &= \left\{ d^2 b \left(\ln(d^2 + b^2) - \frac{4}{3} \right) + \frac{4d^3}{3} \arctan \left(\frac{b}{d} \right) \right. \\ & \quad \left. + \frac{b^3}{9} \left(3 \ln(d^2 + b^2) - 2 \right) \right\} \\ & - \left\{ d^2 a \left(\ln(d^2 + a^2) - \frac{4}{3} \right) + \frac{4d^3}{3} \arctan \left(\frac{a}{d} \right) \right. \\ & \quad \left. + \frac{a^3}{9} \left(3 \ln(d^2 + a^2) - 2 \right) \right\}.\end{aligned}$$

5 Some Explicit Ball Sources in \mathbb{R}^3

In this section we develop explicit formulas for ball sources in \mathbb{R}^3 . Our first motivation is to fit noisy point values by performing integral interpolation to averages of these values over spheres. Such interpolation should be useful in extracting low frequency trends from noisy data. One possible configuration of datapoints and spheres over which to perform integral interpolation is shown in Figure 1. Fortunately, the radial symmetry allows us to calculate all required functions and matrix entries explicitly. Thus there is no need for any numerical integration when performing integral interpolation with ball shaped regions and the parent basic functions considered here.

The formulas developed here could also be used for data smoothing via the implicit smoothing technique of [1]. In that technique one first interpolates to noisy data using the basic function Φ . Then on evaluation one replaces Φ by the smoother function $\Psi = \Phi * K$. If Ψ is known analytically the technique can be applied without performing any convolutions or FFTs. The formulas of this

section show what Ψ is for various choices of parent Φ , when K is chosen as the normalised characteristic function of a sphere with radius c . Formulas for other ball sources, including some derived from compactly supported functions such as the Wendland function [17], can be found in [1].

We proceed to develop the formulas. Define the normalised characteristic function of the sphere with radius c , center the origin, $\mathcal{B}_c(\mathbf{x})$, as follows.

$$\mathcal{B}_c(\mathbf{x}) = \begin{cases} \frac{3}{4\pi c^3}, & |\mathbf{x}| \leq c, \\ 0, & |\mathbf{x}| > c. \end{cases}$$

Clearly the integral of this function over \mathbb{R}^3 is 1. Ball sources made from the convolutions $(\Phi * \mathcal{B}_c)(\mathbf{x})$ can usually be calculated explicitly when Φ is radial.

To calculate the convolutions use the operators

$$(If)(r) = \int_r^\infty sf(s)ds, \quad (Dg)(r) = -\frac{1}{r} \frac{dg}{dr}, \quad r \geq 0,$$

which satisfy

$$f *_{n+2} g = 2\pi D(If *_{n+2} Ig), \tag{6}$$

for compactly supported bounded radial functions f and g . Here, we use the notation f, g both for the even functions of one variable $f(r), g(r)$, and also for the radial functions of several variables $f(|\mathbf{x}|)$ and $g(|\mathbf{x}|)$, with $\mathbf{x} \in \mathbb{R}^d$. The subscript on the convolution symbol $*$ denotes the dimension d . Thus $f *_{n+2} g$ denotes the convolution in \mathbb{R}^{n+2} of the radial functions of $n + 2$ variables $f(|\mathbf{x}|)$ and $g(|\mathbf{x}|)$.

In the approximation theory context these formulas were developed by Wendland [17], based on previous work of Schaback and Wu [14] and Wu [19]. However, they had been previously discovered in the geostatistical context by Matheron. See Chiles and Delfiner [6] for references to relevant geostatistical literature.

In order to use these formulas on non compactly supported functions we need to truncate and shift. For example, consider calculating the convolution of the Gaussian $\Phi(x) = \exp(-\nu^2 x^2)$ with the function \mathcal{B}_c . Then instead of $\Phi(\mathbf{x})$ we use

$$f(x) = \begin{cases} \exp(-\nu^2 x^2) - \exp(-\nu^2 N^2), & |x| < N, \\ 0, & |x| \geq N, \end{cases}$$

where N is large. We then calculate the convolution $f *_{\mathbb{R}^3} \mathcal{B}_c$ using the formula (6). Since convolution with \mathcal{B}_c preserves constants, it follows that

$$(\Phi * \mathcal{B}_c)(\mathbf{x}) = (f * \mathcal{B}_c)(\mathbf{x}) + \exp(-\nu^2 N^2),$$

for all $|\mathbf{x}| < N - c$. Clearly the same device can be used for other non compactly supported radial basic functions.

5.1 Gaussian Ball Source

$$\begin{aligned}\Phi(\mathbf{x}) &= \exp(-\nu^2 x^2), \quad \mathbf{x} \in \mathbb{R}^3. \\ (\Phi * \mathcal{B}_c)(x) &= \frac{3}{8c^3\nu^4|x|} \\ &\quad \times \left\{ \left[\exp(-\nu^2(|x|+c)^2) - \exp(-\nu^2(|x|-c)^2) \right] \right. \\ &\quad \left. + \sqrt{\pi}\nu|x| \left[\operatorname{erf}(\nu(|x|+c)) - \operatorname{erf}(\nu(|x|-c)) \right] \right\}.\end{aligned}$$

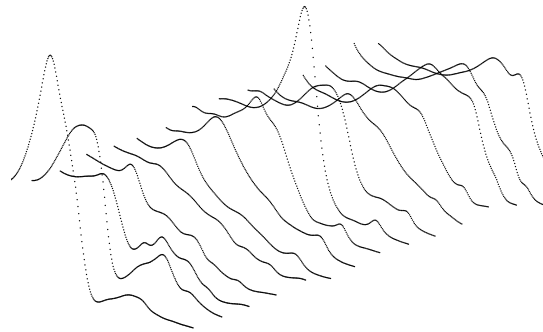
5.2 Linear / Biharmonic Ball Sources:

$$\begin{aligned}\Phi(\mathbf{x}) &= |\mathbf{x}|, \quad \mathbf{x} \in \mathbb{R}^3. \\ (\Phi * \mathcal{B}_c)(x) &= \begin{cases} \frac{3}{4}c + \frac{|x|^2}{2c} - \frac{|x|^4}{20c^3}, & |x| < c, \\ |x| + \frac{c^2}{5|x|}, & |x| \geq c. \end{cases} \\ (\Phi * \mathcal{B}_c * \mathcal{B}_c)(x) &= \begin{cases} \frac{36}{35}c + \frac{2|x|^2}{5c} - \frac{|x|^4}{20c^3} + \frac{|x|^5}{80c^4} - \frac{|x|^7}{4480c^6}, & |x| < 2c, \\ |x| + \frac{2c^2}{5|x|}, & |x| \geq 2c. \end{cases}\end{aligned}$$

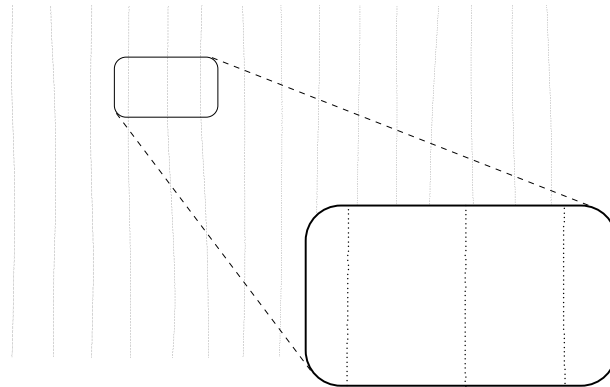
5.3 Cubic / Triharmonic Ball Source

$$\begin{aligned}\Phi(\mathbf{x}) &= |\mathbf{x}|^3, \quad \mathbf{x} \in \mathbb{R}^3. \\ (\Phi * \mathcal{B}_c)(x) &= \begin{cases} \frac{1}{2}c^3 + \frac{3c|x|^2}{2} + \frac{3|x|^4}{10c} - \frac{|x|^6}{70c^3}, & |x| < c, \\ |x|^3 + \frac{6c^2|x|}{5} + \frac{3c^4}{35|x|}, & |x| \geq c. \end{cases} \\ (\Phi * \mathcal{B}_c * \mathcal{B}_c)(x) &= \begin{cases} \frac{32}{21}c^3 + \frac{72c|x|^2}{35} + \frac{6|x|^4}{25c} - \frac{|x|^6}{70c^3} + \frac{3|x|^7}{1120c^4} - \frac{|x|^9}{33600c^6}, & |x| < 2c, \\ |x|^3 + \frac{12c^2|x|}{5} + \frac{72c^4}{175|x|}, & |x| \geq 2c. \end{cases}\end{aligned}$$

6 An Application: Approximating Track Data with Line Sources



(a) Variation in the gravitational attraction.



(b) Track variation.

Fig. 5. Two views of the airborne gravity survey dataset.

In this section we describe a simple greedy algorithm which uses line sources to approximate a track dataset. The motivation is that the sampling along a track is orders of magnitude denser than in the between track direction. It therefore makes little sense to have a point source for every measured point value. Rather we consider approximating a “segment” of point sources by a single line (segment) source. We will develop a greedy algorithm approach to the fitting task and illustrate it by applying it to an airborne gravity survey.

The test data set is a subset of 3351 points taken from a large airborne gravity survey. Two views of the data are given in Figure 5. Note from the top down view that the tracks flown by the aircraft are not straight, and that the “signal” is sampled approximately 24 times more densely in the along

track direction than in the between track direction. Also note that the 3D view shows little “high frequency” variation along a track. We interpret this as meaning that the data will be well fitted by a smooth surface and that the measurements contain little random noise. Therefore there is no need to use a spline smoothing variant of integral interpolation for this dataset.

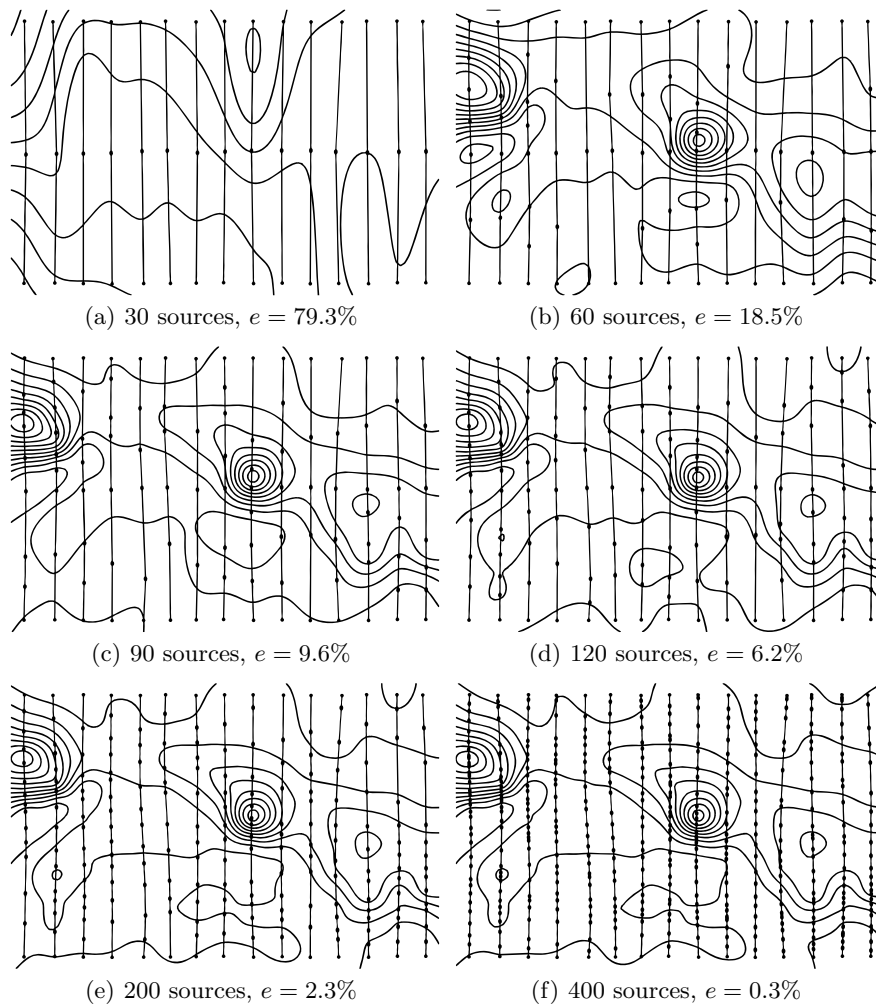


Fig. 6. The greedy algorithm applied to an airborne gravity survey. The approximation is by line sources derived from the thinplate basic function, $\Phi(x) = |\mathbf{x}|^2 \log \mathbf{x}$. e is the relative ℓ_1 error.

A Simple Greedy Algorithm for Integral Interpolation to Track Data

- Step 1.** Divide the data points up into tracks. For each track form a direction vector and use it to order the data along the track.
- Step 2.** Initialize a list of data segments and associated data points and line sources, by making a coarse subdivision of the tracks into line segments.
- Step 3. Do until satisfied**
- Form a line source approximation s by performing integral interpolation to data averages over segments, using the current list of segments.
 - Calculate the ℓ_1 error in the approximation to the subset of data values associated with each segment.
 - Divide a segment associated with the largest ℓ_1 error at the half error point, and replace the corresponding line source by two new line sources.
- end do**

The performance of this simple greedy algorithm on the test dataset is illustrated in Figure 6. For this example the parent basic function is the thin-plate spline $\Phi(\mathbf{x}) = |\mathbf{x}|^2 \log |\mathbf{x}|$. In the figure the piecewise linear curves running up the page correspond to line sources. The curved lines are contour plots of the current fitted surface. The start and end point of a line source are indicated with a heavy dot. These start and end points are chosen as the first and last data points associated with that line source/line segment. The other points associated with such a line segment will, in general, lie close to the segment but not on it. As the algorithm progresses the line segments are divided in an adaptive way by splitting those segments corresponding to the largest ℓ_1 error at the approximate half error point. The plots in the figure clearly show the segments being split preferentially where the action is. That is, splits tend to occur where the underlying function varies most rapidly. Visually at least the behaviour of the data has already been completely captured with a 200 line source fit.

The analogous set of calculations were performed using line sources derived from the linear basic function $\Phi(\mathbf{x}) = |\mathbf{x}|$. The results, which are not shown, were very similar.

Acknowledgement

It is a pleasure to acknowledge helpful conversations with Jeremy Levesley.

References

1. R.K. Beatson and H.-Q. Bui: Mollification formulas and implicit smoothing. *Adv. Comput. Math.*, to appear.
2. R.K. Beatson, W.A. Light, and S. Billings: Fast solution of the radial basis function interpolation equations: domain decomposition methods. *SIAM J. Sci. Comput.* **22**, 2000, 1717–1740.
3. L.L. Boneva, D. Kendall, and I. Stefanov: Spline transformations: three new diagnostic aids for the statistical data analyst. *Proc. Royal Stat. Soc., Series B* **33**, 1971, 1–70.
4. A.L. Brown: Uniform approximation by radial functions. In: *Advances in Numerical Analysis II: Wavelets, subdivision algorithms and radial functions*, W. Light (ed.) (1992), Oxford University Press, Oxford, UK, 203–206.
5. E.W. Cheney and W.A. Light: *An Introduction to Approximation Theory*. Brooks/Cole, Pacific Grove, CA, 2000.
6. J.-P. Chiles and P. Delfiner: *Geostatistics: Modeling Spatial Uncertainty*. Wiley, 1999.
7. J. Duchon: Splines minimizing rotation invariant semi-norms in Sobolev spaces. In: *Constructive Theory of Functions of Several Variables*, W. Schempp and K. Zeller (eds.), Lecture Notes in Mathematics, Springer **571**, 1977, 85–100.
8. N. Dyn and G. Wahba: On the estimation of functions of several variables from aggregated data. *SIAM J. Math. Anal.* **13**, 1982, 134–152.
9. C. Franke and R. Schaback: Solving partial differential equations by collocation using radial basis functions. *Applied Mathematics and Computation* **93**, 1998, 73–82.
10. A. Iske: Reconstruction of functions from generalized Hermite-Birkhoff data. In: *Approximation Theory VIII, Vol 1, Approximation and Interpolation*, C.K. Chui and L.L. Schumaker (eds.), World Scientific, 1995, 257–264.
11. W.A. Light: Some aspects of radial basis function approximation. In: *Approximation Theory, Spline Functions and Applications*, S.P. Singh (ed.), Kluwer Academic, Boston, 1992, 163–190.
12. C.A. Micchelli: Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constructive Approximation* **2**, 1986, 11–22.
13. F.J. Narcowich: Recent developments in approximation via positive definite functions. In: *Approximation Theory IX, Volume 2, Computational Aspects*, C.K. Chui and L.L. Schumaker (eds.), Vanderbilt University Press, 1998, 221–242.
14. R. Schaback and Z. Wu: Operators on radial functions. *J. Comput. and Appl. Math.* **73**, 1996, 257–270.
15. I.J. Schoenberg: Splines and histograms. In: *Spline Functions and Approximation Theory*, A. Sharma and A. Meir (eds.), ISNM 21, Birkhäuser, 1973, 277–327.
16. X. Sun: Scattered Hermite interpolation using radial basis functions. *Linear Algebra and Its Applications* **207**, 1994, 135–146.
17. H. Wendland: Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Adv. Comput. Math.* **4**, 1995, 389–396.
18. Z. Wu: Hermite-Birkhoff interpolation of scattered data by radial basis functions. *Approx. Theory & its Appl.* **8**, 1992, 1–10.
19. Z. Wu: Compactly supported and positive definite radial functions. *Adv. Comput. Math.* **4**, 1995, 283–292.

Shape Control in Powell-Sabin Quasi-Interpolation

Carla Manni

Department of Mathematics, University of Rome “Tor Vergata”, 00133 Roma,
Italy, manni@mat.uniroma2.it

Summary. In this paper we discuss the construction and we analyze the properties of quasi-interpolants based on an extension of C^1 Powell-Sabin quadratic splines over an arbitrary triangulation of a planar domain. These quasi-interpolants possess parameters which allow to control their shape avoiding oscillations and inflections extraneous to the behaviour of the data.

1 Introduction

Bivariate splines over general triangulations of a planar domain are a fundamental tool in numerical analysis. They are commonly used to face problems arising in several different contexts: from scattered data interpolation and approximation to numerical solution of partial differential equations.

The space of C^1 quadratic splines over a Powell-Sabin refinement, [21], of an arbitrary triangulation (Powell-Sabin splines for short) is probably the most popular bivariate spline space (to deal with non gridded data) because it combines a simple structure with a significant flexibility and a sufficient smoothness which make it particularly attractive in practical applications (see for example [9, 12, 13, 20, 21, 22, 25, 26] and references quoted therein). In particular, in the last decade Powell-Sabin splines have been profitably used in the context of scattered data approximation, [9, 17, 26], and, recently, of quasi-interpolation, [20].

The term *quasi-interpolation* denotes a general approach to construct, with low computational cost, efficient local approximants to a given set of data or a given function. A quasi-interpolant (q.i.) for a given function f is usually obtained as linear combination of the elements of a suitable set of functions which are required to be positive, to ensure stability, and to have a small support to achieve local control. The coefficients of the linear combination are the values of linear functionals depending on f and on its derivatives/integrals.

Since the seminal paper [24], quasi-interpolation has received a considerable attention by many authors both in the univariate and the multivariate

setting (see for example [2, 3, 4, 5, 7, 8, 15, 16, 23] and references quoted therein) and interesting applications have been proposed in different fields.

As almost all the quasi-interpolating schemes mentioned above, Powell-Sabin splines and the q.i.s based on them, presented and analyzed in [20], do not possess additional parameters. So, it is not possible to control the shape of the built q.i.s. On the other hand, it is clear that schemes which are able to reproduce the graphical behaviour of the data are generally preferable, and in some cases necessary, in practical application.

In this paper we describe an extension of the quadratic Powell-Sabin B-splines presented in [9, 26, 27] and of some q.i.s discussed in [20] which allows the introduction of shape parameters in the basis functions and in the obtained q.i.s. The introduced parameters allow to control the shape of the built approximation. To be more precise, they act as *tension parameters* that is, for suitable values of them the graph of the q.i.s is “straighten up” avoiding inflections and oscillations extraneous to the behaviour of the data, see [6, 19] and references quoted therein.

To build q.i.s having tension properties we use the so called “parametric approach” which basically consists in constructing the required (quasi-interpolating) function as a particular *parametric* surface, [14, 18].

The remaining of the paper is divided into 5 sections. In the next one we briefly recall the construction and the basic properties of C^1 quadratic splines and of the quadratic Powell-Sabin finite element, both in the functional and in the parametric setting. In Section 3 we briefly summarize, from [9], the construction and some salient properties of quadratic Powell-Sabin B-splines and we discuss how the parametric approach allows us to extend them to a set of functions possessing tension properties. This set of functions is used in Section 4 to build some families of discrete q.i.s possessing shape parameters. Finally, we end in Sections 5 and 6 with some numerical examples and some final remarks respectively.

Through the paper bold characters denote points or vectors in the plane or in the space and the symbol $'$ denotes the transpose operator.

2 Tensioned Powell-Sabin Finite Element

For the sake of completeness, in this Section first we briefly recall the definition and the properties of C^1 quadratic Powell-Sabin splines and their local construction in any triangle of the given triangulation (Powell-Sabin finite element, [21, 22]). Then, we recall, from [18], how the parametric approach can be used to obtain a C^1 finite element possessing tension properties.

In the following the Bézier-Bernstein representation will be used to describe polynomials over triangles (see for example [9, 11, 22]).

Let T be a triangle with vertices $\mathbf{V}_{i_j} := (x_{i_j}, y_{i_j})'$, $j = 1, 2, 3$, and let (u, v, w) be the *barycentric coordinates* of a point $(x, y)' \in \mathbb{R}^2$ with respect to the triangle T , that is the values determined by the linear system

$$\begin{pmatrix} 1 & 1 & 1 \\ x_{i_1} & x_{i_2} & x_{i_3} \\ y_{i_1} & y_{i_2} & y_{i_3} \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} 1 \\ x \\ y \end{pmatrix}.$$

Let \mathbb{P}_n denote the space of algebraic polynomials of degree less than or equal to n . Any element $p \in \mathbb{P}_n$ has a unique representation in barycentric coordinates

$$p(x, y) = \sum_{i+j+k=n} b_{i,j,k} \frac{n!}{i!j!k!} u^i v^j w^k.$$

The coefficients $b_{i,j,k}$ are the Bézier ordinates of the polynomial p with respect to the triangle T . Usually, this representation is called Bézier-Bernstein representation of p and it is schematically represented by associating each coefficient $b_{i,j,k}$ with the domain point

$$(x_{i,j,k}, y_{i,j,k})' \tag{1}$$

having barycentric coordinates $(\frac{i}{n}, \frac{j}{n}, \frac{k}{n})$. The points

$$(x_{i,j,k}, y_{i,j,k}, b_{i,j,k})' \in \mathbb{R}^3, \quad i + j + k = n,$$

are the *Bézier control points* of p .

Let Ω be a polygonal domain in \mathbb{R}^2 and let Δ be a regular triangulation of Ω . We denote by

$$\mathbf{V}_l := (V_{l,x}, V_{l,y})', \quad l = 1, \dots, N_V,$$

the vertices of the given triangulation. A Powell-Sabin refinement, Δ_{PS} , of Δ is the refined triangulation, [21], obtained (see also Figure 1) by subdividing each triangle of Δ into six subtriangles as follows. Select a point, say \mathbf{C}^j , inside any triangle T^j of Δ and connect it with the three vertices \mathbf{V}_p^j , $p = 1, 2, 3$, of T^j and with the points $\mathbf{C}^{j_1}, \mathbf{C}^{j_2}, \mathbf{C}^{j_3}$ where $T^{j_1}, T^{j_2}, T^{j_3}$ are the triangles adjacent to T^j . If T^j is a boundary triangle the undefined \mathbf{C}^{j_i} are specified points (usually the midpoints) inside the corresponding boundary edges. We assume that each segment $\mathbf{C}^j \mathbf{C}^{j_i}$, $i = 1, 2, 3$, intersects the common edge of T^j and T^{j_i} in an interior point, \mathbf{M}_i^j (see Figure 1 where superscripts are omitted for graphical convenience).

We denote by $\mathcal{S}_2^1(\Delta_{PS})$ the space of *quadratic Powell-Sabin splines*, [21], that is the linear space of piecewise quadratic polynomials on Δ_{PS} belonging to $C^1(\Omega)$. The dimension of $\mathcal{S}_2^1(\Delta_{PS})$ is $3N_V$ and any element of the space is determined by its value and its gradient at the vertices of Δ , [9, 21, 22].

Now, let us summarize the local construction of an element of $\mathcal{S}_2^1(\Delta_{PS})$ in a triangle of Δ once its values and its gradients at the three vertices of the triangle are given; this construction is usually referred to as *Powell-Sabin finite element*, [21]. To simplify the notation we omit superscripts and we consider subscripts modulus 3. Let T be a triangle of Δ . Let us denote

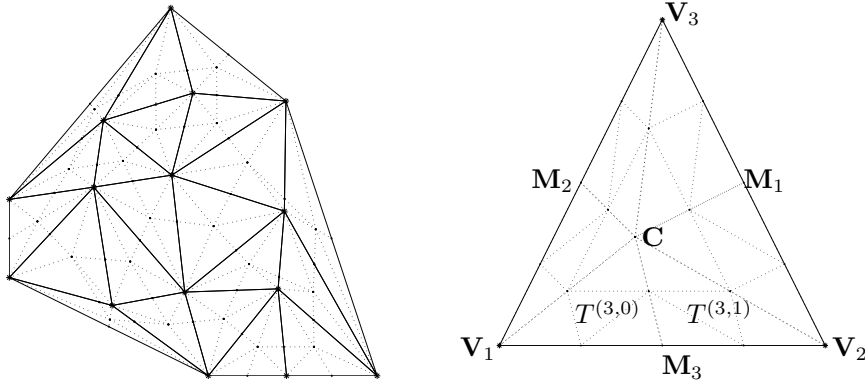


Fig. 1. Left: a Powell-Sabin refinement Δ_{PS} of a triangulation Δ . Right: Powell-Sabin refinement (split) of a single triangle and domain points for a quadratic polynomial in each subtriangle of the split.

$$T^{(p,0)} := \mathbf{V}_{p+1}\mathbf{M}_p\mathbf{C}, \quad T^{(p,1)} := \mathbf{M}_p\mathbf{V}_{p+2}\mathbf{C}, \quad p = 1, 2, 3,$$

the six subtriangles of the Powell-Sabin split of T (see Figure 1, right), where

$$\mathbf{M}_p = (1 - \alpha_p)\mathbf{V}_{p+1} + \alpha_p\mathbf{V}_{p+2}, \quad 0 < \alpha_p < 1.$$

Let a smooth function f be given. The classical Powell-Sabin finite element, $\tilde{s}_T(\cdot; f)$, is defined as

$$\tilde{s}_T|_{T^{(p,q)}}(\mathbf{P}; f) := \sum_{i+j+k=2} \frac{2!}{i!j!k!} u^i v^j w^k B_{i,j,k}^{(p,q)}(f), \quad \mathbf{P} \in T^{(p,q)},$$

$p = 1, 2, 3$, $q = 0, 1$, where (u, v, w) are the barycentric coordinates of \mathbf{P} with respect to $T^{(p,q)}$ and, denoting by \mathbf{e}_p the edge $\mathbf{V}_{p+1} - \mathbf{V}_p$,

$$\begin{aligned} B_{2,0,0}^{(p,0)}(f) &= f(\mathbf{V}_{p+1}), & B_{1,1,0}^{(p,0)}(f) &= f(\mathbf{V}_{p+1}) + \frac{\alpha_p}{2} \langle \nabla f(\mathbf{V}_{p+1}), \mathbf{e}_{p+1} \rangle, \\ B_{0,2,0}^{(p,1)}(f) &= f(\mathbf{V}_{p+2}), & B_{1,1,0}^{(p,1)}(f) &= f(\mathbf{V}_{p+2}) - \frac{1 - \alpha_p}{2} \langle \nabla f(\mathbf{V}_{p+2}), \mathbf{e}_{p+1} \rangle, \end{aligned}$$

while the remaining Bézier ordinates are determined so as to ensure C^1 continuity of \tilde{s}_T across the internal edges of the split, [21].

As it can be checked by considering its Bézier ordinates, the Powell-Sabin finite element interpolates f and its first derivatives at the vertices of T and reproduces \mathbb{P}_2 . In particular, if $\mathbf{P} := (P_x, P_y)' \in T$

$$1 = \tilde{s}_T(\mathbf{P}; 1), \quad P_x = \tilde{s}_T(\mathbf{P}; x), \quad P_y = \tilde{s}_T(\mathbf{P}; y), \quad (2)$$

thus the graph of $\tilde{s}_T(\cdot; f)$ can be interpreted as the graph of a *parametric surface* obtained applying the Powell-Sabin construction componentwise. The

parametric approach consists in inserting some parameters in such a construction, [18]. More precisely, let $\lambda_1, \lambda_2, \lambda_3 \in (0, 1]$ be given parameters. Let us consider the parametric surface $\mathbf{S}_T(\cdot; \lambda_1, \lambda_2, \lambda_3, f)$ whose components are obtained by applying componentwise the Powell-Sabin construction as follows

$$\begin{aligned} \mathbf{S}_{T|T^{(p,q)}}(\tilde{\mathbf{P}}; \lambda_1, \lambda_2, \lambda_3, f) &:= \sum_{i+j+k=2} \frac{2!}{i!j!k!} u^i v^j w^k \mathbf{B}_{i,j,k}^{(p,q)}(\lambda_1, \lambda_2, \lambda_3, f) \\ &= \begin{cases} X_T^{(p,q)}(\tilde{\mathbf{P}}; \lambda_1, \lambda_2, \lambda_3) \\ Y_T^{(p,q)}(\tilde{\mathbf{P}}; \lambda_1, \lambda_2, \lambda_3) \\ Z_T^{(p,q)}(\tilde{\mathbf{P}}; \lambda_1, \lambda_2, \lambda_3, f) \end{cases}, \quad \tilde{\mathbf{P}} \in T^{(p,q)}, \quad (3) \end{aligned}$$

$p = 1, 2, 3$, $q = 0, 1$, where

$$\mathbf{B}_{2,0,0}^{(p,0)}(\lambda_1, \lambda_2, \lambda_3, f) = \begin{pmatrix} \mathbf{V}_{p+1} \\ f(\mathbf{V}_{p+1}) \end{pmatrix} \quad (4)$$

$$\mathbf{B}_{0,2,0}^{(p,1)}(\lambda_1, \lambda_2, \lambda_3, f) = \begin{pmatrix} \mathbf{V}_{p+2} \\ f(\mathbf{V}_{p+2}) \end{pmatrix} \quad (5)$$

$$\mathbf{B}_{1,1,0}^{(p,0)}(\lambda_1, \lambda_2, \lambda_3, f) = \begin{pmatrix} \mathbf{V}_{p+1} \\ f(\mathbf{V}_{p+1}) \end{pmatrix} + \lambda_{p+1} \frac{\alpha_p}{2} \begin{pmatrix} \mathbf{e}_{p+1} \\ \langle \nabla f(\mathbf{V}_{p+1}), \mathbf{e}_{p+1} \rangle \end{pmatrix} \quad (6)$$

$$\mathbf{B}_{1,1,0}^{(p,1)}(\lambda_1, \lambda_2, \lambda_3, f) = \begin{pmatrix} \mathbf{V}_{p+2} \\ f(\mathbf{V}_{p+2}) \end{pmatrix} - \lambda_{p+2} \frac{1-\alpha_p}{2} \begin{pmatrix} \mathbf{e}_{p+1} \\ \langle \nabla f(\mathbf{V}_{p+2}), \mathbf{e}_{p+1} \rangle \end{pmatrix} \quad (7)$$

while the remaining Bézier control points are determined so as to ensure the C^1 continuity of each component of \mathbf{S}_T across the internal edges of the split. In this case the term *Bézier control points* refers to the points $\mathbf{B}_{i,j,k}^{(p,q)}$.

Thanks to (2), from (4)-(7), the graph of $\mathbf{S}_T(\cdot; 1, 1, 1, f)$ coincides with that one of $\tilde{s}_T(\cdot; f)$. On the other hand, if $\lambda_1 = \lambda_2 = \lambda_3 = 0$ the Bézier control points $\mathbf{B}_{i,j,k}^{(p,q)}(0, 0, 0, f)$ belong to the triangle in \mathbb{R}^3 with vertices (\mathbf{V}_p) , so $\mathbf{S}_T(\cdot; 0, 0, 0, f)$ reduces to the same triangle due to the properties of the Bézier-Bernstein representation. Summarizing, the parameters $\lambda_1, \lambda_2, \lambda_3$ act as tension parameters on the graph of the surface patch \mathbf{S}_T , stretching it from the classical Powell-Sabin finite element to the plane interpolating the data positions (see [18] for some graphical examples). The triangular surface patch \mathbf{S}_T will be referred to as *Powell-Sabin tensioned finite element*.

Moreover, it can be proved, [18, Theorem 3.1], that for $\lambda_1, \lambda_2, \lambda_3 \in (0, 1]$ the transformation \mathbf{T}_T defined by the first two components of \mathbf{S}_T :

$$\mathbf{T}_{T|T^{(p,q)}}(\tilde{\mathbf{P}}; \lambda_1, \lambda_2, \lambda_3) := \begin{cases} X_T^{(p,q)}(\tilde{\mathbf{P}}; \lambda_1, \lambda_2, \lambda_3) \\ Y_T^{(p,q)}(\tilde{\mathbf{P}}; \lambda_1, \lambda_2, \lambda_3) \end{cases} \quad p = 1, 2, 3, \quad q = 0, 1. \quad (8)$$

is a one-to-one map of the triangle T . Thus, the graph of \mathbf{S}_T is the graph of a bivariate function. More precisely, setting $\mathbf{P} := \mathbf{T}_{T|T^{(p,q)}}(\tilde{\mathbf{P}}; \lambda_1, \lambda_2, \lambda_3)$, the invertibility of the map \mathbf{T}_T allows us to define the following function

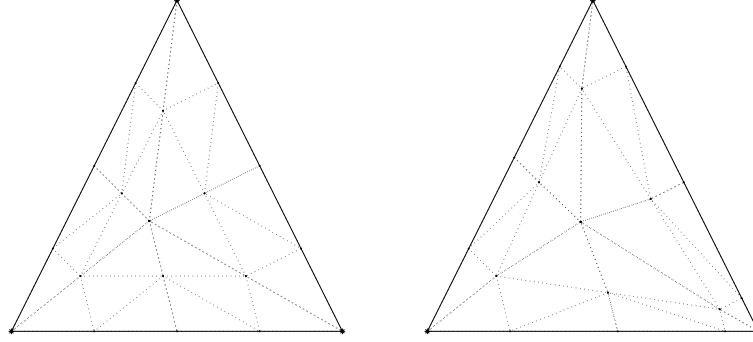


Fig. 2. Powell-Sabin tensioned finite element: projection of the control points onto the x, y plane. Right: $\lambda_i = 1$, left: $\lambda_1 = 1$, $\lambda_2 = .4$, $\lambda_3 = .8$ (vertices numbered counterclockwise from the left bottom corner).

$$s_{T|\mathcal{T}^{(p,q)}}(\mathbf{P}) = s_T^{(p,q)}(\mathbf{P}) := Z_T^{(p,q)}(\tilde{\mathbf{P}}; \lambda_1, \lambda_2, \lambda_3, f), \quad (9)$$

where $\mathcal{T}^{(p,q)}$ denotes the image of $T^{(p,q)}$ by the given transformation \mathbf{T}_T .

Since \mathbf{S}_T interpolates the data positions and the normals at the vertices, for $i = 1, 2, 3$ we have:

$$s_T(\mathbf{V}_i) = f(\mathbf{V}_i), \quad \frac{\partial s_T}{\partial x}(\mathbf{V}_i) = \frac{\partial f}{\partial x}(\mathbf{V}_i), \quad \frac{\partial s_T}{\partial y}(\mathbf{V}_i) = \frac{\partial f}{\partial y}(\mathbf{V}_i). \quad (10)$$

Remark 1. If f is a polynomial of first degree, from (4)-(7), the Bézier control points $\mathbf{B}_{i,j,k}^{(p,q)}$ belong to the plane which is the graph of f , and the same does \mathbf{S}_T due to the properties of Bézier-Bernstein representation. So, the two functions s_T and f have the same graph. That is, s_T reproduces first degree polynomials.

Remark 2. If $\lambda_1 = \lambda_2 = \lambda_3 = 1$ the projections of the Bézier control points onto the plane x, y coincide with the domain points (1), see Figure 2, left. This is no more true if the parameters λ_i take different values, see Figure 2, right. However, due to (4)-(7), for every vertex, \mathbf{V}_p , the triangles formed by the projections onto the x, y plane of Bézier control points which are direct neighbours of the vertex are simply a scaled version (with a scale factor λ_p) of those obtained in the case $\lambda_p = 1$, see Figure 2.

Now, let us consider the smoothness of s_T , see also [18]. From (3)-(7) we have that $\mathbf{S}_T \in C^1(T)$. In addition, \mathbf{T}_T is invertible, so $s_T(x, y)$ is of class C^1 on $\mathbf{T}_T(T)$. Moreover, due to the geometric properties of the Powell-Sabin refinement Δ_{PS} , the collection of the Powell-Sabin finite elements corresponding to each triangle of Δ provides an element of $\mathcal{S}_2^1(\Delta_{PS})$, that is a function in $C^1(\Omega)$, [21, 22]. So, if T, \bar{T} are two adjacent triangles of Δ sharing one edge, from the construction, patching together the corresponding parametric finite elements $\mathbf{S}_T, \mathbf{S}_{\bar{T}}$ we obtain a parametric surface of class C^1 componentwise

across the common edge. Thus, the transformations $\mathbf{T}_T, \mathbf{T}_{\bar{T}}$ are of class C^1 across the common edge. In addition, they are invertible; hence s_T and $s_{\bar{T}}$ define a function of class C^1 in $T \cup \bar{T}$. Summarizing, the function s such that

$$s|_T(\mathbf{P}) := s_T(\mathbf{P}), \quad \mathbf{P} \in T, \quad T \in \Delta,$$

is well defined, is of class C^1 and interpolates values and first derivatives of f at the vertices of the given triangulation.

3 Tensioned Powell-Sabin Quadratic B-splines

In this Section we use the results summarized in Section 2 to construct a family of compactly supported, nonnegative functions, possessing tension properties that will be used as “blending system” in the quasi-interpolation process. We build these functions by means of the parametric approach starting from suitable basis functions for the space $\mathcal{S}_2^1(\Delta_{PS})$ introduced in [9, 26]. For the sake of completeness it is useful to briefly recall from [9] the basic properties of these bases of the space $\mathcal{S}_2^1(\Delta_{PS})$.

Let us associate three functions with any vertex of Δ

$$\{\tilde{B}_l^{(j)}, \quad j = 1, 2, 3, \quad l = 1, \dots, N_V\},$$

such that $\tilde{s} = \sum_{l=1}^{N_V} \sum_{j=1}^3 c_{l,j} \tilde{B}_l^{(j)}$ for all $\tilde{s} \in \mathcal{S}_2^1(\Delta_{PS})$, and

$$\tilde{B}_l^{(j)}(x, y) \geq 0, \quad \sum_{l=1}^{N_V} \sum_{j=1}^3 \tilde{B}_l^{(j)}(x, y) = 1. \tag{11}$$

A system satisfying these properties is often called a “blending system”. The functions $\tilde{B}_l^{(j)}$ will be referred to as *Powell-Sabin B-splines*.

Let Ω_l be the subset of Ω consisting of the points belonging to the union of all the triangles of Δ containing the vertex \mathbf{V}_l and let Δ_l be the restriction of Δ to Ω_l . Any $\tilde{B}_l^{(j)}$ is required to be supported in Ω_l . Thus, $\tilde{B}_l^{(j)}$ is zero with its first derivatives at any vertex of Δ except for \mathbf{V}_l and it is uniquely determined by

$$\tilde{B}_l^{(j)}(\mathbf{V}_l) =: \alpha_l^{(j)}, \quad \frac{\partial}{\partial x} \tilde{B}_l^{(j)}(\mathbf{V}_l) =: \beta_l^{(j)}, \quad \frac{\partial}{\partial y} \tilde{B}_l^{(j)}(\mathbf{V}_l) =: \gamma_l^{(j)}.$$

Straightforward constraints have to be imposed to these values in order to satisfy (11).

Remark 3. From the Bézier-Bernstein representation, we have, [9], that $\tilde{B}_l^{(j)}$ is non negative if and only if the Bézier ordinates associated with the domain points (1) which are direct neighbours of the vertex \mathbf{V}_l are non negative.

Since the three functions $\tilde{B}_l^{(j)}$ are linearly independent the matrix

$$M_l := \begin{pmatrix} \alpha_l^{(1)} & \alpha_l^{(2)} & \alpha_l^{(3)} \\ \beta_l^{(1)} & \beta_l^{(2)} & \beta_l^{(3)} \\ \gamma_l^{(1)} & \gamma_l^{(2)} & \gamma_l^{(3)} \end{pmatrix} \quad (12)$$

is nonsingular and, due to partition of unity constraints, (11), its inverse has the following form

$$M_l^{-1} = \begin{pmatrix} 1 & d_{l,x}^{(1)} & d_{l,y}^{(1)} \\ 1 & d_{l,x}^{(2)} & d_{l,y}^{(2)} \\ 1 & d_{l,x}^{(3)} & d_{l,y}^{(3)} \end{pmatrix}, \quad d_{l,x}^{(j)}, d_{l,y}^{(j)} \in \mathbb{R}, \quad j = 1, 2, 3. \quad (13)$$

Let us consider the points

$$\mathbf{Q}_l^{(j)} := \mathbf{V}_l + \mathbf{d}_l^{(j)}, \quad \mathbf{d}_l^{(j)} := (d_{l,x}^{(j)}, d_{l,y}^{(j)})', \quad j = 1, 2, 3. \quad (14)$$

These points uniquely determine values and gradients of the three functions $\tilde{B}_l^{(j)}$, $j = 1, 2, 3$, at \mathbf{V}_l and possess various interesting properties:

- i) since M_l is non singular, the points $\mathbf{Q}_l^{(j)}$, $j = 1, 2, 3$, are not collinear and, from (12)-(13), $(\alpha_l^{(1)}, \alpha_l^{(2)}, \alpha_l^{(3)})$ are the barycentric coordinates of \mathbf{V}_l with respect to the triangle they form;
- ii) the functions $\tilde{B}_l^{(j)}$, $j = 1, 2, 3$, are non negative if and only if the triangle with vertices $\mathbf{Q}_l^{(j)}$, $j = 1, 2, 3$, contains the domain points (1) which are direct neighbours of \mathbf{V}_l , [9, Section 4] (see also Figure 3, top-left);
- iii) the points $\mathbf{Q}_l^{(j)}$ are *Greville* points, [9, 10, 20], that is

$$p(\cdot) = \sum_{l=1}^{N_V} \sum_{j=1}^3 p(\mathbf{Q}_l^{(j)}) \tilde{B}_l^{(j)}(\cdot), \quad \forall p \in \mathbb{P}_1,$$

so that the triangle they form will be referred to as Greville triangle;

- iv) the B-spline basis has better properties from the computational and the approximation point of view if $\mathbf{Q}_l^{(j)}$ are as close as possible (considering positivity constraints, see property ii)) to \mathbf{V}_l , [20, 26].

Summarizing, the points $\mathbf{Q}_l^{(j)}$, $j = 1, 2, 3$ – and so the triangle they form – are uniquely associated with the triple $\tilde{B}_l^{(j)}$, $j = 1, 2, 3$, and can be efficiently used to identify and describe these functions and their properties instead of $\alpha_l^{(j)}, \beta_l^{(j)}, \gamma_l^{(j)}$. To obtain a “good” B-spline basis of $\mathcal{S}_2^1(\Delta_{PS})$ it suffices to determine for every vertex \mathbf{V}_l , a Greville triangle with small area containing the domain points which are direct neighbours of the vertex.

In the following we will denote $\tilde{B}_l^{(j)}$ by $\tilde{B}_l^{(j)}(\cdot; \mathbf{d}_l^{(1)}, \mathbf{d}_l^{(2)}, \mathbf{d}_l^{(3)})$ whenever we need to emphasize the dependence of the Powell-Sabin B-splines on the points $\mathbf{Q}_l^{(j)}$ (that is on the vectors $\mathbf{d}_l^{(j)}$), $j = 1, 2, 3$.

Now, we are able to describe how to equip a Powell-Sabin B-spline basis with tension parameters. The parametric approach described in Section 2 will be used to this purpose. Let us associate a parameter $\lambda_l \in (0, 1]$ with each vertex of Δ and let us denote

$$\begin{aligned} \Lambda &:= \{\lambda_l, l = 1, \dots, N_V\}, \\ \Lambda_l &:= \{\lambda_j, j \in \mathcal{I}_l\}, \text{ where } \mathcal{I}_l := \{j : \mathbf{V}_j \in \Omega_l\}. \end{aligned}$$

For every vertex \mathbf{V}_l let us consider a triple of vectors $(\mathbf{d}_l^{(1)}, \mathbf{d}_l^{(2)}, \mathbf{d}_l^{(3)})$, and the corresponding Greville triangle and determine the triples $(\alpha_l^{(j)}, \beta_l^{(j)}, \gamma_l^{(j)})$ according to (12)-(13). Let us denote by $B_l^{(j)}(\cdot; \Lambda_l, \mathbf{d}_l^{(1)}, \mathbf{d}_l^{(2)}, \mathbf{d}_l^{(3)})$ (or simply $B_l^{(j)}(\cdot; \Lambda_l)$) the function locally constructed in every triangle of Δ according to (3)-(7) and (9) setting

$$f(\mathbf{V}_l) = \alpha_l^{(j)}, \nabla' f(\mathbf{V}_l) = \begin{pmatrix} \beta_l^{(j)} \\ \gamma_l^{(j)} \end{pmatrix}, \quad f(\mathbf{V}_k) = 0, \nabla' f(\mathbf{V}_k) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \text{ if } k \neq l.$$

From the results of Section 2 and from the properties of Powell-Sabin B-splines we have that $B_l^{(j)}(\cdot; \Lambda_l) \in C^1(\Omega)$, its support is contained in Ω_l , and, from (10)

$$\begin{aligned} B_l^{(j)}(\mathbf{V}_l; \Lambda_l, \mathbf{d}_l^{(1)}, \mathbf{d}_l^{(2)}, \mathbf{d}_l^{(3)}) &= \alpha_l^{(j)}, \\ \frac{\partial}{\partial x} B_l^{(j)}(\mathbf{V}_l; \Lambda_l, \mathbf{d}_l^{(1)}, \mathbf{d}_l^{(2)}, \mathbf{d}_l^{(3)}) &= \beta_l^{(j)}, \\ \frac{\partial}{\partial y} B_l^{(j)}(\mathbf{V}_l; \Lambda_l, \mathbf{d}_l^{(1)}, \mathbf{d}_l^{(2)}, \mathbf{d}_l^{(3)}) &= \gamma_l^{(j)}. \end{aligned} \quad (15)$$

In each triangle of Δ the third component of the Bézier control points in (3) defining $B_l^{(j)}(\cdot; \Lambda_l, \mathbf{d}_l^{(1)}, \mathbf{d}_l^{(2)}, \mathbf{d}_l^{(3)})$ coincides with the values of the Bézier ordinates for the Powell-Sabin B-spline determined by the triple $(\alpha_l^{(j)}, \lambda_l \beta_l^{(j)}, \lambda_l \gamma_l^{(j)})$, that is $\tilde{B}_l^{(j)}(\cdot; \lambda_l^{-1} \mathbf{d}_l^{(1)}, \lambda_l^{-1} \mathbf{d}_l^{(2)}, \lambda_l^{-1} \mathbf{d}_l^{(3)})$ (see (12)-(13)). Thus, from Remarks 2 and 3 and from property ii) we have (see Figure 3, top)

Theorem 1. $B_l^{(j)}(\cdot; \Lambda_l, \mathbf{d}_l^{(1)}, \mathbf{d}_l^{(2)}, \mathbf{d}_l^{(3)})$, $j = 1, 2, 3$, are non negative if and only if the corresponding Greville triangle contains the projections onto the x , y plane of the Bézier control points which are direct neighbours of \mathbf{V}_l . \square

Remark 4. Note that as the parameters λ_l approach 0 the projections onto the x , y plane of the Bézier control points $\mathbf{B}_{1,j,k}^{(p,0)}$ and $\mathbf{B}_{i,1,k}^{(p,1)}$ approach the vertices of Δ , see (4)-(7) and Figure 3, top.

Of course, we have $B_l^{(j)}(\cdot; \Lambda_l, \mathbf{d}_l^{(1)}, \mathbf{d}_l^{(2)}, \mathbf{d}_l^{(3)}) = \tilde{B}_l^{(j)}(\cdot; \mathbf{d}_l^{(1)}, \mathbf{d}_l^{(2)}, \mathbf{d}_l^{(3)})$ if $\lambda_i = 1$, $i \in \mathcal{I}_l$. Let us now briefly analyze the behaviour of the functions $B_l^{(j)}(\cdot; \Lambda_l, \mathbf{d}_l^{(1)}, \mathbf{d}_l^{(2)}, \mathbf{d}_l^{(3)})$ as λ_i , $i \in \mathcal{I}_l$, approach zero. If the triple $(\alpha_l^{(j)}, \beta_l^{(j)}, \gamma_l^{(j)})$, (that is the Greville triangle associated with \mathbf{V}_l) does not

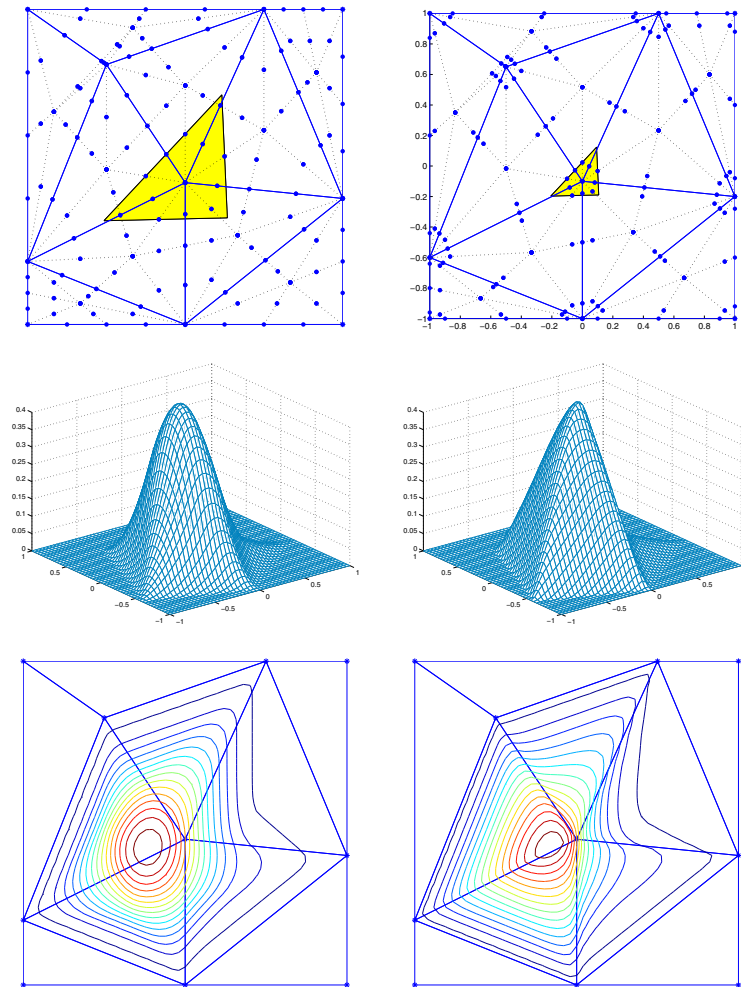


Fig. 3. One of the three B-splines related to a vertex. Left $\lambda_i = 1$; right $\lambda_i = .4$. Top to bottom: Powell-Sabin refinement and Greville triangle associated with the vertex (dots show the projections of the Bézier control points and dotted lines denote the edges of the Powell-Sabin refinement), B-spline, level sets.

change as λ_l decreases, from Section 2, $B_l^{(j)}(.; A_l, \mathbf{d}_l^{(1)}, \mathbf{d}_l^{(2)}, \mathbf{d}_l^{(3)})$ approaches the pyramid taking the value $\alpha_l^{(j)}$ at \mathbf{V}_l . This is no more the case if, while ensuring positivity, the Greville points associated with \mathbf{V}_l approach the vertex as $O(\lambda_l)$ that is the maximal rate consistent with positivity constraints, (see Remark 4) because in such a case the values of the partial derivatives obtained from (12)-(13) are not bounded. However, in any case, since it is equal to one, the sum of the three functions $B_l^{(j)}(.; A_l, \mathbf{d}_l^{(1)}, \mathbf{d}_l^{(2)}, \mathbf{d}_l^{(3)})$, $j = 1, 2, 3$ has zero partial derivatives at \mathbf{V}_l so that it approaches the pyramid with summit $(\mathbf{V}_l, 1)$ supported in Ω_l as λ_i , $i \in \mathcal{I}_l$, approach zero, see Figure 4. This property will be important in the analysis of the behavior of the q.i.s based on the functions $B_l^{(j)}(.; A_l, \mathbf{d}_l^{(1)}, \mathbf{d}_l^{(2)}, \mathbf{d}_l^{(3)})$, $j = 1, 2, 3$, $l = 1, \dots, N_V$, which will be discussed in the next Section.

Summarizing, the parameters λ_i act as tension parameters on the graph of the functions $B_l^{(j)}(.; A_l)$, $j = 1, 2, 3$, $l = 1, \dots, N_V$. So, we will refer to these functions as *tensioned Powell-Sabin B-splines* and we denote by $\mathcal{S}_2^1(\Delta_{PS}; A)$ the linear space they span.

We emphasize that the parameters λ_i have a completely local effect and they can assume different values at different vertices according to the tension effect we want to reach in the corresponding functions, see Figure 4, bottom.

Remark 5. It is worth to note that the space $\mathcal{S}_2^1(\Delta_{PS}; A)$ depends on the tension parameters, A , but is independent of the choice of the vectors $\mathbf{d}_l^{(j)}$, $j = 1, 2, 3$, $l = 1, \dots, N_V$. Once the tension parameters have been fixed, different choices of the sequence of these vectors determine different bases of the same space (see (12), (13) and (15)). Of course, these bases present different performances from a computational point of view. For the “non-tensioned” case ($\lambda_l = 1$, $l = 1, \dots, N_V$), the results presented in [9, 17] and [20] show that bases corresponding to “small” Greville triangles are preferable. This remains true in the tensioned case.

4 Discrete Quasi-Interpolants with Tension Properties

In this Section we construct q.i.s in the space $\mathcal{S}_2^1(\Delta_{PS}; A)$ based on values of a (given) function f without requiring information on its derivatives (discrete q.i.s). So, we consider q.i.s of the following form

$$\begin{aligned} \mathcal{Q}f(.; A) &:= \sum_{l=1}^{N_V} \sum_{j=1}^3 \mu_{l,A_l}^{(j)}(f) B_l^{(j)}(.; A_l, \mathbf{d}_l^{(1)}, \mathbf{d}_l^{(2)}, \mathbf{d}_l^{(3)}), & (16) \\ \mu_{l,A_l}^{(j)}(f) &:= \sum_{k=1}^{N_l^{(j)}} q_{l,A_l}^{(j,k)} f(\mathbf{Z}_{l,A_l}^{(j,k)}), \quad q_{l,A_l}^{(j,k)} \neq 0, \mathbf{Z}_{l,A_l}^{(j,k)} \in \mathbb{R}^2, N_l^{(j)} \in \mathbb{N}. \end{aligned}$$

First we note that the points $\mathbf{Q}_l^{(j)}$ are Greville points even in the “tensioned” case, in fact we have

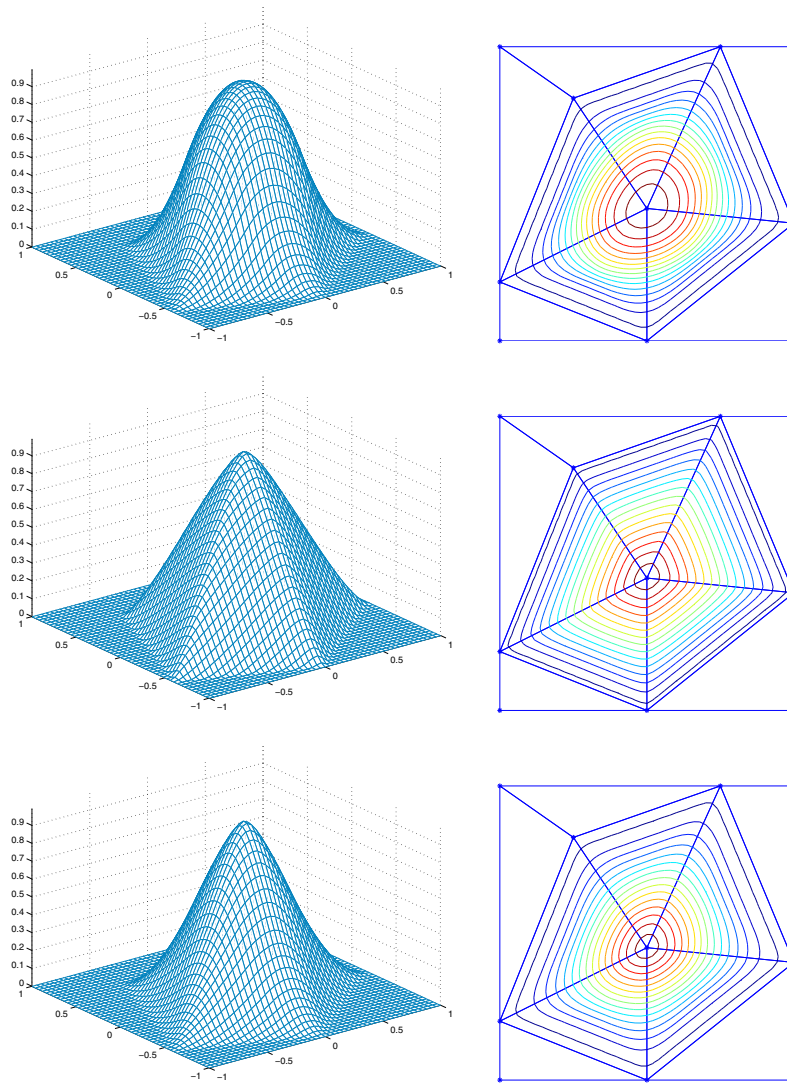


Fig. 4. Sum of the three B-splines related to a vertex and their level sets. Top $\lambda_i = 1$; center $\lambda_i = .4$; bottom $\lambda_i = 1$, except for the central vertex where $\lambda_i = .4$.

Theorem 2. For any set of tension parameters, Λ , let us put

$$\mathcal{Q}f(\cdot; \Lambda) := \sum_{l=1}^{N_V} \sum_{j=1}^3 f(\mathbf{Q}_l^{(j)}) B_l^{(j)}(\cdot; \Lambda_l, \mathbf{d}_l^{(1)}, \mathbf{d}_l^{(2)}, \mathbf{d}_l^{(3)}) \quad (17)$$

then $\mathcal{Q}p(\cdot; \Lambda) = p, \forall p \in \mathbb{P}_1$.

Proof. From (15) and from (12)-(14) we have that $\mathcal{Q}f$ takes the same values and has the same first derivatives as f at the vertices of Δ if f is a polynomial of first degree (see also property iii) in Section 3). Then the assertion follows from Remark 1. \square

If $\lambda_l \neq 1$ for some l , the space of quadratic polynomials is not contained in $\mathcal{S}_2^1(\Delta_{PS}; \Lambda)$ so, in general, it does not make sense to ask for reproduction of polynomials of degree greater than 1 for q.i.s in this space. However, it is possible to provide explicit expressions of $\mu_{l, \Lambda_l}^{(j)}(f)$ in order that the corresponding q.i. be a projector, i.e. reproduces any element of the space.

As an example, for any set of tension parameters, Λ , and for any set of vectors $\{\mathbf{d}_l^{(j)}, j = 1, 2, 3, l = 1, \dots, N_V\}$, let $T_l^{(k)}, k = 1, 2$, be triangles of Δ_l and let $\tilde{\mathbf{W}}_l^{(k)} := (\tilde{W}_{l,x}^{(k)}, \tilde{W}_{l,y}^{(k)})'$ be in $T_l^{(k)}$; $\mathbf{V}_l, \tilde{\mathbf{W}}_l^{(1)}, \tilde{\mathbf{W}}_l^{(2)}$ not collinear. Let us put

$$\tilde{\mathbf{U}}_l^{(k)} := \nu_{k,l} \mathbf{V}_l + (1 - \nu_{k,l}) \tilde{\mathbf{W}}_l^{(k)}, \quad \nu_{k,l} \in (0, 1). \quad (18)$$

Let us denote by $p_{l,k,r}, r = 1, 2, 3$ the indices of the vertices of $T_l^{(k)}$, and

$$\mathbf{P}_l^{(k)} := \mathbf{T}_{T_l^{(k)}}(\tilde{\mathbf{P}}_l^{(k)}; \lambda_{p_{l,k,1}}, \lambda_{p_{l,k,2}}, \lambda_{p_{l,k,3}}), \quad \mathbf{P} = \mathbf{W}, \mathbf{U}, \quad (19)$$

where $\mathbf{T}_{T_l^{(k)}}$ is the transformation defined in (8). Let us consider the family of q.i.s (16) where

$$\mu_{l, \Lambda_l}^{(j)}(f) := f(\mathbf{V}_l) + \zeta_l^{(j,1)} D_l^{(1)} + \zeta_l^{(j,2)} D_l^{(2)}, \quad j = 1, 2, 3, \quad (20)$$

$$D_l^{(k)} := \frac{f(\mathbf{U}_l^{(k)}) + \nu_{l,k}(\nu_{l,k} - 2)f(\mathbf{V}_l) - (1 - \nu_{l,k})^2 f(\mathbf{W}_l^{(k)})}{\nu_{k,l}(1 - \nu_{k,l})}, \quad (21)$$

and the scalars $\zeta_l^{(j,1)}, \zeta_l^{(j,2)}$ are so that

$$\lambda_l^{-1} \mathbf{d}_l^{(j)} = \zeta_l^{(j,1)} (\tilde{\mathbf{W}}_l^{(1)} - \mathbf{V}_l) + \zeta_l^{(j,2)} (\tilde{\mathbf{W}}_l^{(2)} - \mathbf{V}_l), \quad j = 1, 2, 3. \quad (22)$$

Theorem 3. Let \mathcal{Q} be any q.i. of the form (16) with $\mu_{l, \Lambda_l}^{(j)}$ defined according to (18)-(22). If the points of each triple $\mathbf{V}_l, \tilde{\mathbf{W}}_l^{(k)}, \tilde{\mathbf{U}}_l^{(k)}, k = 1, 2$, belong to the same subtriangle of the Powell-Sabin refinement of Δ , then

$$\mathcal{Q}s(\cdot; \Lambda) = s, \quad \forall s \in \mathcal{S}_2^1(\Delta_{PS}; \Lambda).$$

Proof. Let us consider any element of the space $\mathcal{S}_2^1(\Delta_{PS}; \Lambda)$ in its parametric form. For any triangle, T , of Δ , the x and y components, $X_T^{(p,q)}$, $Y_T^{(p,q)}$, only depend on the triangulation Δ , on its Powell-Sabin refinement Δ_{PS} and on the set of tension parameters Λ , hence they are the same for all the elements of the space, see (4)-(7). Thus, two elements in $\mathcal{S}_2^1(\Delta_{PS}; \Lambda)$ coincide if their z components, $Z_T^{(p,q)}$, are the same. The z component, as a function of $\tilde{\mathbf{P}}$, see (3), belongs to the space of quadratic Powell-Sabin splines spanned by the family of Powell-Sabin B-splines $\tilde{B}_l^{(j)}(\cdot; \lambda_l^{-1} \mathbf{d}_l^{(1)}, \lambda_l^{-1} \mathbf{d}_l^{(2)}, \lambda_l^{-1} \mathbf{d}_l^{(3)})$, see Section 3. In addition, for $\mathbf{P} = \mathbf{U}, \mathbf{W}$, from (9)

$$\tilde{B}_l^{(j)}(\tilde{\mathbf{P}}_l^{(k)}; \lambda_l^{-1} \mathbf{d}_l^{(1)}, \lambda_l^{-1} \mathbf{d}_l^{(2)}, \lambda_l^{-1} \mathbf{d}_l^{(3)}) = B_l^{(j)}(\mathbf{P}_l^{(k)}; A_l, \mathbf{d}_l^{(1)}, \mathbf{d}_l^{(2)}, \mathbf{d}_l^{(3)}).$$

Thus, the assert follows because, from [20, Theorem 10], the z component of (16) defines a q.i. in $\mathcal{S}_2^1(\Delta_{PS})$ which reproduces any element of the space. \square

Since the space $\mathcal{S}_2^1(\Delta_{PS}; \Lambda)$ contains only polynomials of first degree for general values of the tension parameters, we have that the proposed q.i.s are in general only second order accurate. This is a common feature of approximating schemes based on tension methods. However, as expected, if the tension parameters λ_p are close to 1 a better approximation behaviour can be reached. As an example, for the q.i.s we have introduced before we have

Theorem 4. *Let \mathcal{Q} be any q.i. of the form (16) with $\mu_{l, A_l}^{(j)}$ defined according to (18)-(22) and let f be a given function of class $C^3(\Omega)$. Let h denote the maximum length of an edge of Δ . If, for some constant K_1*

$$0 \leq 1 - \lambda_l \leq K_1 h^2, \quad l = 1, \dots, N_V, \tag{23}$$

then, there exists a constant K such that

$$\|\mathcal{Q}f(\cdot; \Lambda) - f\| \leq Kh^3.$$

Proof. From (13) and (22) it follows that for any $l = 1, \dots, N_V$

$$\sum_{j=1}^3 \alpha_l^{(j)} \zeta_l^{(j,k)} = 0, \quad k = 1, 2,$$

so that $\mathcal{Q}f(\mathbf{V}_l; \Lambda) = f(\mathbf{V}_l)$. In addition, from (13) and (22),

$$\left(\begin{array}{cc} \sum_{j=1}^3 \beta_l^{(j)} \zeta_l^{(j,1)} & \sum_{j=1}^3 \beta_l^{(j)} \zeta_l^{(j,2)} \\ \sum_{j=1}^3 \gamma_l^{(j)} \zeta_l^{(j,1)} & \sum_{j=1}^3 \gamma_l^{(j)} \zeta_l^{(j,2)} \end{array} \right) = \lambda_l^{-1} \left(\begin{array}{cc} \tilde{W}_{l,x}^{(1)} - V_{l,x} & \tilde{W}_{l,y}^{(1)} - V_{l,y} \\ \tilde{W}_{l,x}^{(2)} - V_{l,x} & \tilde{W}_{l,y}^{(2)} - V_{l,y} \end{array} \right)^{-1}.$$

Moreover, from (2) and (3)-(7),

$$\tilde{\mathbf{P}} = \mathbf{T}_{T_l^{(k)}}(\tilde{\mathbf{P}}; 1, 1, 1), \quad \mathbf{P} = \mathbf{W}, \mathbf{U}, \tag{24}$$

so, from (19), there exists a constant K_2 such that

$$\|\mathbf{U}_l^{(k)} - \tilde{\mathbf{U}}_l^{(k)}\|, \|\mathbf{W}_l^{(k)} - \tilde{\mathbf{W}}_l^{(k)}\| \leq K_2 \max_{1 \leq p \leq N_V} (1 - \lambda_p)h.$$

Thus, setting

$$\tilde{D}_l^{(k)} := \frac{f(\tilde{\mathbf{U}}_l^{(k)}) + \nu_{l,k}(\nu_{l,k} - 2)f(\mathbf{V}_l) - (1 - \nu_{l,k})^2 f(\tilde{\mathbf{W}}_l^{(k)})}{\nu_{k,l}(1 - \nu_{k,l})}, \quad (25)$$

there exists a constant K_3 such that

$$|D_l^{(k)} - \tilde{D}_l^{(k)}| \leq K_3 \max_{1 \leq p \leq N_V} (1 - \lambda_p)h.$$

Finally, from (25)

$$\begin{pmatrix} \tilde{W}_{l,x}^{(1)} - V_{l,x} & \tilde{W}_{l,y}^{(1)} - V_{l,y} \\ \tilde{W}_{l,x}^{(2)} - V_{l,x} & \tilde{W}_{l,y}^{(2)} - V_{l,y} \end{pmatrix}^{-1} \begin{pmatrix} \tilde{D}_l^{(1)} \\ \tilde{D}_l^{(2)} \end{pmatrix} = \nabla' f(\mathbf{V}_l), \text{ if } f \in \mathbb{P}_2.$$

Hence, if $f \in C^3(\Omega)$ and (23) holds, we have that $\nabla \mathcal{Q}f(\mathbf{V}_l; \Delta)$ provides a second order accurate estimate of $\nabla f(\mathbf{V}_l)$. Then the assert follows from a simple generalization of Theorem 3.2 in [19]. For the sake of completeness we note that the constant K depends on the third derivatives of f , on the geometric characteristics of Δ , on K_1 and on the choice of $\tilde{\mathbf{W}}_l^{(k)}$ and $\tilde{\mathbf{U}}_l^{(k)}$. \square

We end this Section discussing the behaviour of the presented q.i.s as the tension parameters approach 0. In this connection it is important to recall that, as the tension parameters tend to 0, the function $\sum_{j=1}^3 B_l^{(j)}$ approaches the pyramid with summit $(\mathbf{V}_l, 1)$ supported in Ω_l (see Section 3). Assuming that

$$\|\mathbf{d}_l^{(j)}\| = \|\mathbf{V}_l - \mathbf{Q}_l^{(j)}\| = O(\lambda_l), \quad j = 1, 2, 3, \quad l = 1, \dots, N_V, \quad (26)$$

for a continuous function f , $f(\mathbf{Q}_l^{(j)})$ approaches $f(\mathbf{V}_l)$ as λ_l tends to 0. So, the q.i. (17) approaches the piecewise linear interpolating f at the vertices of Δ .

To analyze the behaviour of the q.i. defined by (20), in addition to (26) we assume that

$$\|\mathbf{V}_l - \tilde{\mathbf{W}}_l^{(k)}\| = O(\lambda_l), \quad k = 1, 2, \quad l = 1, \dots, N_V,$$

so that, from (3)-(7), (18), (19) and (24)

$$\|\mathbf{V}_l - \mathbf{W}_l^{(k)}\| = O(\lambda_l^2), \quad \|\mathbf{V}_l - \mathbf{U}_l^{(k)}\| = O(\lambda_l^2), \quad k = 1, 2, \quad l = 1, \dots, N_V.$$

Thus, from (21), $|D_l^{(k)}| = O(\lambda_l^2)$, $k = 1, 2$, for any smooth function f . Hence, from (20) and (22), $\mu_{l,\Delta_l}^{(j)}(f)$ approaches $f(\mathbf{V}_l)$ as λ_l tends to 0. As a consequence, the q.i. defined by (20) approaches the piecewise linear function interpolating f at the vertices of Δ as the tension parameters tend to 0.

Remark 6. The q.i. defined by (20) is particularly attractive because it is a projection. Nevertheless, other interesting q.i.s in the space $\mathcal{S}_2^1(\Delta_{PS}; \Lambda)$ can be obtained generalizing to the “tensioned” case the q.i.s proposed in [20]. The resulting q.i.s have an asymptotic behaviour similar to that one of q.i. defined by (20) as the tension parameters tend to 0.

5 Numerical Examples

In this Section we illustrate the numerical performances of the q.i.s presented above by means of some graphical and numerical examples.

In the first two examples we have considered data taken from the function

$$f(x, y) = \max(0, \mathbf{peaks}(4(x - 0.4), 4(y - 0.4))) \quad (27)$$

at the vertices of a nonuniform triangulation of the unit square, see Figure 5, left. Here \mathbf{peaks} denotes the corresponding function of MATLAB. The graph of (27) is depicted in Figure 5, right. The used triangulation Δ , see Figure 5 left, has been selected on purpose, taking into account the shape of the given function.

In the examples we present different families of Greville triangles, that is of vectors $\mathbf{d}_l^{(j)}$, see (14). So, we deal with different bases of the space $\mathcal{S}_2^1(\Delta_{PS}; \Lambda)$, see Remark 5. In any case, these bases have been constructed considering Greville triangles as “small” as possible in agreement with the positivity constraints given in Theorem 1. The used construction is an extension of that one used in the non-tensioned case, for further details see [20].

In the first Example (see Figure 6) the q.i. (17) is presented. The first column of the Figure shows the given triangulation Δ , its Powell-Sabin refinement and the Greville triangles in the non-tensioned case (top) and if a uniform tension, $\lambda_l = .6$, is applied at every vertex (bottom). Reducing the values of the tension parameters induces a reduction of the size of the Greville triangles, see (26). The second column depicts the graph of the q.i. (17) corresponding to the two sets of tension parameters and to the considered bases.

In the second example (Figures 7-8) we present the q.i. defined by (20) which is a projection in $\mathcal{S}_2^1(\Delta_{PS}; \Lambda)$. In addition, a different family of Greville triangles, that is a different B-spline basis, has been considered (Figure 7, left). Even without any tension effect (Figure 7, top) the q.i. shows a significant graphical improvement with respect to q.i. (17). Due to the local influence of the parameters λ_l , the tension effect can be applied, selectively, only in particular regions of Ω . To illustrate this we have reduced the value of the parameters λ_l from 1 to .3 only for the six circled vertices of Δ depicted in Figure 5, left. The resulting q.i., (Figure 7, bottom-right) presents a “fair” aspect and a consistent agreement with the shape of the given function. Figure 8 shows the level sets of the (locally) tensioned q.i. (left) and of the given function (right).

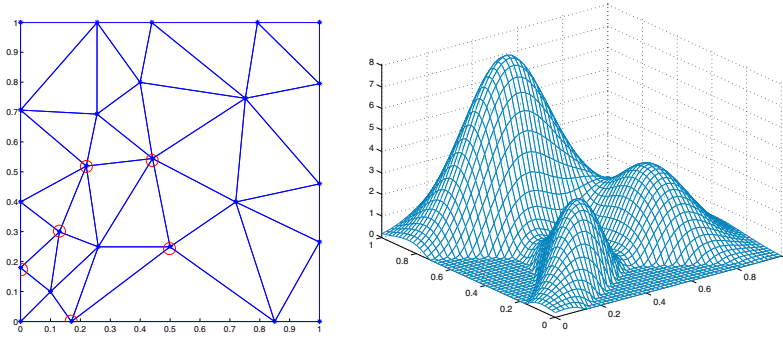


Fig. 5. Examples 1 and 2: triangulation and given function.

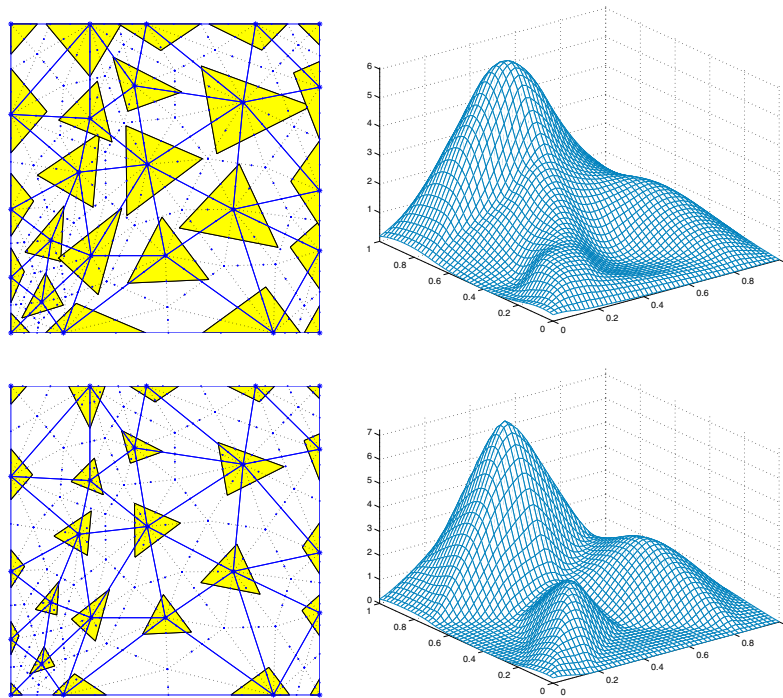


Fig. 6. Example 1: the q.i. (17). Greville triangles and the obtained q.i.s. Top: no tension $\lambda_l = 1, l = 1, \dots, N_V$; Bottom: $\lambda_l = .6, l = 1, \dots, N_V$.

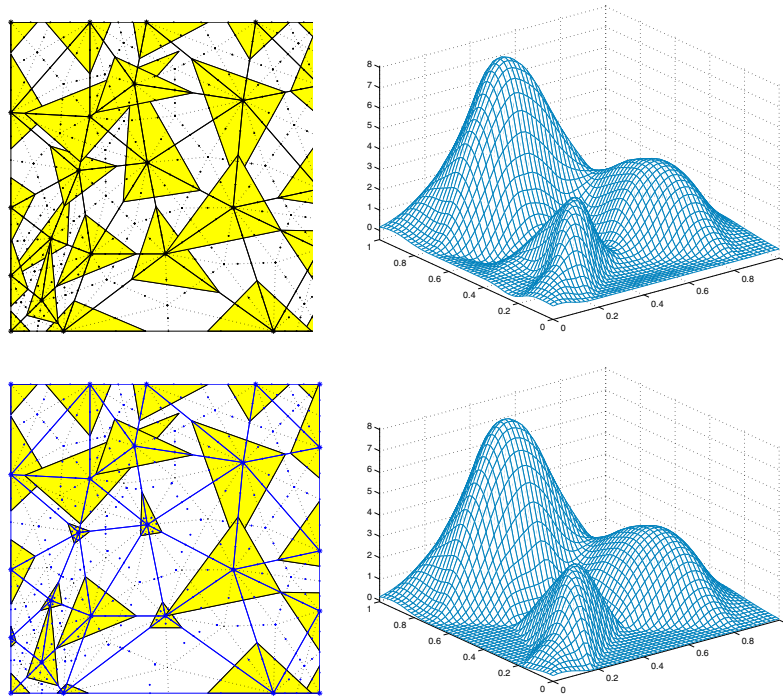


Fig. 7. Example 2: the q.i. defined by (20). Greville triangles and built q.i.s. Top: no tension. Bottom: $\lambda_i = 1$ everywhere except $\lambda_i = .3$ at the six circled vertices depicted in Figure 5 left.

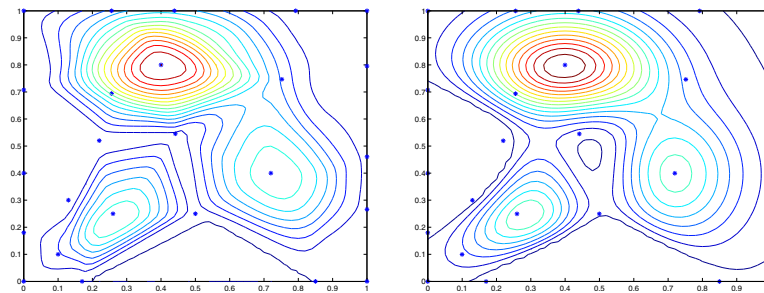


Fig. 8. Example 2.: level sets of the second q.i. in Figure 7 (left), and of the given function (right) (* denote the vertices of Δ).

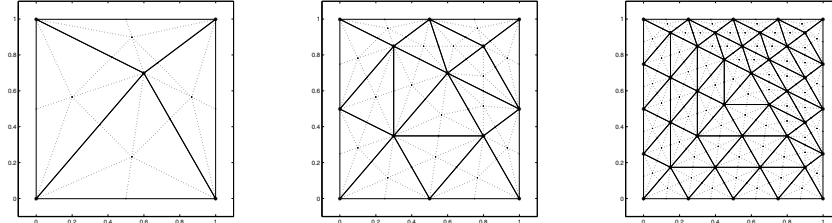


Fig. 9. Triangulations $\Delta^{(k)}$, $k = 0, 1, 2$ and their Powell-Sabin refinements.

Finally, to numerically confirm the approximation power of the proposed q.i.s, we have considered the triangulation $\Delta^{(0)}$ depicted in Figure 9 (left) and the refined triangulations $\Delta^{(k)}$ (see Figure 9, for $k = 0, 1, 2$) obtained considering the midpoint of any edge of $\Delta^{(k-1)}$ and taking the Delaunay triangulation of this new set of vertices. We have applied the q.i. defined by (20) to the function $p(x, y) = x^3 + y^3 - x^2y - xy^2$ over the partitions $\Delta^{(k)}$, $k = 0, 1, 2, 3$. Denoting by x_r, y_s equally spaced points in $[0, 1]$, we have computed in each case

$$\max_{r,s=1,\dots,50} |p(x_r, y_s) - \mathcal{Q}f(x_r, y_s)|. \quad (28)$$

The results are depicted in Table 1. Any row of the table refers to a triangulation $\Delta^{(k)}$ for fixed k . The first column indicates the refinement level while the second one shows the maximum length of an edge of the triangulation. The remaining columns show the values of the tabulated absolute error (28) for the q.i. defined by (20) for different values of the tension parameters. In the third and fourth column the error decreases as the third power of h , according to Theorem 4, while in the last column we have just a second order accuracy since the tension parameters do not satisfy (23).

Table 1. Tabulated error (28) for the q.i. defined by (20)

k	h	$\lambda_l = 1 - \frac{h^2}{2}$	$\lambda_l = 1$	$\lambda_l = \frac{1}{2}$
0	1	.17612	.04346	.17612
1	.5	.01754	.00657	.06819
2	.25	.00157	.00080	.02070
3	.125	.00015	.00010	.00542

6 Conclusion

We have described the construction and discussed some properties of two families of q.i.s, based on an extension of quadratic Powell-Sabin splines, which do not require derivatives in input.

The q.i.s of the second family reproduce any element of the space they belong to and they are third order accurate if the tension parameters are not too small, even if reproduction of quadratic polynomials can not be achieved.

The obtained approximating functions can be seen as particular parametric surfaces with piecewise quadratic components and possess shape parameters which act as tension parameters. The shape parameters easily allow to control the shape of the q.i.s avoiding oscillations and inflections extraneous to the behaviour of the data.

We end the paper noting that, for an efficient application of the proposed q.i.s, as in all approximating schemes based on tension methods, a crucial point is the practical choice of the value of the tension parameters. For the sake of brevity we can not discuss here this important aspect in detail. However, we emphasize that the practical choice of the tension parameters is greatly simplified when they possess a clear geometric meaning. The parametric approach we have used in this paper to construct q.i.s with tension properties is based on shape parameters having an evident geometric interpretation (amplitude of the tangent vectors at the data points with respect to the considered parameterization). Moreover, the Bézier-Bernstein representation used for the q.i.s strengthens this geometric interpretation in the sense that constraints on the shape of the q.i. can be easily translated in (sufficient) constraints on the Bézier control points which can be manipulated in a much easier way. For a more detailed discussion on this point, see [1, 6] and references quoted therein.

References

1. S. Asaturyan, P. Costantini, and C. Manni: Local shape-preserving interpolation by space curves. *IMA J. Numer. Anal.* **21**, 2001, 301–325.
2. C. de Boor and G. Fix: Spline approximation by quasi-interpolants. *J. Approx. Theory* **8**, 1973, 19–45.
3. C. de Boor: Quasi-interpolants and approximation power of multivariate splines. In: *Computation of Curves and Surfaces*, W. Dahmen, M. Gasca and C.A. Micchelli (eds.), Kluwer, Dordrecht, 1990, 313–345.
4. C. de Boor, K. Höllig, and S. Riemenschneider: *Box-splines*, Springer, New York, 1993.
5. C.K. Chui: *Multivariate splines*, SIAM, Philadelphia, 1988.
6. P. Costantini and C. Manni: Geometric construction of spline curves with tension properties. *Aided Geom. Design* **20**, 2003, 579–599.
7. W. Dahmen and C.A. Micchelli: Recent progress in multivariate splines. In: *Approximation Theory IV*, C.K. Chui et al. (eds.), Academic Press, 1983, 27–121.

8. O. Davydov and F. Zeilfelder: Scattered data fitting by direct extension of local polynomials to bivariate splines. *Advances in Comp. Math.* **21**, 2004, 223–271.
9. P. Dierckx: On calculating normalized Powell-Sabin B -splines. *Comput. Aided Geom. Design* **15**, 1997, 61–78.
10. T.N.E. Greville: On the normalization of the B -splines and the location of nodes for the case of unequally spaced knots. In: *Inequalities*, O. Shisha (ed.), Academic Press, New York, 1967, 286–290.
11. J. Hoschek and D. Lasser: *Fundamentals of Computer Aided Geometric Design*. A. K. Peters, Wellesley, Massachusetts, 1993.
12. M. Laghchim-Lahlou and P. Sablonnière: C^r -finite elements of Powell-Sabin type on the three direction mesh. *Advances in Comp. Math.* **6**, 1996, 191–206.
13. M.-J. Lai and L.L. Schumaker: Macro-elements and stable local bases for splines on Powell-Sabin triangulations. *Math. Comp.* **72**, 2003, 335–354.
14. P. Lamberti and C. Manni: Tensioned quasi-interpolation via geometric continuity. *Advances in Comp. Math.* **20**, 2004, 105–127.
15. B.G. Lee, T. Lyche, and L.L. Schumaker: Some examples of quasi-interpolants constructed from local spline projectors. In: *Mathematical Methods for Curves and Surfaces: Oslo 2000*, T. Lyche and L.L. Schumaker (eds.), Vanderbilt University Press, Nashville, 2001, 243–252.
16. T. Lyche and L.L. Schumaker: Local spline approximation methods. *J. Approximation Theory* **15**, 1975, 294–325.
17. J. Maes, E. Vanraes, P. Dierckx, and A. Bultheel: On the stability of normalized Powell-Sabin B -splines. *J. Comput. Applied Math.* **170**, 2004, 181–196.
18. C. Manni: A general parametric framework for functional tension schemes. *J. Comput. Applied Math.* **119**, 2000, 275–300.
19. C. Manni: Local tension methods for bivariate scattered data interpolation. In: *Mathematical Methods for Curves and Surfaces: Oslo 2000*, T. Lyche and L.L. Schumaker (eds.), Vanderbilt University Press, Nashville, 2001, 293–314.
20. C. Manni and P. Sablonnière: Quadratic spline quasi-interpolants on Powell-Sabin partitions. *Advances in Comp. Math.* to appear.
21. M.J.D. Powell and M.A. Sabin: Piecewise quadratic approximations on triangles. *ACM Trans. Math. Software* **3**, 1977, 316–325.
22. P. Sablonnière: Error bounds for Hermite interpolation by quadratic splines on an α -triangulation. *IMA J. Numer. Anal.* **7**, 1987, 495–508.
23. P. Sablonnière: Recent progress on univariate and multivariate polynomial or spline quasi-interpolants. In: *Trends and Applications in Constructive Approximation*, M.G. de Bruijn, D.H. Mache and J. Szabados (eds.), Birkhäuser, Basel, 2005, 229–245.
24. I.J. Schoenberg: Contribution to the problem of approximation of equidistant data by analytic functions. Part A. On the problem of smoothing or graduation. A first class of analytic approximation formulae. *Quart. Appl. Math.* **4**, 1946, 45–99.
25. E. Vanraes: *Powell-Sabin Splines and Multiresolution Techniques*. Ph.D. thesis, Katholieke Universiteit Leuven, 2004.
26. K. Willemans and P. Dierckx: Surface fitting using convex Powell-Sabin splines. *J. Comput. Appl. Math.* **56**, 1994, 263–282.
27. J. Windmolders: *Powell-Sabin Splines for Computer Aided Geometric Design*. Ph.D. thesis, Katholieke Universiteit Leuven, 2003.

Approximation with Asymptotic Polynomials

Philip Cooper¹, Alistair B. Forbes², and John C. Mason¹

¹ School of Computing and Engineering, University of Huddersfield, Huddersfield HD1 3DH, UK, {p.cooper, j.c.mason}@hud.ac.uk

² National Physical Laboratory, Teddington TW11 0LW, UK, alistair.forbes@npl.co.uk

Summary. Asymptotic behaviour associated with physical systems is quite common. However empirical models such as polynomials, splines and Fourier series do not lend themselves to modelling asymptotic behaviour. In this paper, we describe a straightforward modification of polynomial basis functions using a nonlinear weighting function that enables specified types of asymptotic behaviour to be modelled effectively. The weighting function depends on auxiliary parameters that control the effect of the weighting function. With these auxiliary parameters fixed, the approximation problem is linear and can be solved using standard linear least squares techniques. If one or more of the auxiliary parameters is unknown, nonlinear optimization techniques are necessary but they can be implemented in such a way so as to exploit the linearity with respect to the coefficients of the basis functions. In either case, appropriate use of orthogonal polynomials is required to avoid numerical instabilities.

1 Introduction

Asymptotic behaviour associated with physical systems is quite common. For example, a response may decay to a constant as time passes. However empirical models such as polynomials, splines and Fourier series [1, 2] do not lend themselves to modelling asymptotic behaviour. In this paper, we consider an easily implemented method to allow classes of asymptotic behaviour to be modelled effectively. The main idea is to modify polynomial basis functions using a nonlinear weighting function designed to enable the correct type of asymptotic behaviour to be modelled. These basis functions – *asymptotic polynomials* – are described in Section 2. In Section 3, we describe algorithms for approximation with asymptotic polynomials that exploit i) the fact that the basis functions are linear in all but a small number of the parameters, ii) orthogonal polynomials and iii) the fact that nonlinearity is introduced through nonlinear diagonal weighting matrices. In Section 4, we compare asymptotic polynomial and standard (Chebyshev) polynomial fits to metrology data. Our concluding remarks are given in Section 5.

2 Asymptotic Polynomials

Let $\{\phi_j(x)\}_{j=0}^n$ be a set of polynomial basis functions such as Chebyshev polynomials [6]. Define a weighting function

$$w(x) = w(x, \mathbf{b}) = \frac{1}{(1 + s^2(x - t)^2)^{k/2}}, \quad s > 0, \quad k > 0, \quad \mathbf{b} = (s, t, k)^T.$$

The weighting function $w(x)$ is smooth, $0 < w(x) \leq 1$, and $w(x)$ behaves like $|x|^{-k}$ as $|x| \rightarrow \infty$. Defining

$$\tilde{\phi}_j = w(x)\phi_j(x),$$

then

$$\tilde{\phi}(x, \mathbf{a}) = \sum_{j=0}^n a_j \tilde{\phi}_j(x)$$

behaves like x^{n-k} as $|x| \rightarrow \infty$. In particular, if $k = n$, then $\tilde{\phi}$ can model the asymptotic approach to a constant. For x limited to a finite interval, the constant s controls the degree to which asymptotic behaviour is imposed on the model within that interval. We refer to $\mathbf{b} = (s, t, k)^T$ as the *auxiliary* parameters associated with the model $\tilde{\phi}(x, \mathbf{a}, \mathbf{b})$.

Given abscissae $\mathbf{x} = (x_1, \dots, x_m)^T$, we denote by C the basis matrix generated from ϕ_i , i.e., $C_{ij} = \phi_j(x_i)$ and by $\tilde{C} = \tilde{C}(\mathbf{b})$ that from $\tilde{\phi}_i$ so that $\tilde{C}_{ij} = \tilde{\phi}_j(x_i) = w_i C_{ij}$, where $w_i = w(x_i, \mathbf{b})$.

Using the Forsythe method [3], the basis functions ϕ_j can be determined so that the modified basis matrix \tilde{C} is orthogonal, i.e., given abscissae \mathbf{x} and weights $\mathbf{w} = (w_1, \dots, w_m)^T$, we can generate polynomial basis functions $\phi_j(x)$ of degree j such that

$$\sum_{i=1}^m w_i^2 \phi_j^2(x_i) = 1, \quad \sum_{i=1}^m w_i^2 \phi_j(x_i) \phi_l(x_i) = 0, \quad l \neq j.$$

Figure 1 shows the first four orthogonal basis functions $\tilde{\phi}_j$ defined on the interval $[-1, 1]$ using the weight function $w(\mathbf{b})$ with $\mathbf{b} = (3, 0, 4)^T$.

3 Approximation with Asymptotic Polynomials

Suppose $\{(x_i, y_i)\}_{i=1}^m$ represent data points to which we wish to fit an asymptotic polynomial. With $\mathbf{b} = (s, t, k)^T$ fixed, the function $\tilde{\phi}$ is a linear combination of basis functions and the estimate of the coefficients \mathbf{a} is found by solving the linear least-squares system

$$\min_{\mathbf{a}} \|\mathbf{y} - \tilde{C}\mathbf{a}\|^2.$$

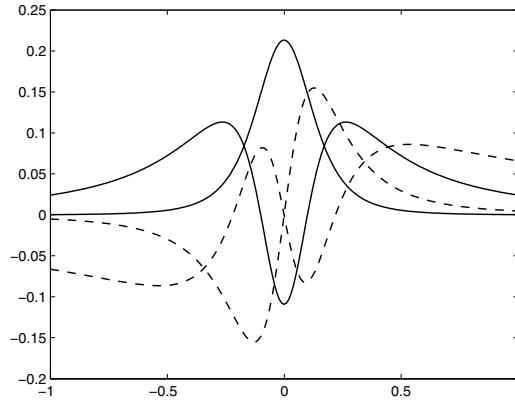


Fig. 1. First four orthogonal asymptotic polynomials generated for weight function $w(\mathbf{b})$ with $\mathbf{b} = (3, 0, 4)^T$.

More useful in practice is to regard one or more of s , t and k as additional parameters to be determined as part of the optimization in which case the matrix $\tilde{C} = \tilde{C}(\mathbf{b})$ is now a nonlinear function of \mathbf{b} and the fitting problem becomes

$$\min_{\mathbf{a}, \mathbf{b}} \|\mathbf{y} - \tilde{C}(\mathbf{b})\mathbf{a}\|^2, \tag{1}$$

a nonlinear least-squares problem.

The optimization problem (1) can be solved using the Gauss-Newton algorithm, for example [4], which requires (estimates of) the derivatives of the summand functions. Writing $\tilde{C}(\mathbf{b})$ as $\tilde{C}(\mathbf{b}) = W(\mathbf{b})C$ and $\mathbf{h}(\mathbf{a}, \mathbf{b}) = \mathbf{y} - W(\mathbf{b})C\mathbf{a}$, then the Jacobian matrix of partial derivatives of \mathbf{h} with respect to the optimization parameters is determined from

$$\frac{\partial \mathbf{h}}{\partial a_j} = -\tilde{\phi}_j, \quad \frac{\partial \mathbf{h}}{\partial b_l} = -\left(\frac{\partial W}{\partial b_l}\right)C\mathbf{a}.$$

Given an initial estimate \mathbf{b}_0 of the parameters \mathbf{b} , the polynomial basis can be chosen to be orthogonal with respect to the weights $w(\mathbf{b}_0)$ so that for \mathbf{b} close to \mathbf{b}_0 , the associated Jacobian matrix is relatively well-conditioned. In order to maintain well-conditioned matrices, we can periodically reparametrize the polynomials based on the current estimate of the auxiliary parameters \mathbf{b} .

By eliminating the parameters \mathbf{a} from the optimization it is possible to use an optimal parametrization throughout. We first consider the more general nonlinear least-squares problem

$$\min_{\mathbf{a}, \mathbf{b}} \mathbf{h}^T(\mathbf{a}, \mathbf{b})\mathbf{h}(\mathbf{a}, \mathbf{b}), \quad \mathbf{h}(\mathbf{a}, \mathbf{b}) = \mathbf{y} - C(\mathbf{b})\mathbf{a}, \tag{2}$$

where $C(\mathbf{b})$ is an $m \times n$ matrix, $m > n$, depending on parameters \mathbf{b} . The conditions for optimality require that, at the solution,

$$C^T(\mathbf{b})C(\mathbf{b})\mathbf{a} = C^T(\mathbf{b})\mathbf{y}, \quad (3)$$

that is, \mathbf{a} satisfies the normal equations for \mathbf{a} to be a least-squares solution of $C(\mathbf{b})\mathbf{a} = \mathbf{y}$. Equation (3) defines $\mathbf{a} = \mathbf{a}(\mathbf{b})$ implicitly as functions of \mathbf{b} . Writing $\mathbf{f}(\mathbf{b}) = \mathbf{y} - C(\mathbf{b})\mathbf{a}(\mathbf{b})$, (2) is equivalent to

$$\min_{\mathbf{a}} \mathbf{f}^T(\mathbf{b})\mathbf{f}(\mathbf{b}),$$

a nonlinear least-squares problem involving only the parameters \mathbf{b} . To solve this, we need to calculate \mathbf{a} , \mathbf{f} and

$$\mathbf{f}_l = -C_l\mathbf{a} - C\mathbf{a}_l, \quad (4)$$

where the subscript l means derivative with respect to b_l . Differentiating the normal equations (3) with respect to b_l , we find

$$\mathbf{a}_l = (C^T C)^{-1} [C_l^T \mathbf{f} - C^T C_l \mathbf{a}]. \quad (5)$$

We note here that (4) only requires us to calculate $C\mathbf{a}_l$ and \mathbf{a}_l is not required on its own. If C has QR decomposition $C = QR$, $Q \in \mathbb{R}^{m \times n}$, $R \in \mathbb{R}^{n \times n}$ [5] then

$$R\mathbf{a} = \mathbf{q}, \quad \text{where} \quad \mathbf{q} = Q^T \mathbf{f}, \quad (6)$$

and, from (5),

$$C\mathbf{a}_l = QR^{-T}C_l^T \mathbf{f} - QQ^T C_l \mathbf{a}.$$

If \mathbf{c}_l and \mathbf{q}_l are such that

$$R^T \mathbf{c}_l = C_l^T \mathbf{f}, \quad \mathbf{q}_l = Q^T (C_l \mathbf{a}), \quad (7)$$

then

$$C\mathbf{a}_l = Q(\mathbf{c}_l - \mathbf{q}_l).$$

In this way \mathbf{f} and its derivatives \mathbf{f}_l with respect to parameters b_l can be found by solving systems of equations (6) and (7) involving the upper-triangular matrix R and its transpose.

In applying this approach to approximation with asymptotic polynomials, we note that $\mathbf{f}(\mathbf{b}) = \mathbf{y} - \tilde{C}(\mathbf{b})\mathbf{a}(\mathbf{b})$ and its derivatives are necessarily independent of the choice of basis functions used to represent the polynomials. In particular, we can choose the Forsythe basis so that \tilde{C} is orthogonal. This means that \mathbf{f} and its derivatives can be calculated using only matrix-vector multiplications since R is the identity matrix in (6) and (7) (and $Q = \tilde{C}$).

One further efficiency gain can be made using the fact that $\tilde{C} = W(\mathbf{b})C$, where $W(\mathbf{b})$ is a diagonal weighting matrix with diagonal elements $w_i(\mathbf{b})$ with $w_i(\mathbf{b}) > 0$. Writing

$$\frac{\partial w_i}{\partial b_l} = d_{i,l} w_i, \quad \text{i.e.,} \quad d_{i,l} = \frac{1}{w_i} \frac{\partial w_i}{\partial b_l},$$

then $\tilde{C}_l = D_l \tilde{C}$ where D_l is the diagonal matrix with diagonal elements $d_{i,l}$. The quantities $\tilde{C}_l^T \mathbf{f} = \tilde{C}^T (D_l \mathbf{f})$ and $\tilde{C}_l \mathbf{a} = D_l (\tilde{C} \mathbf{a})$ used in (7) involve D_l only in vector-vector calculations; the matrices \tilde{C}_l need not be calculated.

The Gauss-Newton algorithm for minimizing a sum of squares $F(\mathbf{b}) = \mathbf{f}^T(\mathbf{b})\mathbf{f}(\mathbf{b})/2$ works well if the Jacobian matrix J , $J_{il} = \partial f_i / \partial b_l$, is such that $J^T J$ is a good approximation to the Hessian matrix of second partial derivatives of $F(\mathbf{b})$. We recall that the Newton step \mathbf{p}_N to update $\mathbf{b} := \mathbf{b} + \mathbf{p}_N$ is the solution of

$$H \mathbf{p}_N = -\mathbf{g}, \quad \mathbf{g} = J^T \mathbf{f}, \quad H = \nabla_{\mathbf{b}}^2 F = J^T J + \sum_i f_i \nabla_{\mathbf{b}}^2 f_i.$$

While the Gauss-Newton update step \mathbf{p}_{GN} solves $J^T J \mathbf{p}_{GN} = -J^T \mathbf{f}$, i.e., \mathbf{p}_{GN} is the least-squares solution of $J \mathbf{p}_{GN} = -\mathbf{f}$. A Newton update step leads to quadratic convergence near the solution while a Gauss-Newton update step has linear convergence, the rate of which depending on the adequacy of the approximation of $J^T J$ to the Hessian H . The approximation will be good if the summand functions are close to linear in optimization parameters. For the case of asymptotic polynomial approximation, the functions can have significant curvature, inhibiting the convergence of a Gauss-Newton algorithm. For this reason, there can be computational advantages in using a Newton update. In our implementation, we have used finite differences to approximate H .

4 Example Applications

Figure 2 shows a polynomial of degree 6 and an asymptotic polynomial of degree 3 fits to the sigmoid curve

$$y = \frac{2}{1 + e^{-x}} - 1.$$

(In many circumstances the response of a system to a step change in input has a sigmoid-type behaviour.) The asymptotic polynomial fit is indistinguishable from the sigmoid curve and the maximum error of approximation is less than 2.5×10^{-4} . The solution parameters are $\mathbf{b} = (0.091, 0.0, 3.0)^T$. The degree 6 polynomial fit is much worse. (In the examples considered here the degree of the standard polynomial is 3 more than the asymptotic polynomial so that both models have the same number of parameters.)

Figure 3 shows standard polynomial and asymptotic polynomial fits to data representing material properties of aluminium. In Figure 4, fits are compared on data representing the efficiency of the human eye response as a function of wavelength in daylight (photopic) conditions. In both cases, the asymptotic polynomial fits give a better representation of the data. In Figure 4, the asymptotic polynomial fit is barely distinguishable from the data.

Table 1 compares the norms of the update parameter \mathbf{p} for consecutive iterations of the Newton and Gauss-Newton methods, which were used to fit

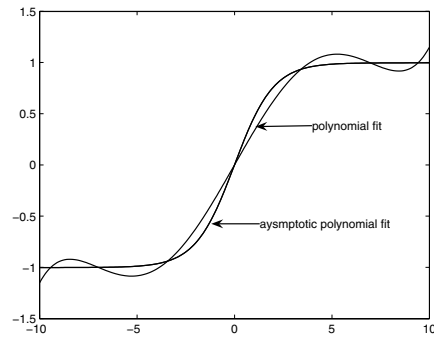


Fig. 2. Polynomial of degree 6 and asymptotic polynomial of degree 3 fits to a sigmoid curve.

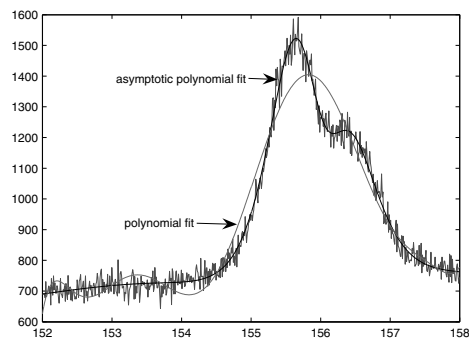


Fig. 3. Polynomial of degree 9 and asymptotic polynomial of degree 6 fits to measurements of material properties (for aluminium).

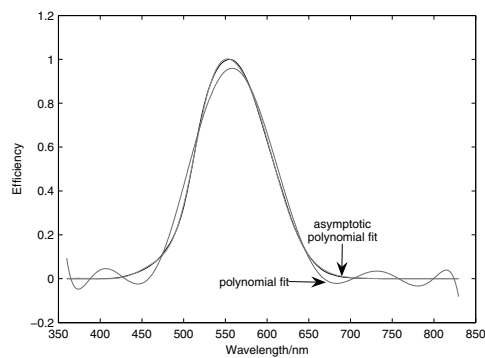


Fig. 4. Polynomial of degree 9 and asymptotic polynomial of degree 6 fits to the photopic efficiency function.

a degree 6 asymptotic polynomial to photopic efficiency function data (Figure 4). The results clearly demonstrate the superior performance of the Newton method in this particular example.

Table 1. Norm of update step \mathbf{p} in Newton and Gauss-Newton methods for the photopic efficiency function example (Figure 4).

Iteration	Gauss-Newton $\ \mathbf{p}\ _2$	Newton $\ \mathbf{p}\ _2$
1	0.8496	0.6573
2	0.3354	0.2203
3	0.1380	0.0019
4	0.0568	2.075 e-06
5	0.0235	3.855 e-13
6	0.0097	

5 Concluding Remarks

Data reflecting asymptotic behaviour can be modelled by polynomial basis functions multiplied by a nonlinear weighting function depending on three auxiliary parameters. Efficient and numerically stable optimization algorithms can be developed using polynomial basis functions orthogonal with respect to the weighting function. A parameter elimination scheme has been implemented that allows the approximation problem to be solved compactly. The model can easily be extended to allow for different asymptotic behaviour as $x \rightarrow \infty$ and $x \rightarrow -\infty$. Examples show that such asymptotic polynomial approximations can be much more effective than standard polynomial approximations.

Acknowledgement

This work was partially supported by the Department of Trade and Industry's Software Support for Metrology Programme.

References

1. R.M. Barker, M.G. Cox, A.B. Forbes, and P.M. Harris: *Software Support for Metrology Best Practice Guide 4: Modelling Discrete Data and Experimental Data Analysis*. Technical report, National Physical Laboratory, Teddington, 2004.

2. R. Boudjemaa, A.B. Forbes, P.M. Harris, and S. Langdell: *Multivariate Empirical Models and their Use in Metrology*. Technical Report, National Physical Laboratory, Teddington, 2003.
3. G.E. Forsythe: Generation and use of orthogonal polynomials for data fitting with a digital computer. *SIAM Journal* **5**, 1957, 74–88.
4. P. Gill, W. Murray, and M.H. Wright: *Practical Optimization*. Academic Press, London, 1981.
5. G.H. Golub and C.F. Van Loan: *Matrix Computations*. 3rd edition, John Hopkins University Press, Baltimore, 1996.
6. D.C. Handscombe and J.C. Mason: *Chebyshev Polynomials*. Chapman & Hall/CRC Press, London, 2003.

Spline Approximation Using Knot Density Functions

Andrew Crampton¹ and Alistair B. Forbes²

¹ School of Computing and Engineering, University of Huddersfield, Huddersfield HD1 3DH, UK, a.crampton@hud.ac.uk

² National Physical Laboratory, Teddington, Middlesex, TW11 0LW, UK, alistair.forbes@npl.co.uk

Summary. This paper is concerned with the approximation of discrete data using univariate B-splines. Specifically, we focus on the need to locate spline knots optimally in order to improve the fidelity of the B-spline model to the data. It is well understood that knot placement can have a significant effect on the quality of a spline approximant. However optimizing with respect to the number and placement of knots is generally difficult. In this paper, we describe an approach in which the density of knots is controlled by a knot density function depending on a small number of parameters. Optimizing with respect to these additional parameters is straightforward and can lead to significant improvements in the approximating spline.

1 Introduction

Knot placement can have a significant effect on the quality of a spline approximant. However optimizing with respect to the number and placement of knots is generally difficult. In this paper, we describe an approach in which an initial placement of knots is modified using a knot density function depending on a small number of parameters. Optimizing with respect to these additional parameters is straightforward and can lead to significant improvements in the approximating spline.

This paper is organized as follows. In Section 2, we describe the formulation of B-spline approximants in terms of flexible knot sets – flexi-knots – and discuss how such sets can be determined from cumulative density functions. Approximating data with flexi-knot splines is discussed in Section 3, together with regularization considerations and methods of solution. Example applications are presented in Section 4 with two types of knot density functions applied to initial knot sets determined using a uniform distribution and a knot insertion/deletion algorithm. Our concluding remarks are given in Section 5.

2 Definition of Flexi-Knot Splines

Let $\boldsymbol{\lambda}_0 = (\lambda_{0,1}, \dots, \lambda_{0,N})^T$, $0 < \lambda_{0,1} < \dots < \lambda_{0,N}$, be N distinct knots in the interval $[0, 1]$, and let $K(x, \mathbf{b})$ be a curve depending on parameters \mathbf{b} defined on $[0, 1]$ such that $K(0, \mathbf{b}) = 0$, $K(1, \mathbf{b}) = 1$ and $K'(x, \mathbf{b}) > 0$, $0 \leq x \leq 1$. We refer to such a K as a (cumulative) knot density curve. Given $\boldsymbol{\lambda}_0$ and K , the corresponding *flexi-knots* $\boldsymbol{\lambda} = \boldsymbol{\lambda}(\mathbf{b})$ are defined by

$$\lambda_k = K(\lambda_{0,k}, \mathbf{b}), \quad k = 1, \dots, N,$$

and therefore functions of the knot density curve parameters \mathbf{b} . The flexi-knot spline $s(x, \mathbf{a}, \mathbf{b})$ of order n is a linear combination

$$s(x, \mathbf{a}, \mathbf{b}) = \sum_{j=1}^Q a_j N_j(x, \boldsymbol{\lambda}(\mathbf{b})),$$

of the $Q = n + N$ B-spline basis functions $N_j(x, \boldsymbol{\lambda}(\mathbf{b}))$ determined on the flexi-knot set $\boldsymbol{\lambda}(\mathbf{b})$.

2.1 Example Knot Density Functions

Cumulative density functions (CDFs) for nonzero probability density functions (PDFs), defined on finite intervals are natural candidates for knot density functions. Here we describe two such functions, a piecewise linear CDF and the CDF associated with the beta distribution.

Piecewise Linear CDF

Let $\mathbf{b} = (p, q)^T$, $0 < p, q < 1$ and define

$$K(x, \mathbf{b}) = \begin{cases} qx/p, & x \leq p, \\ (1-q)(x-1)/(1-p) + 1, & x > p. \end{cases}$$

The knot density in the interval $[0, p]$ is q/p and that in the interval $[p, 1]$ is $(1-q)/(1-p)$. This type of knot density function can be useful if there is asymmetry in the behaviour of the data. It can also be used in combination with other knot density functions to divide an interval into two subintervals to which separate density functions are applied.

We can easily generalize this approach to an arbitrary number of subintervals. A piecewise linear cumulative density function with an arbitrary number n_K of control points (p_k, q_k) , $k = 1, \dots, n_K$, can be defined by repeating the process for generating a CDF with one control point. Let $\mathbf{u} = (u_1, \dots, u_{n_K})^T$ and $\mathbf{v} = (v_1, \dots, v_{n_K})^T$ be such that $0 < u_k, v_k < 1$, $k = 1, \dots, n_K$. Set $(p_{n_K}, q_{n_K}) = (u_{n_K}, v_{n_K})$ and for $k = n_K - 1, \dots, 1$, $(p_k, q_k) = (u_k p_{k+1}, v_k q_{k+1})$. The first line segment is $y = q_1 x / p_1$, the last

is $y = (1 - q_{n_K})(x - p_{n_K})/(1 - p_{n_K}) + q_{n_K}$ and intermediary segments of the form

$$y = m_k(x - p_k) + q_k, \quad m_k = \frac{q_{k+1} - q_k}{p_{k+1} - p_k}.$$

We note that in order to determine a uniformly spaced set of control points we set $u_k = v_k = k/(k + 1)$.

The control points define the CDF parametrically. As with many parametric representations, there can be more than one set of parameters \mathbf{p} and \mathbf{q} that represent the same shape. For example any \mathbf{p} and \mathbf{q} with $\mathbf{p} = \mathbf{q}$ represents the line $y = x$. In any optimization problem involving the parameters $\mathbf{b}^T = (\mathbf{p}^T, \mathbf{q}^T)$, a regularization term may be needed in order to make the optimization problem well posed; see Section 3.

Cumulative Density Function for the Beta Distribution

Another suitable distribution for constructing knot density functions is the (standard) beta distribution defined in the interval $[0, 1]$ which has PDF

$$p(x, p, q) = \frac{x^{p-1}(1-x)^{q-1}}{B(p, q)},$$

where $B(p, q)$ is the Beta function. The beta distribution has two shape parameters $\mathbf{b} = (p, q)^T$, $p, q > 0$. Its CDF is known as the incomplete beta function ratio defined as

$$K(x, p, q) = \int_0^x p(t, p, q) dt.$$

Figure 1 graphs the beta PDFs for i) $(p, q) = (1, 1)$, ii) $(p, q) = (0.5, 4)$, iii) $(p, q) = (4, 2)$ and iv) $(p, q) = (0.25, 0.5)$ while Figure 2 graphs the corresponding CDFs. The two shape parameters accord the distribution a wide range of qualitative behaviour.

3 Approximation with Flexi-Knot Splines

Suppose we have data $\{(x_i, y_i), i = 1, \dots, m\}$ with $\mathbf{x} = (x_1, \dots, x_m)^T$ and $\mathbf{y} = (y_1, \dots, y_m)^T$. We associate to \mathbf{x} the $m \times Q$ matrix $C = C(\mathbf{b})$ of basis functions defined by

$$C(i, j) = N_j(x_i, \boldsymbol{\lambda}(\mathbf{b})).$$

As with standard splines, C is a banded matrix with bandwidth n , the order of the spline. With $\mathbf{b} = \mathbf{b}_0$ fixed, the function $s(x, \mathbf{a}) = s(x, \mathbf{a}, \mathbf{b}_0)$ is a linear combination of basis functions and estimates of the parameters \mathbf{a} are found by solving the linear least-squares system

$$\min_{\mathbf{a}} \|\mathbf{y} - C_0 \mathbf{a}\|^2, \quad C_0 = C(\mathbf{b}_0).$$

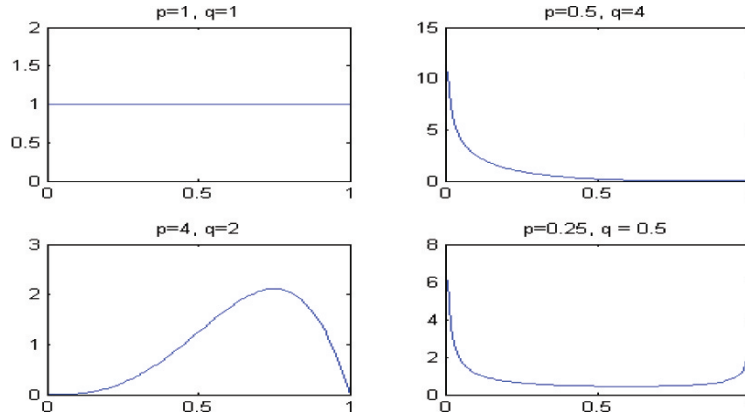


Fig. 1. Beta distribution PDFs for four sets of shape parameters $\mathbf{b} = (p, q)$.

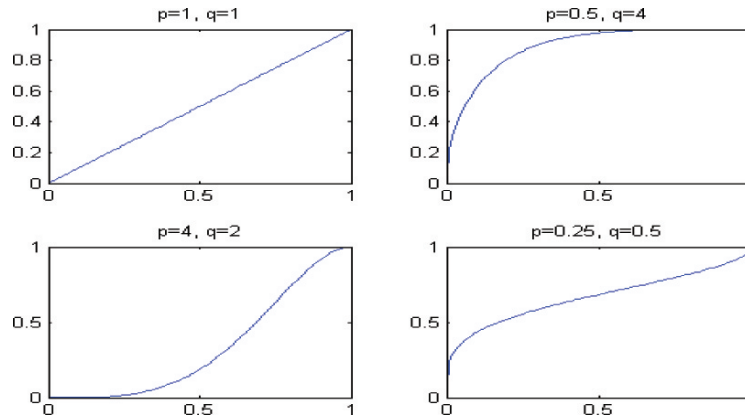


Fig. 2. Beta distribution CDFs for four sets of shape parameters $\mathbf{b} = (p, q)$.

More useful in practice is to regard \mathbf{b} as additional parameters to be determined as part of the optimization in which case the matrix $C = C(\mathbf{b})$ is now a nonlinear function of \mathbf{b} and the fitting problem becomes

$$\min_{\mathbf{a}, \mathbf{b}} \|\mathbf{y} - C(\mathbf{b})\mathbf{a}\|^2, \tag{1}$$

a nonlinear least-squares problem.

We can also include an additional regularization term of the form $H(\mathbf{b}) = \mathbf{h}^T(\mathbf{b})\mathbf{h}(\mathbf{b})$ into the objective function. This term can be used, for example, to control how far the flexi-knots are allowed to depart from the initial knot-set λ_0 or to improve the conditioning of the optimization problem.

For example, for the piecewise linear density function (Subsection 2.1), we let $b_k = 1/(1 + e^{-\tau_k})$ and parameterize the knot density function in terms of $\boldsymbol{\tau} = (\tau_1, \tau_2)^T$. In this case, the regularization term can have the form

$$H(\boldsymbol{\tau}) = w^2[(\tau_1 - 1)^2 + (\tau_2 - 1)^2].$$

As the weight w increases, the flexi-knots are biased more towards the original knot set. For the generalized piecewise linear density function the quantities u_k and v_k can be parameterized as $u_k = 1/(1 + e^{-\tau_k})$, etc., so that $\tau_k = \log u_k/(1 - u_k)$.

In the case of knot density curves determined from a beta distribution CDF, we let $p = \exp(\tau_1)$ and $q = \exp(\tau_2)$ and employ a regularization term of the form

$$H(\boldsymbol{\tau}) = w^2[\tau_1^2 + \tau_2^2],$$

again, biasing the flexi-knots towards the initial knot set for large w .

3.1 Gauss-Newton Algorithm

The optimization problem (1) can be solved using the Gauss-Newton algorithm [7]. Setting $\mathbf{f} = \mathbf{y} - C(\mathbf{b})\mathbf{a}$, the Jacobian matrix associated with (1) is determined from

$$\frac{\partial \mathbf{f}}{\partial a_j} = -C(:, j), \quad \frac{\partial \mathbf{f}}{\partial b_l} = -\frac{\partial C}{\partial b_l} \mathbf{a}.$$

To calculate the derivatives with respect to \mathbf{b} , we are required to evaluate

$$\frac{\partial N_j}{\partial \lambda_q}(x, \boldsymbol{\lambda}),$$

the derivative of the j th basis function with respect to the q th knot. For distinct internal knots and order $n \geq 2$, this derivative is calculated as follows [8]. Let $\boldsymbol{\tau}_q$ be the expanded knot set

$$\boldsymbol{\tau}_q = (\lambda_1, \dots, \lambda_q, \lambda_q, \dots, \lambda_N)^T,$$

that is, $\boldsymbol{\tau}_q$ is the same as $\boldsymbol{\lambda}$ but with the q th knot repeated. Then

$$\frac{\partial N_j}{\partial \lambda_q}(x, \boldsymbol{\lambda}) = d_{j,q} - d_{j-1,q}, \quad d_{j,q} = \frac{N_{j+1}(x, \boldsymbol{\tau}_q)}{\tau_{j+1} - \tau_{j-n+1}}.$$

Derivatives of the knot density function with respect to \mathbf{b} are also required. For a piecewise linear density curve, they are easily calculated. Derivatives for density curves arising as CDFs may be more difficult to evaluate. Algorithms for the derivative of the incomplete beta function are described in [1]. Alternatively, finite difference approximations can be used.

3.2 Elimination of the Parameters \mathbf{a}

It is possible to eliminate the parameters \mathbf{a} from the optimization completely. Fixing \mathbf{b} , the optimal \mathbf{a} in (1) represents the least-squares solution of

$$\min_{\mathbf{a}} \|\mathbf{y} - C\mathbf{a}\|, \quad C = C(\mathbf{b}).$$

The normal equations for the solution \mathbf{a} ,

$$C^T C \mathbf{a} = C^T \mathbf{y}, \quad (2)$$

implicitly define $\mathbf{a} = \mathbf{a}(\mathbf{b})$ as a function of \mathbf{b} and we can think of $\mathbf{f}(\mathbf{b}) = \mathbf{y} - C(\mathbf{b})\mathbf{a}(\mathbf{b})$ as a function of \mathbf{b} alone. In order to apply the Gauss-Newton algorithm to minimize $\mathbf{f}^T(\mathbf{b})\mathbf{f}(\mathbf{b})$, we require the derivatives of \mathbf{a} with respect to b_l . Differentiating (2) with respect to b_l with \mathbf{a} regarded as a function of \mathbf{b} , we have

$$C_l^T C \mathbf{a} + C^T C_l \mathbf{a} + C^T C \mathbf{a}_l = C_l^T \mathbf{y},$$

where the subscript l means differentiation with respect to b_l . This equation allows us to solve for \mathbf{a}_l in terms of C_l . A similar approach is described in more detail in [2].

For the case in which \mathbf{b} has only a small number of parameters, a very simple approach is to use function-only optimization. For data vectors \mathbf{x} and \mathbf{y} and auxiliary parameters \mathbf{b} , the objective function value $F = F(\mathbf{b})$ can be evaluated by the following steps:

- I Given \mathbf{x} , λ_0 and \mathbf{b} , evaluate flexi-knots $\boldsymbol{\lambda} = K(\lambda_0, \mathbf{b})$.
- II Evaluate nonzero elements of the banded matrix C .
- III From C and \mathbf{y} , calculate spline coefficients \mathbf{a} and residual vector $\mathbf{f} = \mathbf{y} - C\mathbf{a}$.
- IV From \mathbf{f} , calculate objection function value $F = \mathbf{f}^T \mathbf{f} / 2$.

A function-only approach can be effective for model fits involving a small number of knot density parameters. In step IV, the objective function can be modified to include a term $\mathbf{h}^T(\mathbf{b})\mathbf{h}(\mathbf{b})/2$ reflecting prior knowledge about \mathbf{b} . All of the calculations can be implemented so as to exploit the banded structure in the matrix C [4, 6]. This makes the function evaluations of F extremely cheap, computationally.

4 Example Applications

In this section, we illustrate the behaviour of flexi-knot splines in approximating metrology data representing thermo-physical measurements (related to heat flow as a function of temperature) graphed in Figure 3.

4.1 Uniform and Piecewise Linear Flexi-Knots

Figure 3 graphs spline fits determined from 15 uniformly spaced interior knots and from flexi-knots modified using an optimized piecewise linear density function. The two knot sets are also indicated. The associated residuals are graphed in Figure 4 and the knot density function used to determine the flexi-knots is illustrated in Figure 5. The flexi-knot fit is far superior with the maximum residual over 50 times smaller in absolute value compared with that for the fit based on a uniform knot set.

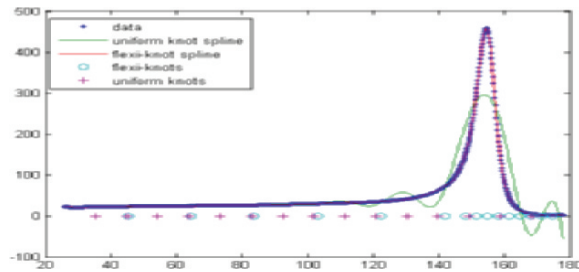


Fig. 3. Uniform and flexi-knot spline fits with 15 interior knots to measurements of thermo-physical properties with the flexi-knots determined using a piecewise linear density function.

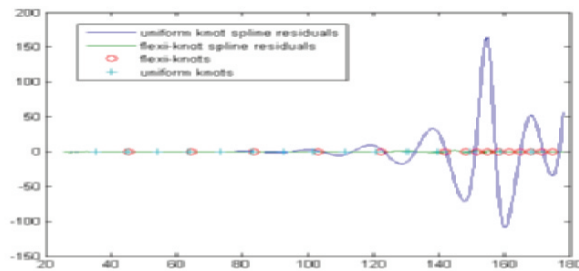


Fig. 4. Residuals associated with uniform and flexi-knot spline fits in Figure 3.

4.2 Knot Placement Algorithm and Beta Distribution Flexi-Knots

Uniform knot sets are known to perform poorly for data representing changing local behaviour. The flexi-knot approach can also be effective in improving fits determined using knot placement algorithms, as we illustrate below using a

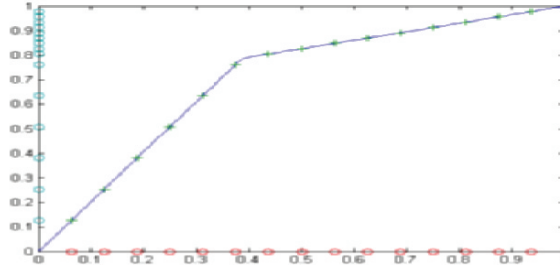


Fig. 5. Knot density function for the flexi-knot spline fit in Figure 3.

beta CDF knot density function. The initial knot placement strategy used is a modified insertion and deletion approach described in [5] and briefly explained below.

Knot insertion. For the knot insertion strategy, a small number of interior knots are initially chosen and uniformly distributed in $I = [x_{min}, x_{max}]$. For this initial knot set λ , we construct a B-spline approximant of order n and compute the vector of absolute residuals \mathbf{r} . The absolute values of the residuals, corresponding to abscissae not already in the knot set, are ordered and the abscissae values corresponding to the largest K of the ordered set (in the examples below, $K = 4$) are added to the knot set. At each iteration, a new spline is formed on the updated knot set and the process is repeated until

$$\text{var}(\mathbf{r}) \leq \text{TOL}_1.$$

Choosing a suitable TOL_1 depends largely on the particular application but can usefully be determined from an estimate of the standard deviation of noise in the data. The knot insertion approach generates a spline that is defined by significantly fewer knots than might ordinarily be required. The distribution of the knots is generally far from uniform with knots concentrated where the slope of the underlying curve represented by the data changes most rapidly.

Knot removal. In the current application, we delete knots based on a forward and backward difference examination of the spline coefficients obtained from the knot insertion algorithm. This ensures that a minimal number of spline coefficients are used to represent intervals in the data where little or no curvature is present. Thus, we remove redundancy whilst ensuring that the coefficients required to adequately recover the underlying curve are kept.

Specifically, let $\Delta a_k = a_{k+1} - a_k$ define the forward differences of \mathbf{a} and let the backward differences be defined as $\nabla a_k = a_k - a_{k-1}$. Define now the sum

$$\mathcal{D}a_k = |\Delta a_k| + |\nabla a_k|, \quad \text{for } k = 2, 3, \dots, p - n + 1,$$

where p is the number of B-spline coefficients and n is the order of the spline. The interior knot λ_k , corresponding to the B-spline coefficient a_k , is removed

from the knot set if $\mathcal{D}a_k \leq \text{TOL}_2$. A suitable choice for TOL_2 can again be obtained from an estimate of the standard deviation of noise in the data.

Figure 6 shows residuals computed from spline fits with 15 interior knots to measurements of thermo-physical properties (data is shown in Figure 3), with the flexi-knots determined using a beta distribution CDF applied to the initial distribution obtained using knot insertion and deletion (knot placement). The two sets of knots are also illustrated. The use of the knot density function reduces the maximum residual by a factor of 2. Comparing these results with those derived from a uniformly distributed initial knot set, we see that for this example the simple approach of assigning uniform knots and then optimising with respect to the knot density parameters is competitive with more elaborate knot placement algorithms.

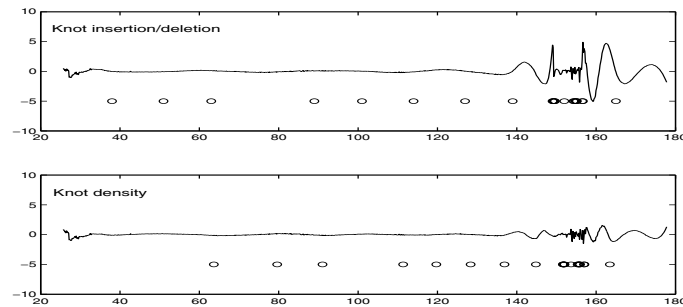


Fig. 6. Residuals associated with knot placement and flexi-knot spline fits to measurements of thermo-physical properties (Fig. 3) with the flexi-knots determined using a beta distribution CDF.

5 Concluding Remarks

In this paper we have presented an effective approach for locating the spline knots. We have shown how an initial knot placement can be optimized with respect to a small number of auxiliary parameters controlling the shape of a knot density function. The optimization can be performed efficiently by taking into account the banded structure in the matrix of evaluated B-spline basis functions. The examples presented demonstrate that the flexi-knot spline fits can model the data much more effectively than those based on uniform knots and can significantly improve the quality of the approximation when used to update knot distributions determined from knot insertion/deletion algorithms.

Acknowledgement

This work was partially supported by the Department of Trade and Industry's Software Support for Metrology Programme.

References

1. R.J. Boik and J.F. Robinson: Derivatives of the incomplete beta function. *J. Stat. Software* **3**, 1998, 1–20.
2. P. Cooper, A.B. Forbes, and J.C. Mason: Approximation with asymptotic polynomials. This volume.
3. M.G. Cox: The numerical evaluation of B-splines. *J. Inst. of Math. and its Applications* **8**, 1972, 36–52.
4. M.G. Cox: The least squares solution of overdetermined linear equations having band or augmented band structure. *IMA J. Numer. Analysis* **1**, 1981, 3–22.
5. M.G. Cox, P.M. Harris, and P.D. Kenward: Data approximation by polynomial splines. In: *Algorithms for Approximation IV*, University of Huddersfield, 2002, 331–345.
6. M.G. Cox, A.B. Forbes, P.M. Fossati, P.M. Harris, and I.M. Smith: *Techniques for the Efficient Solution of Large Scale Calibration Problems*. Technical report CMSC 25/03, National Physical Laboratory, Teddington, UK, May 2003.
7. P.E. Gill, W. Murray, and M.H. Wright: *Practical Optimization*. Academic Press, London, 1981.
8. P.M. Harris and I.M. Smith: *Testing Algorithms for Free-Knot Spline Approximation*. Technical Report CMSC 48/04, National Physical Laboratory, Teddington, UK, March 2004.

Neutral Data Fitting by Lines and Planes

Tim Goodman¹ and Chris Tofallis²

¹ Department of Mathematics, University of Dundee, Dundee DD1 5RD, UK,
tgoodman@maths.dundee.ac.uk

² University of Hertfordshire Business School, Hatfield, Herts AL10 9AB, UK,
c.tofallis@herts.ac.uk

1 Introduction

Given data on two or three variables we are interested in fitting a line or plane for the purposes of modelling the relationship between these variables. Most of the literature on this subject approaches this problem by selecting one of these variables and treating it differently from the others in the fitting procedure. This is acceptable if the purpose is to make predictions of that variable, but if we are seeking an underlying scientific law or a law-like relationship then it would seem more reasonable to treat all variables in the same way, unless there are particular reasons for not doing so. Thus in this paper we consider procedures for fitting lines or planes which treat all the variables equally.

Our approach to choosing a method is to stipulate certain desirable properties which one would expect a fitting procedure to possess, and then prove that there is a unique procedure which satisfies these properties. Attempts at laying down desirable properties for fitting lines to data have been made by the Nobel laureate Paul Samuelson [7] and the noted statistician William Kruskal [4]. In contrast to their approaches, we note that considering a line which ‘best fits’ given data suggests that the line minimises some measure of error between the data and the line, and so we consider desirable properties for the error measure itself. In Section 2 we stipulate and motivate seven such properties and show that they define the measure of error uniquely, up to a scaling factor. Our procedure for fitting a line to data is then to choose the line which minimises the sum of the squares of these errors. The resulting method is the same as that considered by Samuelson [7] and Kruskal [4]. In fact the method had appeared earlier in various contexts, see [5, 8, 9]. More recently it has been recommended in [1]. Since it has been given various names and accreditations, we shall refer to it simply as ‘neutral data fitting’, both to preserve our neutrality and to indicate that no variable is given special treatment. We note that a Bayesian approach to fitting a line to data, where both variables are treated the same, makes different assumptions and results in different solutions, see [10].

In Section 3 we extend neutral data fitting to the case of fitting a plane to data in three dimensions. Analogous properties for the error between a data point and a plane lead to a corresponding definition for the error. Then our choice of plane to fit the data is that which minimises the sum of the squares of the errors of the data. For neutral data fitting in two dimensions there are certain exceptional cases where there are two lines which best fit the data, and similarly for three dimensions there are exceptional cases where there are two, three or four planes of best fit. We show that in all other cases the plane of best fit is unique and is determined by the unique solution in a given interval of a quartic equation. In general this root will need to be determined numerically but we show that for certain special classes of data there are closed form solutions. To our knowledge no such type of data fitting technique has been considered before.

For further historical details and descriptions of situations where neutral data fitting could be applied, see [2].

2 Neutral Data Fitting in Two Dimensions

Suppose two real variables x and y are connected through a relationship which is symmetric in the sense that y does not depend on x any more or less than x depends on y . We are given a sequence of data $(x, y) = (\alpha_i, \beta_i)$ in \mathbb{R}^2 , $i = 1, \dots, n$, $n \geq 2$, and wish to find a straight line which ‘best fits’ the data. The usual procedure is to define some measure of the ‘error’ between a point (α, β) and a line L , and then choose the line L which minimises some ‘aggregate’ of these errors over the points (α_i, β_i) , $i = 1, \dots, n$. The measure of the error between (α, β) and L is some non-negative number which we denote by $F(\alpha, \beta, L)$. The usual choice of aggregate is the sum of the squares of the errors, i.e., we minimise $\sum_{i=1}^n F(\alpha_i, \beta_i, L)^2$ over all lines L . Of course we could choose other aggregates, e.g. $\sum_{i=1}^n F(\alpha_i, \beta_i, L)^p$ for some p , $1 \leq p < \infty$, or $\max_{i=1, \dots, n} F(\alpha_i, \beta_i, L)$, but our first concern here is not with this but with the choice of error function $F(\alpha, \beta, L)$.

Any line L has an equation of the form $ax + by + c = 0$, for real numbers a, b, c . Since we are assuming that there is some relationship between x and y , we do not consider lines which are parallel to the x - or y -axes. So we may denote the error between a point (α, β) and a line with equation $ax + by + c = 0$ by $F(\alpha, \beta, a, b, c)$, for α, β, a, b, c in \mathbb{R} with $a, b \neq 0$. We shall consider various properties which we would reasonably expect such a function F to satisfy and we shall prove that these properties determine F up to a positive constant multiple. (The formulae below hold for all α, β, a, b, c in \mathbb{R} with $a, b \neq 0$.)

Property 1: For any number $\lambda \neq 0$, the equation $\lambda ax + \lambda by + \lambda c = 0$ gives the same line as the equation $ax + by + c = 0$. So we must have

$$F(\alpha, \beta, \lambda a, \lambda b, \lambda c) = F(\alpha, \beta, a, b, c), \quad \lambda \neq 0. \quad (1)$$

Property 2: Clearly the error should be zero if and only if (α, β) lies on L , i.e.,

$$F(\alpha, \beta, a, b, c) = 0 \iff a\alpha + b\beta + c = 0. \quad (2)$$

Property 3: We would not expect the error to depend on the choice of origin of co-ordinates, i.e., if we shift both the point and line by the same vector, then the error should be unchanged. If the vector is (u, v) , then the point (α, β) is shifted to $(\alpha + u, \beta + v)$ and the line $ax + by + c = 0$ is shifted to $a(x - u) + b(y - v) + c = 0$. Thus we have

$$F(\alpha + u, \beta + v, a, b, c - au - bv) = F(\alpha, \beta, a, b, c), \quad u, v \in \mathbb{R}. \quad (3)$$

Property 4: A crucial assumption is that we treat x and y equally. Thus the error should be unchanged if we interchange x and y , i.e.,

$$F(\beta, \alpha, b, a, c) = F(\alpha, \beta, a, b, c). \quad (4)$$

Property 5: We would expect the error to be unchanged under a reflection of the x -variable, i.e., a reflection in the y -axis. Thus we have

$$F(-\alpha, \beta, -a, b, c) = F(\alpha, \beta, a, b, c). \quad (5)$$

Of course, (4) and (5) imply that the same holds for a reflection of the y -variable.

Property 6: If we scale x and y by a factor $\lambda > 0$, it is reasonable to have a corresponding change for the error, i.e.,

$$F(\lambda\alpha, \lambda\beta, a, b, \lambda c) = \lambda F(\alpha, \beta, a, b, c), \quad \lambda > 0. \quad (6)$$

Property 7: In Property 6 we considered scaling both x and y . We now consider a change of scale in an individual variable, say x . It would seem reasonable that the scaling of the error is independent of the choice of point and line, which is equivalent to the optimal line being always preserved by a change of scale in the x -variable. Thus we have

$$F(\lambda\alpha, \beta, a, \lambda b, \lambda c) = f(\lambda)F(\alpha, \beta, a, b, c), \quad \lambda > 0, \quad (7)$$

for some function $f : (0, \infty) \rightarrow (0, \infty)$. Of course a similar result follows if we consider a scaling of y .

Theorem 1. *If F is a function from $\{(\alpha, \beta, a, b, c) \in \mathbb{R}^5 : ab \neq 0\}$ to $[0, \infty)$ satisfying (1)-(7), then for some $k > 0$ this error function will take the form*

$$F(\alpha, \beta, a, b, c) = k \frac{|a\alpha + b\beta + c|}{|ab|^{\frac{1}{2}}}, \quad \alpha, \beta, a, b, c \in \mathbb{R}.$$

Proof. By (7), (4) and (1) we have for $\alpha, \beta, a, b, c \in \mathbb{R}$, $ab \neq 0$,

$$\begin{aligned} F(\lambda\alpha, \lambda\beta, a, b, \lambda c) &= f(\lambda)F\left(\alpha, \lambda\beta, a, \frac{b}{\lambda}, c\right) \\ &= f(\lambda)F\left(\lambda\beta, \alpha, \frac{b}{\lambda}, a, c\right) \\ &= f(\lambda)^2F\left(\beta, \alpha, \frac{b}{\lambda}, \frac{a}{\lambda}, \frac{c}{\lambda}\right) \\ &= f(\lambda)^2F(\alpha, \beta, a, b, c), \end{aligned}$$

and so by (6),

$$f(\lambda) = \sqrt{\lambda}, \quad \lambda > 0. \quad (8)$$

By (1) and (3), for α, β, a, b, c in \mathbb{R} , $ab \neq 0$, we have

$$F(\alpha, \beta, a, b, c) = F\left(\alpha, \beta + \frac{c}{b}, \frac{a}{b}, 1, 0\right). \quad (9)$$

By (2) let us define G using:

$$F(\alpha, \beta, a, 1, 0) = \frac{|a\alpha + \beta|}{|a|^{\frac{1}{2}}}G(\alpha, \beta, a). \quad (10)$$

for a function G defined for a, α, β in \mathbb{R} , $a \neq 0$. By (6) we have

$$G(\lambda\alpha, \lambda\beta, a) = G(\alpha, \beta, a), \quad (11)$$

and by (1), (7) and (8),

$$G\left(\lambda\alpha, \beta, \frac{a}{\lambda}\right) = G(\alpha, \beta, a), \quad (12)$$

where (11) and (12) hold for $\lambda > 0$, α, β, a in \mathbb{R} , $a \neq 0$. Applying (11) and (12) gives

$$G(\alpha, \beta, a) = G\left(\frac{\alpha}{\beta}, 1, a\right) = G\left(1, 1, \frac{a\alpha}{\beta}\right) \quad (13)$$

for $\alpha, \beta > 0$, $a \neq 0$.

Now by (5),

$$G(-\alpha, \beta, -a) = G(\alpha, \beta, a), \quad (14)$$

and by (4)

$$G(\alpha, -\beta, -a) = G(\alpha, \beta, a), \quad (15)$$

where (14) and (15) hold for α, β, a in \mathbb{R} , $a > 0$. Thus (13) holds for any $\alpha, \beta \neq 0$. For convenience, we write

$$g(t) = G(1, 1, t), \quad t \neq 0,$$

so that by (9), (10) and (13),

$$F(\alpha, \beta, a, b, c) = \frac{|a\alpha + b\beta + c|}{|ab|^{\frac{1}{2}}} g\left(\frac{a\alpha}{b\beta + c}\right) \quad (16)$$

for α, β, a, b, c in \mathbb{R} , $ab \neq 0$, $\alpha \neq 0$, $b\beta + c \neq 0$. Applying (3) also gives for $u \in \mathbb{R}$,

$$F(\alpha, \beta, a, b, c) = \frac{|a\alpha + b\beta + c|}{|ab|^{\frac{1}{2}}} g\left(\frac{a\alpha + au}{b\beta + c - au}\right), \quad (17)$$

provided $a + u \neq 0$, $b\beta + c \neq au$. Putting $\alpha = t$, $a = b = \beta = 1$, $c = 0$ in (16) and (17) gives

$$g(t) = g\left(\frac{t+u}{1-u}\right),$$

provided $t \neq 0$, $t+u \neq 0$, $u \neq 1$. In particular,

$$g(1) = g\left(\frac{1+u}{1-u}\right), \quad u \neq \pm 1,$$

and so $g(t) = g(1)$ for all $t \neq 0, -1$. Putting $g(1) = k$, we see from (2) that $k > 0$. Now for any α, β, a, b, c in \mathbb{R} , $ab \neq 0$, choosing u with $u + \alpha \neq 0$, $au \neq b\beta + c$ and substituting into (17) gives the result. \square

We now return to the problem of finding a line with equation $ax+by+c=0$ which best fits the data $(x, y) = (\alpha_i, \beta_i)$, $i = 1, \dots, n$, $n \geq 2$. We shall take the aggregate of the errors to be the sum of squares, and hence must find a, b, c in \mathbb{R} , $ab \neq 0$, which minimises

$$f(a, b, c) := \sum_{j=1}^n \frac{(a\alpha_j + b\beta_j + c)^2}{|ab|}.$$

(Clearly the constant k in Theorem 1 is irrelevant.)

First suppose $ab > 0$. Then there is no loss of generality in supposing $a > 0$, $b > 0$, $ab = 1$ (since we can always transform one of the variables by multiplying by -1 to ensure the coefficients are positive, and we can divide through the equation by a constant to ensure $ab = 1$). The problem then becomes to minimise

$$f(a, b, c) = \sum_{j=1}^n (a\alpha_j + b\beta_j + c)^2$$

over a, b, c in \mathbb{R} , $a, b > 0$, $ab = 1$. Since $f(a, b, c) \rightarrow \infty$ as we approach the boundary of this region, the minimum occurs when the Lagrangian

$$g(a, b, c, \lambda) := \sum_{j=1}^n (a\alpha_j + b\beta_j + c)^2 + \lambda(ab - 1)$$

satisfies $\frac{\partial g}{\partial a} = \frac{\partial g}{\partial b} = \frac{\partial g}{\partial c} = 0$, i.e.,

$$\begin{aligned}
& \sum_{j=1}^n \alpha_j (a\alpha_j + b\beta_j + c) + \lambda b \\
&= \sum_{j=1}^n \beta_j (a\alpha_j + b\beta_j + c) + \lambda a \\
&= \sum_{j=1}^n (a\alpha_j + b\beta_j + c) = 0.
\end{aligned}$$

Putting

$$\begin{aligned}
\bar{\alpha} &= \frac{1}{n} \sum_{j=1}^n \alpha_j, & \bar{\beta} &= \frac{1}{n} \sum_{j=1}^n \beta_j, & \sigma^2 &= \frac{1}{n} \sum_{j=1}^n \alpha_j^2, & \tau^2 &= \frac{1}{n} \sum_{j=1}^n \beta_j^2, \\
\nu &= \frac{1}{n} \sum_{j=1}^n \alpha_j \beta_j, & \bar{\lambda} &= \frac{1}{2n} \lambda,
\end{aligned}$$

where $\sigma, \tau > 0$, this becomes

$$a\sigma^2 + b\nu + c\bar{\alpha} + \bar{\lambda}b = 0, \quad (18)$$

$$a\nu + b\tau^2 + c\bar{\beta} + \bar{\lambda}a = 0, \quad (19)$$

$$a\bar{\alpha} + b\bar{\beta} + c = 0. \quad (20)$$

Substituting for c from (20) into (18) and (19) gives

$$a(\sigma^2 - \bar{\alpha}^2) + b(\nu - \bar{\alpha}\bar{\beta}) + \bar{\lambda}b = 0,$$

$$a(\nu - \bar{\alpha}\bar{\beta}) + b(\tau^2 - \bar{\beta}^2) + \bar{\lambda}a = 0,$$

and eliminating gives

$$a^2(\sigma^2 - \bar{\alpha}^2) = b^2(\tau^2 - \bar{\beta}^2). \quad (21)$$

We are not considering the trivial case $\alpha_j = \bar{\alpha}$, $j = 1, \dots, n$, when all the data lie on the line $x = \bar{\alpha}$. Thus

$$\sigma^2 - \bar{\alpha}^2 = \frac{1}{n} \left(\sum_{j=1}^n \alpha_j^2 - n\bar{\alpha}^2 \right) = \frac{1}{n} \sum_{j=1}^n (\alpha_j - \bar{\alpha})^2 > 0.$$

Similarly,

$$\tau^2 - \bar{\beta}^2 = \frac{1}{n} \sum_{j=1}^n (\beta_j - \bar{\beta})^2 > 0.$$

Now from (20) and $ab = 1$, we have

$$\begin{aligned} \frac{1}{n}f(a, b, c) &= a^2\sigma^2 + b^2\tau^2 + c^2 + 2\nu + 2ac\bar{\alpha} + 2bc\bar{\beta} \\ &= a^2\sigma^2 + b^2\tau^2 + 2\nu - (a\bar{\alpha} + b\bar{\beta})^2 \\ &= a^2(\sigma^2 - \bar{\alpha}^2) + b^2(\tau^2 - \bar{\beta}^2) + 2(\nu - \bar{\alpha}\bar{\beta}), \end{aligned}$$

and by (21) and $ab = 1$,

$$\frac{1}{2n}f(a, b, c) = (\sigma^2 - \bar{\alpha}^2)^{\frac{1}{2}}(\tau^2 - \bar{\beta}^2)^{\frac{1}{2}} + \nu - \bar{\alpha}\bar{\beta}. \tag{22}$$

If $ab < 0$, we similarly assume $ab = -1$ and again derive (18)-(20). In this case,

$$\frac{1}{2n}f(a, b, c) = (\sigma^2 - \bar{\alpha}^2)^{\frac{1}{2}}(\tau^2 - \bar{\beta}^2)^{\frac{1}{2}} - \nu + \bar{\alpha}\bar{\beta}. \tag{23}$$

So, if $\nu - \bar{\alpha}\bar{\beta} = \sum_{j=1}^n(\alpha_j - \bar{\alpha})(\beta_j - \bar{\beta}) < 0$, then f attains its minimum when $ab > 0$, while if $\nu - \bar{\alpha}\bar{\beta} > 0$, the minimum occurs when $ab < 0$.

To summarise, the optimal line passes through the mean of the data $(\bar{\alpha}, \bar{\beta})$, by (20), and has slope m , where by (21),

$$m^2 = \frac{\sum_{j=1}^n(\beta_j - \bar{\beta})^2}{\sum_{j=1}^n(\alpha_j - \bar{\alpha})^2}.$$

If $\sum_{j=1}^n(\alpha_j - \bar{\alpha})(\beta_j - \bar{\beta}) \neq 0$, then by (22) and (23), m has the same sign as $\sum_{j=1}^n(\alpha_j - \bar{\alpha})(\beta_j - \bar{\beta})$, the covariance. If $\sum_{j=1}^n(\alpha_j - \bar{\alpha})(\beta_j - \bar{\beta}) = 0$, then there are two optimal lines with slopes $\pm m$. In statistical terms, our line has a slope of magnitude given by the ratios of the standard deviations of the variables. An exact form for the confidence interval of the slope due to Jolicoeur and Mosimann is given in [6].

3 Three Dimensions

Suppose that (α, β, γ) is a point in \mathbb{R}^3 and $ax + by + cz + d = 0$ is the equation of a plane. Then it can be shown, as in Section 1, that if $F(\alpha, \beta, \gamma, a, b, c, d)$ represents a measure of the error of the point with respect to the plane which satisfies properties analogous to (1)-(7) in Section 1, then

$$F(\alpha, \beta, \gamma, a, b, c, d) = k \frac{|a\alpha + b\beta + c\gamma + d|}{|abc|^{\frac{1}{3}}},$$

for a constant $k > 0$. Since the equation of the plane is invariant under multiplication by a non-zero number, we may assume $abc > 0$.

Now take points $(\alpha_i, \beta_i, \gamma_i)$, $i = 1, \dots, n$, $n \geq 3$. We shall again take the aggregate of the errors of these points from a plane to be the sum of squares. Thus to find a plane with equation $ax + by + cz + d = 0$ which best fits the above data, we need to find a, b, c, d in \mathbb{R} , $abc > 0$, which minimise

$$f(a, b, c, d) = \sum_{j=1}^n \frac{(a\alpha_j + b\beta_j + c\gamma_j + d)^2}{(abc)^{\frac{2}{3}}}. \tag{24}$$

At the minimum we shall have $\frac{\partial f}{\partial d} = 0$ and so

$$a\bar{\alpha} + b\bar{\beta} + c\bar{\gamma} + d = 0, \tag{25}$$

where

$$\bar{\alpha} = \frac{1}{n} \sum_{j=1}^n \alpha_j, \quad \bar{\beta} = \frac{1}{n} \sum_{j=1}^n \beta_j, \quad \bar{\gamma} = \frac{1}{n} \sum_{j=1}^n \gamma_j,$$

i.e., the required plane passes through the mean of the data $(\bar{\alpha}, \bar{\beta}, \bar{\gamma})$. Substituting from (25) into (24), we need to find a, b, c in \mathbb{R} , $abc > 0$, which minimise

$$f(a, b, c) = \sum_{j=1}^n \frac{(a(\alpha_j - \bar{\alpha}) + b(\beta_j - \bar{\beta}) + c(\gamma_j - \bar{\gamma}))^2}{(abc)^{\frac{2}{3}}}.$$

We may ignore the trivial case $\alpha_j = \bar{\alpha}$, $j = 1, \dots, n$, i.e., when all the data lie in the plane $x = \bar{\alpha}$. Similarly, we ignore the cases $\beta_j = \bar{\beta}$, $j = 1, \dots, n$ and $\gamma_j = \bar{\gamma}$, $j = 1, \dots, n$. Then we define $s_1, s_2, s_3 > 0$ by

$$s_1^2 = \frac{1}{n} \sum_{j=1}^n (\alpha_j - \bar{\alpha})^2, \quad s_2^2 = \frac{1}{n} \sum_{j=1}^n (\beta_j - \bar{\beta})^2, \quad s_3^2 = \frac{1}{n} \sum_{j=1}^n (\gamma_j - \bar{\gamma})^2.$$

We also define

$$\begin{aligned} s_{12} &= \frac{1}{n} \sum_{j=1}^n (\alpha_j - \bar{\alpha})(\beta_j - \bar{\beta}), \\ s_{23} &= \frac{1}{n} \sum_{j=1}^n (\beta_j - \bar{\beta})(\gamma_j - \bar{\gamma}), \\ s_{13} &= \frac{1}{n} \sum_{j=1}^n (\alpha_j - \bar{\alpha})(\gamma_j - \bar{\gamma}). \end{aligned}$$

Putting

$$x = as_1, \quad y = bs_2, \quad z = cs_3, \quad \lambda = \frac{s_{23}}{s_2s_3}, \quad \mu = \frac{s_{13}}{s_1s_3}, \quad \nu = \frac{s_{12}}{s_1s_2},$$

we have

$$f(a, b, c) = n(s_1s_2s_3)^{\frac{2}{3}}g(x, y, z),$$

where

$$g(x, y, z) = \frac{x^2 + y^2 + z^2 + 2\lambda yz + 2\mu xz + 2\nu xy}{(xyz)^{\frac{2}{3}}}. \tag{26}$$

Thus the problem is equivalent to minimising $g(x, y, z)$ over x, y, z in \mathbb{R} , where $xyz > 0$.

Now suppose $\lambda = 1$. Thus there is some constant $\alpha \neq 0$ such that

$$\beta_j - \bar{\beta} = \alpha(\gamma_j - \bar{\gamma}), \quad j = 1, \dots, n,$$

and so the data lie on the plane $y - \alpha z = 0$. This plane is not among those considered by this method. Indeed, we assume that there is some relation involving all the variables and since the case $\lambda = 1$ would deem the x -variable to be irrelevant, we ignore this case. Similarly, by making a transformation, we may assume $\lambda \neq -1$, i.e., $|\lambda| < 1$. Similarly, we assume $|\mu| < 1$, $|\nu| < 1$. Note that $g(x, y, z)$ is invariant under making the same permutation of (x, y, z) and of (λ, μ, ν) . It is also invariant under the following transformations:

$$\begin{aligned} x &\rightarrow -x, & \mu &\rightarrow -\mu, & \nu &\rightarrow -\nu, \\ y &\rightarrow -y, & \lambda &\rightarrow -\lambda, & \nu &\rightarrow -\nu, \\ z &\rightarrow -z, & \lambda &\rightarrow -\lambda, & \mu &\rightarrow -\mu. \end{aligned}$$

There is therefore no loss of generality in assuming either $0 \leq \lambda \leq \mu \leq \nu < 1$ or $-1 < \lambda < 0, 0 \leq \mu, \nu < 1$.

We now give a result describing the complete solution to the above minimisation problem. We first give several special cases where the solution can be described explicitly, and then in Case 6 we give the generic case where the solution is given in terms of a solution of a quartic equation. Of course, any solution for (x, y, z) can be multiplied by any non-zero constant to give another solution. Due to lack of space, the proof cannot be included here; it can be found in [2].

Theorem 2. *The minimum value of (26) over x, y, z in \mathbb{R} , $xyz > 0$ is given as follows:*

Case 1: *If $\mu = \nu = 0$, then*

$$(x, y, z) = (\sqrt{1 + \lambda}, 1, 1) \quad \text{or} \quad (\sqrt{1 + \lambda}, -1, -1), \quad \lambda < 0,$$

$$(x, y, z) = (1, 1, 1), (1, -1, -1), (-1, 1, -1) \quad \text{or} \quad (-1, -1, 1), \quad \lambda = 0.$$

Case 2: *If $\lambda = \mu = \nu > 0$, then*

$$(x, y, z) = (1 + \lambda, -1, -1), (-1, 1 + \lambda, -1) \quad \text{or} \quad (-1, -1, 1 + \lambda).$$

Case 3: *If $0 \leq \lambda < \mu = \nu$ or $\lambda < 0 < \mu = \nu$, then*

$$(x, y, z) = (\mu + \sqrt{\mu^2 + 4\lambda + 4}, -2, -2).$$

Case 4: *If $0 < \lambda = \mu < \nu$, then*

$$(x, y, z) = (-\alpha, \beta, -1) \quad \text{or} \quad (\beta, -\alpha, -1),$$

where

$$\alpha = \frac{1}{2} \left(\sqrt{\mu^2 + 4 \frac{1-\mu^2}{1-\nu}} - \mu \right), \quad \beta = \frac{1}{2} \left(\sqrt{\mu^2 + 4 \frac{1-\mu^2}{1-\nu}} + \mu \right).$$

Case 5: If $\mu \geq 0$, $\nu > 0$, $\lambda = -\nu$, then

$$(x, y, z) = (2, -\nu - \sqrt{\nu^2 - 4\mu + 4}, -2).$$

Case 6: Suppose $0 \leq \lambda < \mu < \nu$ or $\lambda < 0 \leq \mu < \nu$, $\mu, \nu \neq -\lambda$. Then

$$(x, y, z) = (1, -\alpha, -\beta),$$

where α satisfies $P(-\alpha) = 0$, where

$$P(X) := (1 - \lambda^2)X^4 + \lambda(\mu - \lambda\nu)X^3 + 2(\lambda\mu\nu - 1)X^2 + \mu(\lambda - \mu\nu)X + 1 - \mu^2,$$

$$\text{and } \beta = \frac{1-\alpha^2}{\mu+\lambda\alpha}.$$

If $\lambda > -\mu$, then $-\alpha$ is the unique zero of P in $(-1, 0)$ and $0 < \beta < 1$.

If $-\nu < \lambda < -\mu$, then $-\alpha$ is the unique zero of P in $(-\infty, -1)$ and $0 < \beta < 1$.

If $\lambda < -\nu$, then $-\alpha$ is the unique zero of P in $(-\infty, -1)$ and $\beta > 1$.

Acknowledgement

We wish to thank the referee for informing us about the Bayesian approach to the problem for two variables.

References

1. N.R. Draper and H. Smith: *Applied Regression Analysis*. 3rd edition, Wiley, New York, 1998.
2. T.N.T. Goodman and C. Tofallis: Neutral data fitting in two and three dimensions. Working Paper, Business School, University of Hertfordshire, 2003.
3. K.A. Kermack and J.B.S. Haldane: Organic correlation and allometry. *Biometrika* **37**, 1950, 30–41.
4. W.H. Kruskal: On the uniqueness of the line of organic correlation. *Biometrics* **9**, 1953, 47–58.
5. W.E. Ricker: Linear regressions in fishery research. *J. Fisheries Research Board of Canada* **30**, 1973, 409–434.
6. W.E. Ricker: Computation and uses of central trend lines. *Canadian J. of Zoology* **62**, 1984, 1897–1905.
7. P.A. Samuelson: A note on alternative regressions. *Econometrica* **10**, 1942, 80–83.
8. G. Stromberg: Accidental systematic errors in spectroscopic absolute magnitudes for dwarf GoK2 stars. *Astrophysical J.* **92**, 1940, 156ff.
9. H. Sverdrup: Druckgradient, Wind und Reibung an der Erdoberfläche. *Ann. Hydrogr. u. Maritimen Meteorol. (Berlin)* **44**, 1916, 413–427.
10. A. Zellner: *An Introduction to Bayesian Inference in Econometrics*. Wiley, New York, 1971, second printing Krieger, Malabar, 1987.

Approximation on an Infinite Range to Ordinary Differential Equations Solutions by a Function of a Radial Basis Function

Damian P. Jenkinson and John C. Mason

School of Computing and Engineering, University of Huddersfield, Huddersfield
HD1 3DH, UK, {d.p.jenkinson,j.c.mason}@hud.ac.uk

Summary. A function $y = g(L)$ of a linear form $L(x) = \sum_{j=1}^n c_j \phi_j(x)$ has already been adopted in the approximation of a variety of smooth functions, especially those that behave like a power of x as $x \rightarrow \infty$. In particular, Mason [7] in his thesis considers $g(L) = L^{-R}$ for approximating a decaying function, where R is a power of 2 and $L(x)$ is a polynomial. Mason and Upton [8] use $g(L) = L^{-R}$ and $g(L) = e^L$, and in the latter case adopt a basis of Gaussian radial basis functions. Also, Crampton et al. [3] discuss “additive” linear iteration algorithms which are in general convergent to near-best approximations, and Dunham and Williams [5] discuss the existence of best approximations of the form $g(L)$, especially L^{-R} .

In the present study, we find that approximations of the form $g(L(x))$, where L is a radial basis function (RBF) of the cubic, multiquadric or inverse-multiquadric form, are effective for approximating functions that behave on $[0, \infty)$ like x^α for small x and like x^β for large x , where α, β are known and finite. Numerical methods, based on weighted least squares, are adopted for the same selection of (nonlinear) ordinary differential equation (ODE) solutions as that considered by rational approximation in [7] (namely the Thomas-Fermi equation, the Blasius equation and Dawson’s integral), and RBF sums perform with similar, if slightly less accurate, versatility. Accuracy of 2 to 4 decimals, by comparison with known solutions, is readily achievable, without the need to adopt high degrees in the basis.

1 Introduction

A special method of nonlinear approximation for a function $y = f(x)$, $x \in \mathcal{R}$ has the form

$$y = f(x) \approx g(L(x)), \quad (1)$$

where g is a given, fixed 1-1 function and $L(x)$ is a linear form, in this case an RBF such as the cubic sum

$$L(x) = \sum_{j=1}^n c_j |x - \lambda_j|^3,$$

and where λ_j are chosen centres and c_j are coefficients to be determined. Then (1) can be rewritten as an equivalent (exact) equation by defining ϵ as the error:

$$f = g(L) + \epsilon, \text{ where } y = f(x), L = L(x), \epsilon = \epsilon(x). \quad (2)$$

Hence, from (2), writing $G(f) \equiv g^{-1}(f)$ and letting a dash denote a derivative with respect to f ,

$$L = g^{-1}(f - \epsilon) = G(f - \epsilon),$$

and, by Taylor series expansion,

$$L = G(f) - \epsilon G'(f) + \frac{\epsilon^2}{2} G''(f) + \dots$$

Then writing

$$w = \left[\frac{1}{G'(f)} \right] = \left[(g^{-1})'(f) \right]^{-1}, \quad (3)$$

gives

$$wL = wG(f) - \epsilon + \frac{\epsilon^2}{2} \frac{G''(f)}{G'(f)} + \dots \quad (4)$$

Here we are assuming that $g^{-1}(f)$ exists in some interval containing all data, $g(L)$ is differentiable with respect to L , and $H(f) \equiv G'(f)/G''(f)$ is bounded away from zero on some interval containing all the data.

Hence

$$w(x)L(x) \approx w(x)g^{-1}(f(x)), \quad (5)$$

so that (5) has an error of order $-\epsilon$ which, neglecting $\mathcal{O}(\epsilon^2)$, is proportional to that in (2). Then (5) is a weighted linear approximation which may be determined by solving in a least squares sense an $m \times n$ system of overdetermined linear equations $wL = wG(f)$ with $x = x_k$ ($k = 1, 2, \dots, m$). The weight function w has an alternative form equivalent to (3), namely

$$w = g'(g^{-1}(f)), \quad (6)$$

which can be verified by differentiating

$$g((g^{-1})(f)) = f,$$

with respect to f to give

$$[g'(g^{-1}(f))] [(g^{-1})'(f)] = 1.$$

Consider two specific examples as follows:

1. $g(L) = L^R$ (R real): $g'(L) = RL^{R-1}$,
 $G(f) = g^{-1}(f) = f^{\frac{1}{R}}$. Hence

$$w = g'(g^{-1}(f)) = R \left(f^{\frac{1}{R}}\right)^{R-1} = Rf^{\frac{R-1}{R}},$$

and by (4) we have

$$H(f) = \frac{G'(f)}{G''(f)} = \frac{Rf}{(1-R)}, \text{ and } wL = Rf - \epsilon + \mathcal{O}\left(\frac{\epsilon^2}{f}\right),$$

Here the three assumptions after (4) are verifiable on an appropriate interval for $R > 0$. In the case $R < 0$ we would normally require for the validity of the method that f should be bounded away from zero. However this is not the case if, for example, $f(x) \equiv \exp(-x) \approx L^{-4}$ on the range $[0, \infty)$. Nevertheless the method appears to work very well in practice in this case (see Mason [7]).

2. $g(L) = \exp(L)$: $g'(L) = \exp(L)$, $G(f) = g^{-1}(f) = \log(f)$. Hence

$$w = \left[(g^{-1})'(f)\right]^{-1} = f.$$

Also $H(f) = G'(f)/G''(f) = f^{-1}/(-f^{-2}) = -f$.

For $L = g^{-1}(f - \epsilon) = \log(f) + \log(1 - \epsilon/f) \approx \log(f) - \epsilon/f + \mathcal{O}(\epsilon^2/f^2)$.

Now $w = f$, so $fL \approx f \log(f) - \epsilon + \mathcal{O}(\epsilon^2/f)$.

Again the first two assumptions after (4) may readily be checked. In this case we require again that f should be bounded away from zero.

2 Special End Point Behaviour

Consider a many times differentiable function $f(x)$ defined on $[0, \infty)$ which has the end point (or asymptotic) behaviour

$$y \sim x^\alpha \text{ at } x = 0, y \sim x^\beta \text{ as } x \rightarrow \infty. \quad (7)$$

Then we can often find a form of approximation with the same behaviour, and we give three classical ODEs (see Mason [7]) with such behaviour, namely

1. Thomas-Fermi equation,
2. Dawson's integral,
3. Blasius equation.

This leads us in each case to a $g(L)$ closely related to L^R , where L is a sum of radial basis functions of the cubic form, namely

$$f \sim x^\alpha (L(x))^R,$$

In his thesis [7], Mason adopted rational approximations of the form

$$f \sim x^\alpha \left(\frac{A_p(x)}{B_q(x)} \right)^R,$$

where A_p, B_q are polynomials of degree p, q respectively. Here we use the shorthand notation $A_p(x)$ to denote $a_0 + a_1x + a_2x^2 + \dots + a_px^p$, namely a polynomial of degree p in x . From (7) it follows asymptotically that

$$x^\beta = x^\alpha (a_px^pb_q^{-1}x^{-q})^R \text{ and hence that } R = \frac{\beta - \alpha}{p - q}.$$

We find that closely comparable accuracies are achieved for functions g of both rational functions and RBFs.

3 Summary of Previous Contributions

The fitting of end conditions (7) is demonstrated by Mason ([7] 1965), and the case of $f \approx g(L)$ (approximation by function of a linear form) was introduced for L^R by Appel ([1] 1962), extended to minimax norms by Carta ([2] 1978), improved by Mason and Upton ([8] 1989) and analysed by Crampton et al. ([3] 2004). In this paper we focus on the Appel algorithm and we do not need to adopt the linear iteration algorithm of Mason and Upton ([8] 1989).

Applications to ODEs were introduced by Mason ([7] 1965) and published by Ziegler ([9] 1981), and existence theory for best approximation by L^R was discussed by Dunham and Williams ([5] 1981). For a degree $2p + 1$ RBF,

$$f \sim x^\alpha \left[\sum_{j=1}^n c_j |x - \lambda_j|^{2p+1} \right]^R \sim x^\beta \text{ as } x \rightarrow \infty,$$

and hence $R = (\beta - \alpha) / (2p + 1)$.

4 Nonlinear ODEs With Known Solutions and Behaviour

4.1 Thomas-Fermi Equation

This equation defines the ‘‘ordinary Thomas-Fermi function’’ $y = f(x)$ (see Mason [7]) by

$$x(y'')^2 = y^3, \quad y(0) = 1, \quad y'(\infty) = 0,$$

and has $y \sim x^0, x^{-3}$ as $x = 0, x \rightarrow \infty$, respectively and, setting $t = x^{\frac{1}{2}}$

$$y \sim t^0, t^{-6} \text{ as } t = 0, t \rightarrow \infty.$$

Thus it is clear, with the variable $t = x^{\frac{1}{2}}$ in place of x , that

$$\alpha = 0, \quad \beta = -6.$$

More precisely, Kobayashi et al. [6] determine $y'(0)$ ($= A$ say) as $y'(0) = A = -1.5880710\dots$ They also show that, for some d_3, d_4, \dots

- i) $y \sim 1 + d_2t^2 + d_3t^3 + \dots$ ($d_2 = A$) as $t \rightarrow 0$
- ii) $y \sim 144x^{-3} = 144t^{-6}$ as $t \rightarrow \infty$.

Function of a Cubic RBF

Choose

$$y = f(x) \approx g(L(t^2)) = \left(\sum_{j=1}^n c_j |t^2 - \lambda_j|^3 \right)^R = L^R \text{ where } L \sim t^0, t^3,$$

as $t \rightarrow 0, \infty$. Hence

$$\alpha = 0, \quad \beta = -6 = 3R, \text{ and thus } R = -2.$$

Rational Function

For a rational function approximation

$$f \approx F = \left(\frac{A_p(t)}{B_q(t)} \right)^R, \tag{8}$$

and choosing $q = p + 3$ we obtain as $t \rightarrow \infty$,

$$f \sim t^{-3R} \sim t^{-6}.$$

Hence $R = 2, \alpha = 0$.

4.2 Dawson's Integral

Here (see Davis [4])

$$y' + 2xy - 1 = 0, \quad y(0) = 0,$$

and

$$y \sim x, x^{-1} \text{ as } x \rightarrow 0, \infty,$$

i.e.,

$$\alpha = 1, \beta = -1.$$

Function of a Cubic RBF

A function of a cubic RBF $g(L(x))$ gives

$$y = f(x) \approx g(L(x)) = x(L)^R,$$

where L is a cubic RBF,

$$y \sim x, x^{-1} \text{ and hence } \alpha = 1, \beta = -1, \text{ as } x \rightarrow 0, \infty.$$

Now

$$\alpha = 1, \beta = 1 + 3R = -1, \text{ and thus } R = -\frac{2}{3}.$$

Rational Function

Here we choose

$$y \approx \left(x \left(\frac{A_p(x^2)}{B_{p+1}(x^2)} \right) \right)^R \sim (xCx^{-2})^R = (Cx^{-1})^R \text{ as } x \rightarrow \infty, \quad (9)$$

where

$$y \sim x, x^{-1} \text{ as } x \rightarrow 0, \infty.$$

Hence

$$\alpha = 1, \beta = -1, R = 1.$$

4.3 Blasius Equation

Here (see Davis [4]) we consider

$$y''' + yy'' = 0, y(0) = y'(0) = 0, y'(\infty) = 2,$$

and

$$y \sim x^2, x \text{ as } x \rightarrow 0, \infty.$$

Function of a Cubic RBF

Here

$$y \approx F = x^2 [R_3(x)]^R,$$

and

$$y \sim (x^2, x^{2+3R}) = (x^2, x) \text{ for } R = -\frac{1}{3}, \text{ at } x = (0, \infty).$$

Rational Function

There is an expansion for y in powers x^2, x^5, \dots , namely

$$y \sim x^2 (e_0 + e_1 x^3 + e_2 x^6 + \dots),$$

and

$$y \sim x^2, x \text{ as } x \rightarrow 0, \infty.$$

Thus a natural approximation is

$$y \approx x^2 \left[\frac{A_p(x^s)}{B_q(x^s)} \right]^R, \text{ with } \beta = 1 = 2 + s(p - q)R. \tag{10}$$

Choosing $q = p + 1$ gives

$$\beta = 1 = 2 + s(-1)R \Rightarrow R = \frac{1}{s}.$$

For example, $s = 3$ gives $R = 1/3$ and $s = 1$ gives $R = 1$.

5 Numerical Examples

It should be emphasised that, in all the examples considered here, we are not “solving” ODEs but rather we are fitting data taken from known solutions of ODEs. Let R_3 denote a cubic RBF with centres $\{\lambda_j\}_{j=1}^{10}$.

5.1 Ordinary Thomas-Fermi Function

The following data are given by Kobayashi et al. [6].

Choose $m = 20$, and, for $i = 1, 2, \dots, 20$, choose:

x_i : .05, .1, .2, .45, .7, .95, 1.2, 1.6, 2.1, 2.6, 3.2, 4.2, 5.5, 8, 11, 16, 20, 60, 200, 700.

y_i : .93519, .88170, .79306, .63233, .52079, .43806, .37424, .29810, .23159, .18480, .14482, .10136, .68160e-1, .36587e-1, .20250e-1, .94241e-2, .57849e-2, .39391e-3, .14502e-4, .38618e-6.

Function of a Cubic RBF: $y \approx \left[R_3 \left(x^{\frac{1}{2}} \right) \right]^{-2}$

Here,

$$t_i = x_i^{\frac{1}{2}}, \lambda_j = t_{2j-1},$$

and so there are 10 centres for 20 data. In the absolute fit, w is defined in (3), and (6), and in the relative fit w is then divided by y .

Table 1. Thomas-Fermi approximation errors for function of a cubic RBF.

	Abs fit	Rel fit
Absolute error: $\ \epsilon\ _\infty$	0.05011	0.00424
Relative error: $\ \epsilon/y\ _\infty$	0.76	0.00137
$A \equiv (\sum c_j)^R$	23.77	163.7

Rational Function: $y \approx \left[A_p \left(x^{\frac{1}{2}} \right) / B_{p+3} \left(x^{\frac{1}{2}} \right) \right]^2$

Based on form (8), it is found that $\|\epsilon\|_\infty \approx 0.0001$ for $p = 3$, $m = 20$ and $n = 2p + 4 = 10$ coefficients $\{c_j\}_{j=1}^{10}$. Note that $\|\epsilon\|_\infty$ is here taken over m data, giving

$$\max_i |\epsilon_i|.$$

It is clear in the computation in Table 1 that a function g of a cubic RBF is effective, by comparison with a rational function, as a form of approximation to f . Both forms produce about the same size of error for a comparable computing task.

5.2 Dawson’s Integral

Solution data are provided by Davis [4]. We specify:
 x_i : .5, 1, 2, 2.5, 3, 3.5, 4, 5, 6, 10.
 y_i : .42444, .53808, .30134, .22308, .17827, .14962, .12935, .10213, .84543e-1, .50254e-1.

Function of a Cubic RBF: $y \approx x [R_3(x)]^{-\frac{2}{3}}$

Here $m = 10$ and $n = 5$ for the results shown in Table 2.

Table 2. Dawson’s integral approximation errors for function of a cubic RBF.

	Abs fit	Rel fit
Absolute error: $\ \epsilon\ _\infty$	0.00272	0.00537
Relative error: $\ \epsilon/y\ _\infty$	0.158	0.001144
$(\sum c_j)^R$	272	649

Rational Function: $y \approx [xA_p(x^2) / B_{p+1}(x^2)]$

This corresponds to (9) and we found that $\|\epsilon\|_\infty \approx 0.003$ for $p = 2$, $m = 10$ and $n = 5$. From Table 2 it is clear that all approximations, both rational and RBF, are comparable and effective.

5.3 Blasius Equation

Solution data are given by Davis as follows:

x_i : .2, .4, .8, 1.2, 1.4, 4, 4.2, 4.4, 5, 10, 100, 1000.

y_i : .026560, .10611, .42032, .92230, 3.08535, 6.27923, 6.67923, 7.07923, 8.27923, 18.27923, 198.27923, 1998.27923.

Function of a Cubic RBF: $y \approx F = x^2 [R_3(x)]^{-\frac{1}{3}}$

Table 3. Blasius equation approximation errors for function of a cubic RBF.

	Abs fit	Rel fit
Absolute error: $\ \epsilon\ _\infty$	0.013	0.0019
Relative error: $\ \epsilon/y\ _\infty$	0.0016	0.0033
$F'(\infty)$	2.0022	1.999997
$(F - 2x)(\infty)$	-2.1664	-1.717420

Here $m = 12$, $n = 6$. Especially good results are obtained (see Table 3) for a relative fit, with $y'(\infty)$ correct to 5 decimal places and $y \approx 2x = -1.72$ correct to 1-2 decimal places as $x \rightarrow \infty$.

Rational Function: $y \approx [x^2 A_p(x) / B_{p+1}(x)]$

This corresponds to (10) with $R = 1$, $s = 1$. Here $\|\epsilon\|_\infty \approx 0.002$ for $p = 2$, $m = 12$ and $n = 6$. Again both RBFs and rational functions are very effective and comparably accurate.

6 Conclusions

For three numerical examples given in Section 5, corresponding to known solutions of different differential equations, both RBF and rational approximations are very effective and of a comparable accuracy when a suitable number of parameters are adopted and a suitable form is chosen. It is shown to be advantageous to choose a form of approximation which matches the data at $x = 0$, $x \rightarrow \infty$.

References

1. K. Appel: *Rational Approximation of Decay-Type Functions*, BIT: Volume 2(2), 1962, 69–75.
2. D.G. Carta: *Minimax Approximation by Rational Functions of the Inverse Polynomial*, BIT: Volume 18, 1978, 490–492.
3. A. Crampton, D.P. Jenkinson, and J.C. Mason: *Iteratively Weighted Approximation Algorithms For Nonlinear Problems using Radial Basis Function Examples*, Applied Numerical Analysis and Computational Mathematics: Volume 2(1), T.E.. Simos, G. Psihoyios, and E.R. Simon (eds.), WILEY-VCH Verlag, Weinheim, 2004, 165–179.
4. H.T. Davis: *Introduction to Nonlinear Differential and Integral Equations*, U. S. Atomic Energy Commission, 1960.
5. C.B. Dunham and J. Williams: *Rate of Convergence of Discretization in Chebyshev Approximation*, Mathematics of Computation: Volume 37(155), 1981, 135–139.
6. S. Koboyashi, T. Matsukma, S. Nagai, and K. Umeda: *Accurate Value of the Initial Slope of the Ordinary Thomas-Fermi Function*, Physical Society of Japan: Volume 10(9), 1955, 759–762.
7. J.C. Mason: *Some new approximations for the solution of differential equations*, D. Phil Thesis, Oxford University, 1965.
8. J.C. Mason, and N.K. Upton: *Linear Algorithms for Transformed Linear Forms*, Approximation Theory VI: Volume 2, C.K. Chui, L.L. Schumaker and J.D. Ward (eds.), Academic Press, Inc, 1989, 417–420.
9. Z. Ziegler (ed.): *Approximation Theory and Applications*, Academic Press, New York (contains: J.C. Mason Some applications and drawbacks of Padé approximants), 1981, 207–223.

7 Appendix (Blasius Equation)

An approximation of high accuracy may be obtained of the rational function form

$$y = f(x) \approx F = 2x - 1.72077 + 1.72077 (B_q(x))^R, \quad (11)$$

where $B_q = 1 + b_1x + \dots + b_qx^q$ and R is a negative integer. This is a rational function as well as a function of a linear form and for $q = 12$ and $R = -4$ an absolute accuracy of 10^{-5} can be achieved. Full details are given in Mason [7] and Ziegler [9].

Similarly a cubic RBF may be adopted in place of B_q in (11) such as

$$y = f(x) \approx F(x) = 2x - 1.72077 + 1.72077 \left[\sum_{j=1}^n c_j |x - \lambda_j|^3 \right]^R, \quad (12)$$

where R is a negative integer such as $R = -4$. Both forms (11) and (12) reproduce the first two forms of the dominant behaviour $y \sim Ax + b$ as $x \rightarrow \infty$, and a rapidly decaying correction is then determined, to be added to $Ax + b$.

Weighted Integrals of Polynomial Splines

Mladen Rogina

Department of Mathematics, University of Zagreb, 10002 Zagreb, Croatia,
rogina@math.hr

Summary. The construction of weighted splines by knot insertion techniques such as de Boor and Oslo - type algorithms leads immediately to the problem of evaluating integrals of polynomial splines with respect to the positive measure possessing piecewise constant density. It is for such purposes that we consider one possible way for simple and fast evaluation of primitives of products of a polynomial B-spline and a positive piecewise constant function.

1 Introduction and Motivation

Weighted splines appear in many applications, the most well-known being the cubic version where they arise naturally in minimizing functionals like $V(f) := \sum_{i=1}^n (w_i \int_{t_i}^{t_{i+1}} [D^2 f(t)]^2 dt)$, $w_i > 0$, sometimes also accompanied by the control of first derivatives: $V(f) := \sum_{i=1}^n (w_i \int_{t_i}^{t_{i+1}} [D^2 f(t)]^2 dt + \nu_i \int_{t_i}^{t_{i+1}} [Df(t)]^2 dt)$, $\nu_i \geq 0$, $w_i > 0$, see [6, 7, 9] and [11] for a bivariate version.

The parametric version is often used as a polynomial alternative to the exponential tension spline in computer-aided geometric design, and some shape-preserving software systems (MONCON, TRANSPLINE) have been written for that purpose [13, 9, 10]. It is known that the associated B-splines can be calculated by the knot insertion algorithms. For the cubic version of weighted splines, explicit expressions for the knot insertion matrices exist, which are of the very simple form [8, 14]. In the case of the knot insertion algorithms can in principle be obtained by specializing the general theory of Chebyshev blossoming [12].

Weighted splines can also be evaluated by an integrated version of the derivative formula [15], which can also be used to define most general Chebyshev B-splines [1]:

$$B_{i,d\sigma}^n(x) = \frac{1}{C_{n-1}(i)} \int_{t_i}^x B_{i,d\sigma^{(1)}}^{n-1} d\sigma_2 - \frac{1}{C_{n-1}(i+1)} \int_{t_{i+1}}^x B_{i+1,d\sigma^{(1)}}^{n-1} d\sigma_2, \quad (1)$$

where $B_{i,d\sigma}^n(x)$ is the n^{th} -order Chebyshev spline, $d\sigma = (d\sigma_2 \dots d\sigma_n)^T$ is the measure vector and $d\sigma^{(1)} = (d\sigma_3 \dots d\sigma_n)^T$ is the measure vector with respect to the first reduced system. We assume that $d\sigma_i$ are some Stieltjes measures, and that all

the B-splines in question are normalized so as to make a partition of unity. The constants in the denominators are integrals of B-splines over its support, with respect to the measure that is missing in the definition of $d\sigma^{(1)}$:

$$C_{n-1}(i) := \int_{t_i}^{t_{i+n-1}} B_{i,d\sigma^{(1)}}^{n-1} d\sigma_2.$$

The numerical stability of (1) is doubtful (even for polynomial splines), so evaluation by knot insertion is preferred. However, for weighted splines we need only very simple measures, which are all but one Lebesgue measures, and the one that is not has density which is piecewise constant and positive. To be more precise, weighted B-splines are piecewisely spanned by the Chebyshev system of *weighted powers*:

$$\begin{aligned} u_1(x) &= 1, \\ u_2(x) &= \int_a^x d\tau_2, \\ u_3(x) &= \int_a^x d\tau_2 \int_a^{\tau_2} \frac{d\tau_3}{w(\tau_3)}, \\ &\vdots \\ u_k(x) &= \int_a^x d\tau_2 \int_a^{\tau_2} \frac{d\tau_3}{w(\tau_3)} \int_a^{\tau_3} d\tau_4 \cdots \int_a^{\tau_{k-1}} d\tau_k. \end{aligned}$$

Finally, one can use algorithms for ordinary polynomial splines and avoid explicit mentioning of weighted splines, but even then integration of products of polynomial splines and piecewise constant function must be performed, as shown by de Boor [3], who also gives closed formulæ for some lower order splines.

2 Recurrence for Integrals of Polynomial B-Splines

Whatever approach we choose, in order to evaluate weighted splines we need to calculate the integrals of ordinary polynomial B-splines

$$C_k(j) = \int_{t_j}^{t_{j+k}} B_j^k(\tau) \frac{d\tau}{w(\tau)}.$$

In what follows, we assume that B_j^k are normalized so as to make the partition of unity, and that the knot sequence $\{t_j\}$, possibly containing multiple knots, coincides with the breakpoint sequence for w . For notation purposes, let $w|_{[t_i, t_{i+1})} = w_i$ which makes w right-continuous. We want to find a recurrence for primitives of polynomial B-splines with respect to the piecewise constant positive function w , i.e.,

$$\int_{t_i}^x B_i^k(\tau) \frac{d\tau}{w(\tau)}, \quad x \in [t_i, t_{i+k}],$$

and, specially:

$$\int_{t_j}^{t_{j+1}} B_i^k(\tau) \frac{d\tau}{w(\tau)}, \quad j = i, \dots, i + k - 1.$$

Let $x \in [t_j, t_{j+1})$, then

$$\begin{aligned}
 \int_{t_i}^x B_i^k(\tau) \frac{d\tau}{w(\tau)} &= \sum_{s=i}^{j-1} \int_{t_s}^{t_{s+1}} B_i^k(\tau) \frac{1}{w_s} d\tau + \frac{1}{w_j} \int_{t_j}^x B_i^k(\tau) d\tau \\
 &= \sum_{s=i}^{j-1} \frac{1}{w_s} \left(\int_{t_i}^{t_{s+1}} B_i^k(\tau) d\tau - \int_{t_i}^{t_s} B_i^k(\tau) d\tau \right) \\
 &\quad + \frac{1}{w_j} \left(\int_{t_i}^x B_i^k(\tau) d\tau - \int_{t_i}^{t_j} B_i^k(\tau) d\tau \right) \\
 &= \sum_{s=i}^{j-1} \frac{1}{w_s} \frac{t_{i+k} - t_i}{k} \left(\sum_{r=i}^s B_r^{k+1}(t_{s+1}) - \sum_{r=i}^{s-1} B_r^{k+1}(t_s) \right) \\
 &\quad + \frac{1}{w_j} \frac{t_{i+k} - t_i}{k} \left(\sum_{r=i}^j B_r^{k+1}(x) - \sum_{r=i}^{j-1} B_r^{k+1}(t_j) \right), \tag{2}
 \end{aligned}$$

by the well known formula for integrals of polynomial splines [16, p. 200] and [2, pp. 150-151]. Let

$$\bar{\alpha}_{i,j+1}^{k+1}(x) := \sum_{r=i}^j B_r^{k+1}(x) \quad \text{and} \quad \alpha_{i,j+1}^{k+1} := \bar{\alpha}_{i,j+1}^{k+1}(t_{j+1}). \tag{3}$$

Then in terms of $\bar{\alpha}$'s formula (2) can be written as

$$\int_{t_i}^x B_i^k(\tau) \frac{d\tau}{w(\tau)} = \frac{t_{i+k} - t_i}{k} \left(\sum_{s=i}^{j-1} \frac{1}{w_s} \left(\alpha_{i,s+1}^{k+1} - \alpha_{i,s}^{k+1} \right) + \frac{1}{w_j} \left(\bar{\alpha}_{i,j+1}^{k+1}(x) - \alpha_{i,j}^{k+1} \right) \right). \tag{4}$$

We claim that $\bar{\alpha}_{i,j+1}^{k+1}(x)$ can be evaluated as convex combination of lower order quantities $\bar{\alpha}_{i,j}^k(x)$. By de Boor–Cox recurrence

$$\begin{aligned}
 \sum_{r=i}^j B_r^{k+1}(x) &= \sum_{r=i}^j \left(\frac{x - t_r}{t_{r+k} - t_r} B_r^k(x) + \frac{t_{r+k+1} - x}{t_{r+k+1} - t_{r+1}} B_{r+1}^k(x) \right) \\
 &= \sum_{r=i}^j \frac{x - t_r}{t_{r+k} - t_r} B_r^k(x) + \sum_{r=i}^j B_{r+1}^k(x) - \sum_{r=i}^j \frac{x - t_{r+1}}{t_{r+k+1} - t_{r+1}} B_{r+1}^k(x) \\
 &= \sum_{r=i+1}^j \left(\frac{x - t_r}{t_{r+k} - t_r} - \frac{x - t_r}{t_{r+k} - t_r} \right) B_r^k(x) + \frac{x - t_i}{t_{i+k} - t_i} B_i^k(x) + \sum_{r=i}^{j-1} B_{r+1}^k(x) \\
 &= \frac{x - t_i}{t_{i+k} - t_i} B_i^k(x) + \sum_{r=i+1}^j B_r^k(x) = \frac{x - t_i}{t_{i+k} - t_i} B_i^k(x) + \bar{\alpha}_{i+1,j+1}^k(x),
 \end{aligned}$$

because $B_{j+1}^k(x) = 0$ for $x \in [t_j, t_{j+1})$. Thus we have proved the recurrence

$$\bar{\alpha}_{i,j+1}^{k+1}(x) = \frac{x - t_i}{t_{i+k} - t_i} B_i^k(x) + \bar{\alpha}_{i+1,j+1}^k(x), \tag{5}$$

for $x \in [t_j, t_{j+1})$ and $j = i, \dots, i + k - 1$. We proceed to manipulate (5) to get a more symmetric expression. Obviously,

$$\begin{aligned}
 \bar{\alpha}_{i,j+1}^k(x) &= \sum_{r=i}^j B_r^k(x) = B_i^k(x) + \sum_{r=i+1}^j B_r^k(x) \\
 &= B_i^k(x) + \bar{\alpha}_{i+1,j+1}^k(x),
 \end{aligned}$$

whence $B_i^k(x) = \bar{\alpha}_{i,j+1}^k(x) - \bar{\alpha}_{i+1,j+1}^k(x)$, which, when substituted in (5) gives

$$\begin{aligned} \bar{\alpha}_{i,j+1}^{k+1}(x) &= \frac{x - t_i}{t_{i+k} - t_i} \left(\bar{\alpha}_{i,j+1}^k(x) - \bar{\alpha}_{i+1,j+1}^k(x) \right) + \bar{\alpha}_{i+1,j+1}^k(x) \\ &= \frac{x - t_i}{t_{i+k} - t_i} \bar{\alpha}_{i,j+1}^k(x) + \bar{\alpha}_{i+1,j+1}^k(x) \left(1 - \frac{x - t_i}{t_{i+k} - t_i} \right). \end{aligned}$$

Finally, we have the recurrence

$$\bar{\alpha}_{i,j+1}^{k+1}(x) = \frac{x - t_i}{t_{i+k} - t_i} \bar{\alpha}_{i,j+1}^k(x) + \frac{t_{i+k} - x}{t_{i+k} - t_i} \bar{\alpha}_{i+1,j+1}^k(x), \tag{6}$$

for $x \in [t_j, t_{j+1})$ and $j = i, \dots, i + k - 1$.

We need to evaluate

$$\frac{1}{w_j} \frac{t_{i+k} - t_i}{k} \left(\sum_{r=i}^j B_r^{k+1}(x) - \sum_{r=i}^{j-1} B_r^{k+1}(t_j) \right) = \frac{t_{i+k} - t_i}{k w_j} \left(\bar{\alpha}_{i,j+1}^{k+1}(x) - \alpha_{i,j}^{k+1} \right),$$

but have no way of telling whether the subtraction of $\bar{\alpha}$'s will result in dangerous cancellation of significant digits; therefore we must find another way of evaluating differences of $\bar{\alpha}$'s. To this end, let

$$\bar{\delta}_{i,j}^{k+1}(x) := \bar{\alpha}_{i,j+1}^{k+1}(x) - \alpha_{i,j}^{k+1}.$$

From (6) we have

$$\begin{aligned} \bar{\delta}_{i,j}^{k+1}(x) &= \frac{x - t_i}{t_{i+k} - t_i} \bar{\alpha}_{i,j+1}^k(x) + \frac{t_{i+k} - x}{t_{i+k} - t_i} \bar{\alpha}_{i+1,j+1}^k(x) - \frac{t_j - t_i}{t_{i+k} - t_i} \alpha_{i,j}^k - \frac{t_{i+k} - t_j}{t_{i+k} - t_i} \alpha_{i+1,j}^k \\ &= \frac{t_j - t_i}{t_{i+k} - t_i} \bar{\delta}_{i,j}^k(x) + \frac{t_{i+k} - x}{t_{i+k} - t_i} \bar{\delta}_{i+1,j}^k(x) + \frac{x - t_j}{t_{i+k} - t_i} \left(\bar{\alpha}_{i,j+1}^k(x) - \alpha_{i+1,j}^k \right). \end{aligned} \tag{7}$$

Further,

$$\begin{aligned} \bar{\alpha}_{i,j+1}^k(x) - \alpha_{i+1,j}^k &= \bar{\alpha}_{i,j+1}^k(x) - \bar{\alpha}_{i+1,j+1}^k(x) + \bar{\alpha}_{i+1,j+1}^k(x) - \alpha_{i+1,j}^k \\ &= \bar{\alpha}_{i,j+1}^k(x) - \bar{\alpha}_{i+1,j+1}^k(x) + \bar{\delta}_{i+1,j}^k(x) \\ &= \sum_{r=i}^j B_r^k(x) - \sum_{r=i+1}^j B_r^k(x) + \bar{\delta}_{i+1,j}^k(x) \\ &= B_i^k(x) + \bar{\delta}_{i+1,j}^k(x), \end{aligned} \tag{8}$$

where the last line follows from the defining equation (3) for $\bar{\delta}_{i+1,j}^k(x)$. On substituting (8) in (7) we get

$$\bar{\delta}_{i,j}^{k+1}(x) = \frac{t_j - t_i}{t_{i+k} - t_i} \bar{\delta}_{i,j}^k(x) + \frac{t_{i+k} - t_j}{t_{i+k} - t_i} \bar{\delta}_{i+1,j}^k(x) + \frac{x - t_j}{t_{i+k} - t_i} B_i^k(x),$$

for $x \in [t_j, t_{j+1})$ and $j = i, \dots, i + k - 1$. Finally, from (4) we have

$$\frac{k}{t_{i+k} - t_i} \int_{t_i}^x B_i^k(\tau) \frac{d\tau}{w(\tau)} = \sum_{s=i}^{j-1} \frac{\delta_{i,s}^{k+1}}{w_s} + \frac{1}{w_j} \bar{\delta}_{i,j}^{k+1}(x), \tag{9}$$

with

$$\delta_{i,s}^{k+1} := \bar{\delta}_{i,s}^{k+1}(t_{s+1}),$$

$x \in [t_j, t_{j+1})$ and $j = i, \dots, i + k - 1$. Specially,

$$\frac{k}{t_{i+k} - t_i} \int_{t_i}^{t_{i+k}} B_i^k(\tau) \frac{d\tau}{w(\tau)} = \sum_{s=i}^{i+k-1} \frac{\delta_{i,s}^{k+1}}{w_s},$$

and by (9)

$$\frac{k}{t_{i+k} - t_i} \int_{t_j}^{t_{j+1}} B_i^k(\tau) d\tau = w_j \left(\int_{t_i}^{t_{j+1}} B_i^k(\tau) \frac{d\tau}{w(\tau)} - \int_{t_i}^{t_j} B_i^k(\tau) \frac{d\tau}{w(\tau)} \right) = \delta_{i,j}^{k+1},$$

where $\delta_{i,j}^{k+1}$ is calculated recursively:

$$\begin{aligned} \delta_{i,j}^2 &= \begin{cases} 1 & \text{for } j = i, \\ 0 & \text{for } j \neq i, \end{cases} \\ \delta_{i,j}^{k+1} &= \frac{t_j - t_i}{t_{i+k} - t_i} \delta_{i,j}^k + \frac{t_{i+k} - t_j}{t_{i+k} - t_i} \delta_{i+1,j}^k + \frac{t_{j+1} - t_j}{t_{i+k} - t_i} B_i^k(t_{j+1}), \end{aligned} \tag{10}$$

for $j = i, \dots, i + k - 1$.

3 Conclusion

There are other ways of calculating weighted integrals of polynomial splines, like Gaussian integration or conversion to Bezier form, and also some approximative ones [17]. In fact, (10) is a special case of recurrence used to evaluate inner products of B-splines ([4]) in which one of the B-splines is of order one. The proof given here is more in the spirit of ‘B-splines without divided differences’ [5], contains some new recurrences (5), and can be extended to obtain a recurrence for inner products. For inner products though, the greater complexity ($O(k^4)$) compared to Gaussian integration ($O(k^3)$) makes the recurrence seldom used, while for weighted splines it is preferable, being of the same complexity and machine independent.

Acknowledgement

This research was supported by grant 0037114, by the Ministry of Science, Education and Sports of the Republic of Croatia.

References

1. D. Bister and H. Prautzsch: A new approach to Tchebycheffian B-splines. In: *Curves and Surfaces with Applications in CAGD*, A. Le Méhauté, C. Rabut, L.L. Schumaker (eds.), Vanderbilt University Press, Nashville, 1997, 387–394.
2. C. de Boor: *A Practical Guide to Splines*. Springer, New York, 1978.

3. C. de Boor: Calculation of the smoothing spline with weighted roughness measure. *Math. Models Methods Appl. Sci.* **11**(1), 2001, 33–41.
4. C. de Boor, T. Lyche and L.L. Schumaker: On calculating with B-splines II: integration. In: *Numerische Methoden der Approximationstheorie*, L. Collatz, H. Werner and G. Meinardus (eds.), Birkhäuser, Basel, 1976, 123–146.
5. C. de Boor and K. Höllig: B-splines without divided differences. In: *Geometric Modeling*, G. Farin (ed.), SIAM, Philadelphia, 1987, 21–27.
6. L. Bos and K. Salkauskas: Weighted splines based on piecewise polynomial weight functions. In: *Curve and Surface Design*, H. Hagen (ed.), SIAM, Philadelphia, 1999, 87–98.
7. L. Bos and K. Salkauskas: Limits of weighted splines based on piecewise constant weight functions. *Rocky Mountain J. Math.* **23**, 1993, 483–493.
8. T. Bosner: Knot insertion algorithms for weighted splines. In: *Proceedings of the Conference on Applied Mathematics and Scientific Computing*, Z. Drmač, M. Marušić and Z. Tutek (eds.), Springer, 2005, 151–160.
9. T.A. Foley: Interpolation with interval and point tension controls using cubic weighted ν -splines. *ACM Trans. Math. Software* **13**(1), 1987, 68–96.
10. T.A. Foley: Local control of interval tension using weighted splines. *CAGD* **3**, 1986, 281–294.
11. T.A. Foley: Weighted bicubic spline interpolation to rapidly varying data. *ACM Trans. on Graphics* **6**, 1987, 1–18.
12. M.-L. Mazure: Blossoming: a geometrical approach. *Constr. Approx.* **15**, 1999, 33–68.
13. S. Pruess: Alternatives to the exponential spline in tension. *Math. Comp.* **33**, 1979, 1273–1281.
14. M. Rogina: A knot insertion algorithm for weighted cubic splines. In: *Curves and Surfaces with Applications in CAGD*, A. Le Méhauté, C. Rabut, L.L. Schumaker (eds.), Vanderbilt University Press, Nashville, 1997, 387–394.
15. M. Rogina: Algebraic proof of the B-spline derivative formula. In: *Proceedings of the Conference on Applied Mathematics and Scientific Computing*, Z. Drmač, M. Marušić, Z. Tutek (eds.), Springer, 2005, 273–281.
16. L.L. Schumaker: *Spline Functions: Basic Theory*. John Wiley & Sons, New York, 1981.
17. A.H. Vermeulen, R.H. Bartels, and G.R. Heppler: Integrating products of B-splines. *SIAM J. Sci. Stat. Comput.* **13**(4), 1992, 1025–1038.

Part V

Differential and Integral Equations

On Sequential Estimators for Affine Stochastic Delay Differential Equations

Uwe Küchler¹ and Vyacheslav Vasiliev²

¹ Institute of Mathematics, Humboldt University Berlin, D-10099 Berlin, Germany, kuechler@mathematik.hu-berlin.de

² Department of Applied Mathematics and Cybernetics, University of Tomsk, 634050 Tomsk, Russia, vas@mail.tsu.ru

Summary. This paper presents a sequential estimation procedure for two dynamic parameters in affine stochastic differential equations with one time delayed term. The estimation procedure is based on the least square method with weights and yields estimators with guaranteed accuracy in the sense of the L_q -norm ($q \geq 2$). The proposed procedures work for all values of the parameters from \mathbb{R}^2 outside of some lines. The asymptotic behavior of the duration of the observations is investigated. It is shown, that the proposed method can be applied to affine stochastic differential equations with p time delayed terms.

1 Preliminaries

Affine and more general stochastic differential equations with time delay are used to model phenomena in economics, biology, technics and other sciences incorporating time delay. Often one has to estimate underlying parameters of the model from the observations of the running process.

Consider the stochastic differential equation with time delay given by

$$dX(t) = \sum_{i=0}^p \vartheta_i X(t - r_i) dt + dW(t), \quad t \geq 0, \quad (1)$$

$$X(s) = X_0(s), \quad s \in [-r, 0]. \quad (2)$$

Here $(W(t), t \geq 0)$ denotes a realvalued standard Wiener process on some probability space (Ω, \mathcal{F}, P) with respect to a filtration $\mathcal{F} = (\mathcal{F}_t, t \geq 0)$ from \mathcal{F} . The parameters $r_i, \vartheta_i, i = 0, \dots, p$, are real numbers with $0 = r_0 < r_1 < \dots < r_p =: r$. The initial process $(X_0(s), s \in [-r, 0])$ also defined on (Ω, \mathcal{F}, P) is supposed to be cadlag, \mathcal{F}_0 -measurable and satisfies

$$E \int_{-r}^0 X_0^2(s) ds < \infty.$$

Such differential equations with time delayed terms appear in different sciences, see e.g. [3, 8].

The problem consists in estimating the parameters $\vartheta = (\vartheta_i, i = 0, \dots, p)$ in a sequential way, based on continuous observation of $X(\cdot)$. The $(r_i, i = 0, \dots, p)$ are assumed to be known. The estimation of unknown time delays r_i demands other techniques and will be treated in forthcoming papers. See also [4] for first corresponding results.

It is well-known that (1) has a uniquely determined solution $(X(t), t \geq -r)$ which admits the representation

$$\left. \begin{aligned} X(t) &= \sum_{j=1}^p \vartheta_j \int_{-r_j}^0 x_0(t-s-r_j)X_0(s)ds \\ &\quad + x_0(t)X_0(0) + \int_0^t x_0(t-s)dW(s), \quad t > 0 \\ X(t) &= X_0(t), \quad t \in [-r, 0] \end{aligned} \right\} \quad (3)$$

and satisfies $E_\vartheta \int_0^T X^2(s)ds < \infty$ for every T with $0 < T < \infty$ (see e.g. [1, 2]). Here the function $x_0(\cdot)$ denotes the fundamental solution of the corresponding to (1),(2) deterministic linear equation

$$\begin{aligned} x_0(t) &= 1 + \sum_{j=0}^p \int_0^t \vartheta_j x_0(s-r_j)ds, \quad t \geq 0, \\ x_0(s) &= 0, \quad s \in [-r, 0]. \end{aligned}$$

Following [1], we can find a real γ and for every i with $0 \leq i \leq p$ a certain $\xi_i \geq 0$, polynomials $Q_i(\cdot)$ and $R_i(\cdot)$ and a certain v_i with $\gamma < v_p \leq v_{p-1} \leq \dots \leq v_0$ such that the following holds

$$x_0(t) = \sum_{i=0}^p (Q_i(t) \cos \xi_i t + R_i(t) \sin \xi_i t) e^{v_i t} + o(e^{\gamma t}) \quad \text{as } t \rightarrow \infty. \quad (4)$$

In this paper for the sake of simplicity we shall restrict ourselves to the case $p = 1$. The general case can be treated analogously, see the above remarks.

In this paper we shall construct a sequential estimator for the parameter $\vartheta = (\vartheta_0, \vartheta_1)'$ from observation of the process which satisfies

$$dX(t) = \vartheta_0 X(t)dt + \vartheta_1 X(t-1)dt + dW(t), \quad t \geq 0, \quad (5)$$

with the initial conditions

$$X(t) = X_0(t), \quad t \in [-1, 0]. \quad (6)$$

The asymptotic properties of the maximum likelihood estimators (MLE's) of the unknown parameter $\vartheta = (\vartheta_0, \vartheta_1)'$ have been investigated in [1].

Sequential parameter estimation problems for the drift of diffusions with time delay have been studied e.g. in [5, 6].

We assume that the parameter ϑ belongs to some fixed $\Theta \subset \mathbb{R}^2$ which will be specified below and we shall construct a sequential estimator for ϑ having a preassigned accuracy in the sense of the L_q -norm, which will be defined also below. To construct Θ we introduce the following notations, see [1] for details.

Let $s = u(r)$ ($r < 1$) and $s = w(r)$ ($r \in \mathbb{R}^1$) be the functions given by the parametric representation $(r(\xi), s(\xi))$ in \mathbb{R}^2 :

$$r(\xi) = \xi \cot \xi, \quad s(\xi) = -\xi / \sin \xi$$

with $\xi \in (0, \pi)$ and $\xi \in (\pi, 2\pi)$, respectively.

Consider the set Λ of all (real or complex) roots of the characteristic equation corresponding to (5)

$$\lambda - \vartheta_0 - \vartheta_1 e^{-\lambda} = 0$$

and put

$$v_0 = v_0(\vartheta) = \max\{\Re(\lambda) | \lambda \in \Lambda\},$$

$$v_1 = v_1(\vartheta) = \max\{\Re(\lambda) | \lambda \in \Lambda, \Re(\lambda) < v_0\}.$$

It can easily be shown that $-\infty < v_1 < v_0 < \infty$. By $m(\lambda)$ we denote the multiplicity of the solution $\lambda \in \Lambda$.

The estimation procedure will be constructed for all parameters ϑ from the set Θ defined by

$$\Theta = \Theta_1 \cup \Theta_2 \cup \Theta_3 \cup \Theta_4,$$

where

$$\Theta_1 = \{\vartheta \in \mathbb{R}^2 | v_0(\vartheta) < 0\}, \quad \Theta_2 = \Theta'_2 \cup \Theta''_2, \quad \Theta_3 = \Theta'_3 \cup \Theta''_3$$

with

$$\Theta'_2 = \{\vartheta \in \mathbb{R}^2 | v_0(\vartheta) > 0, v_1(\vartheta) > 0, m(v_0(\vartheta)) = 1, v_0(\vartheta) \in \Lambda \text{ and } v_1(\vartheta) \in \Lambda\},$$

$$\Theta''_2 = \{\vartheta \in \mathbb{R}^2 | v_0(\vartheta) > 0, v_1(\vartheta) > 0, m(v_0(\vartheta)) = 1, v_0(\vartheta) \in \Lambda \text{ and } v_1(\vartheta) \notin \Lambda\},$$

$$\Theta'_3 = \{\vartheta \in \mathbb{R}^2 | v_0(\vartheta) > 0 \text{ and } v_0(\vartheta) \notin \Lambda\},$$

$$\Theta''_3 = \{\vartheta \in \mathbb{R}^2 | v_0(\vartheta) > 0; v_0(\vartheta) \in \Lambda, m(v_0) = 2\},$$

$$\Theta_4 = \{\vartheta \in \mathbb{R}^2 | v_0(\vartheta) > 0, v_1(\vartheta) < 0, m(v_0(\vartheta)) = 1 \text{ and } v_0(\vartheta) \in \Lambda\}.$$

Note that this decomposition is very related to a classification used in [1], where can be found a figure, which helps to visualize these sets. In particular, Θ_1 is the set of all those ϑ , for which (5) admits a stationary solution.

In [5, 6] a more restricted region Θ was considered only.

2 Sequential Estimation Procedure

The sequential estimation procedures which will be constructed in the sequel base on the maximum likelihood estimator (MLE)

$$\hat{\vartheta}_{MLE}(S, T) = G_{XX}^{-1}(S, T)\Phi_{XX}(S, T), \quad G_{XX}(S, T) = \int_S^T \phi_{XX}(t)\phi'_{XX}(t)dt,$$

$$\phi_{XX}(t) = \begin{pmatrix} X(t) \\ X(t-1) \end{pmatrix}, \quad \Phi_{XX}(S, T) = \int_S^T \phi_{XX}(t)dX(t).$$

We shall put $F(T) = F(0, T)$ for all the functions $F(S, T)$, defined on the interval $[S, T]$, $0 \leq S < T$.

Denote by $\varphi_0(T)$ and $\varphi_1(T)$ the smallest and the largest eigenvalues of the information matrix $G_{XX}(T)$, respectively. According to [1] and [5, 6] their asymptotic behaviour for $T \rightarrow \infty$ is different for different parameters ϑ :

Region	$\varphi_0(T)$	$\varphi_1(T)$
Θ_1	T	T
Θ_2	e^{2v_1T}	e^{2v_0T}
Θ'_3	e^{2v_0T}	e^{2v_0T}
Θ''_3	$T^{-2}e^{2v_0T}$	$T^2e^{2v_0T}$
Θ_4	T	e^{2v_0T}

Define $\lambda = e^{v_0}$, $Y(t) = X(t) - \lambda X(t - 1)$ and $Z(t) = Y(t) - T^{-1}\lambda X(t - 1)$. Now we put $V(T) = I$ (2×2 identity matrix) in the cases Θ_1, Θ'_3 ,

$$V(T) = \begin{pmatrix} 1 & -\lambda \\ 1 & 0 \end{pmatrix},$$

in the cases Θ_2, Θ_4 and

$$V(T) = \begin{pmatrix} 1 & -(1 + T^{-1})\lambda \\ 1 & 0 \end{pmatrix}$$

in the case Θ''_3 . Moreover, we introduce the matrices

$$\begin{aligned} G_{YX}(S, T) &= \int_S^T \phi_{YX}(t, T) \phi'_{YX}(t, T) dt, & \phi_{YX}(t, T) &= V(T)\phi_{XX}(t), \\ \Phi_{YX}(S, T) &= \int_S^T \phi_{YX}(t, T) dX(t), & \zeta_{YX}(S, T) &= \int_S^T \phi_{YX}(t, T) dW(t), \\ \bar{\varphi}(T) &= \text{diag}\{\varphi_0(T), \varphi_1(T)\}, & \tilde{\Phi}_{YX}(S, T) &= \bar{\varphi}^{-\frac{1}{2}}(T)\Phi_{YX}(S, T), \\ \tilde{G}_{YX}(S, T) &= \bar{\varphi}^{-\frac{1}{2}}(T)G_{YX}(S, T)\varphi_0^{-\frac{1}{2}}(T), & \tilde{\zeta}_{YX}(S, T) &= \bar{\varphi}^{-\frac{1}{2}}(T)\zeta_{YX}(S, T). \end{aligned}$$

Using the introduced notations, the MLE $\hat{\vartheta}_{MLE}(S, T)$ can be written as

$$\hat{\vartheta}_{MLE}(S, T) = \varphi_0^{-\frac{1}{2}}(T)\tilde{G}_{YX}^{-1}(S, T)\tilde{\Phi}_{YX}(S, T). \tag{7}$$

It has the normed deviation

$$\varphi_0^{\frac{1}{2}}(T)(\hat{\vartheta}_{MLE}(S, T) - \vartheta) = \tilde{G}_{YX}^{-1}(S, T)\tilde{\zeta}_{YX}(S, T).$$

In [1] the representation (7) was used to investigate the properties of the MLE $\hat{\vartheta}_{MLE}(T)$, which are similar to the properties of $\hat{\vartheta}_{MLE}(S, T)$ under the condition $S = o(T)$ as $T \rightarrow \infty$. In particular it was shown, that the eigenvalues of the matrix $\tilde{G}_{YX}(T)$ have positive finite bounds for all T large enough in all the regions of parameters and the vector $\tilde{\zeta}_{YX}(T)$, introduced above has zero mean and bounded variance.

We cannot use the matrices $V(T)$, $\bar{\varphi}(T)$, $\tilde{G}_{YX}(S, T)$, and $\tilde{\Phi}_{YX}(S, T)$ in the construction of sequential estimators directly in view of their dependence from the unknown parameters $\alpha = (v_0, v_1)$. Therefore we shall use a modified version of the estimator $\hat{\vartheta}_{MLE}(S, T)$ from (7) being a weighted least squares estimator to construct appropriate sequential plans. And at the same time the first part of the observable process $(X(t), -1 \leq t \leq S)$ will be used for the estimation of the parameter α .

From (3), (4) it follows, that the eigenvalues of the information matrix of the process (1) for $p > 1$ have similar asymptotic behaviour as for the considered above case $p = 1$. Thus the proposed method may be applied for the estimation problem of the parameters of the SDDE's of the p -th order, i.e., of type (1).

Define the L_q -norm on the space of random vectors as $\|\cdot\|_q = (E_\vartheta \|\cdot\|^q)^{\frac{1}{q}}$, where $\|a\| = (\sum_{i=0}^p a_i^2)^{1/2}$. For any $\varepsilon > 0$ and arbitrary $q \geq 2$ we shall construct a sequential procedure ϑ_ε^* to estimate ϑ with ε -accuracy in the sense

$$\|\vartheta_\varepsilon^* - \vartheta\|_q^2 \leq \varepsilon.$$

Estimators with such a property may be used in various adaptive procedures occurring in control, prediction or filtration of stochastic processes.

2.1 Estimation Procedure for the Cases $\Theta_1, \dots, \Theta_4$

In this subsection we shall construct the sequential estimation plans for each of the regions $\Theta_1, \dots, \Theta_4$ separately. Afterwards in the following subsection we shall define our estimators for $\vartheta \in \Theta$ as a combination of these sequential estimators. Let us fix up a real number $q \geq 2$.

Estimation Procedure for the Cases Θ_1 and Θ'_3

The common property of Θ_1 and Θ'_3 consists in the equal asymptotic behaviour of both $\varphi_0(T)$ and $\varphi_1(T)$ as $T \rightarrow \infty$.

Denote by $SEP1(\varepsilon) = (T_1(\varepsilon), \vartheta_1^*(\varepsilon))$ the sequential estimation plan for $\vartheta \in \Theta_1 \cup \Theta'_3$ with prescribed accuracy $\varepsilon > 0$, where the duration of observations $T_1(\varepsilon)$ and the estimator $\vartheta_1^*(\varepsilon)$ of ϑ are defined as follows:

$$T_1(\varepsilon) = \tau_1(\sigma_1(\varepsilon), \varepsilon), \quad \vartheta_1^*(\varepsilon) = S_1^{-1}(\sigma_1(\varepsilon), \varepsilon) \sum_{n=1}^{\sigma_1(\varepsilon)} \beta_1^q(n, \varepsilon) \vartheta_1(n, \varepsilon). \quad (8)$$

To explain the quantities in (8) firstly we choose an unboundedly increasing sequence of positive numbers $(c_n)_{n \geq 1}$, such that $\sum_{n \geq 1} c_n^{-q/2} < \infty$. Now define

$$\tau_1(n, \varepsilon) = \inf \left\{ T > 0 : \left(\int_0^T X^2(t) dt \right)^{q/2} + \left(\int_0^T X^2(t-1) dt \right)^{q/2} = (\varepsilon^{-1} c_n)^{q/2} \right\},$$

and

$$\vartheta_1(n, \varepsilon) = G_{XX}^{-1}(\tau_1(n, \varepsilon)) \cdot \Phi_{XX}(\tau_1(n, \varepsilon)),$$

$$G_1(n, \varepsilon) = (\varepsilon^{-1}c_n)^{-1}G_{XX}(\tau_1(n, \varepsilon)),$$

$$\beta_1(n, \varepsilon) = \|G_1^{-1}(n, \varepsilon)\|^{-1},$$

$$S_1(N, \varepsilon) = \sum_{n=1}^N \beta_1^q(n, \varepsilon),$$

$$\sigma_1(\varepsilon) = \inf\{N \geq 1 : S_1(N, \varepsilon) \geq \delta_1^{-1}\varrho\},$$

where $\delta_1 \in (0, 1)$ is arbitrary but fixed and $\varrho = b_q 2^{\frac{q-2}{q}} \sum_{n \geq 1} c_n^{-q/2}$,

$$b_q = 2^{q-1} [3^{q-1} + 2^{q/2} (1 + q^q)] \left[\frac{q+1}{(q-1)^{q-1}} \right]^{q/2},$$

for $q > 2$ and $b_2 = 1$.

It should be pointed out, that for $q = 2$ the sequential plan SEP1(ε) completely coincides with the sequential plan presented in [5].

Estimation Procedure for the Case Θ_2

Define by SEP2 (ε) = ($T_2(\varepsilon), \vartheta_2^*(\varepsilon)$) the sequential estimation plan for $\vartheta \in \Theta_2$ as follows:

$$T_2(\varepsilon) = \tau_2(\sigma_2(\varepsilon), \varepsilon), \quad \vartheta_2^*(\varepsilon) = S_2^{-1}(\sigma_2(\varepsilon), \varepsilon) \sum_{n=1}^{\sigma_2(\varepsilon)} \beta_2^q(n, \varepsilon) \vartheta_2(n, \varepsilon).$$

In addition to the definitions introduced in the previous subsection, we use the notations

$$\begin{aligned} \lambda_t &= \int_0^t X(s)X(s-1) ds / \int_0^t X^2(s-1) dt, \\ Y_t &= X(t) - \lambda_t X(t-1), \quad \hat{\phi}_{YX}(t) = (Y_t, X(t))^T, \\ \hat{G}_{YX}(S, T) &= \int_S^T \hat{\phi}_{YX}(t) \hat{\phi}'_{YX}(t) dt, \quad \hat{\Phi}_{YX}(S, T) = \int_S^T \hat{\phi}_{YX}(t) dX(t), \\ \nu_2(n, \varepsilon) &= \inf \left\{ T > 0 : \int_0^T Y_t^2 dt = (\varepsilon^{-1}c_n)^\delta \right\}, \\ \tau_2(n, \varepsilon) &= \inf \left\{ T > \nu_2(n, \varepsilon) : \left(\frac{1}{\varepsilon^{-1}c_n} \int_{\nu_2(n, \varepsilon)}^T Y_t^2 dt \right)^{q/2} \right. \\ &\quad \left. + \left(\frac{1}{(\varepsilon^{-1}c_n)^{\alpha_2(n, \varepsilon)}} \int_{\nu_2(n, \varepsilon)}^T X^2(t) dt \right)^{q/2} = 1 \right\}, \end{aligned}$$

$$\begin{aligned} \alpha_2(n, \varepsilon) &= \ln \int_0^{\nu_2(n, \varepsilon)} X^2(t) dt / \delta \ln(\varepsilon^{-1} c_n), \\ \vartheta_2(n, \varepsilon) &= \hat{G}_{YX}^{-1}(\nu_2(n, \varepsilon), \tau_2(n, \varepsilon)) \cdot \hat{\Phi}_{YX}(\nu_2(n, \varepsilon), \tau_2(n, \varepsilon)), \\ \Psi_2(n, \varepsilon) &= \text{diag} \left\{ \varepsilon^{-1} c_n, (\varepsilon^{-1} c_n)^{\alpha_2(n, \varepsilon)} \right\}, \\ G_2(n, \varepsilon) &= (\varepsilon^{-1} c_n)^{-1/2} \Psi_2^{-1/2}(n, \varepsilon) \hat{G}_{YX}(\nu_2(n, \varepsilon), \tau_2(n, \varepsilon)), \\ \beta_2(n, \varepsilon) &= \|G_2^{-1}(n, \varepsilon)\|^{-1}, \\ S_2(N, \varepsilon) &= \sum_{n=1}^N \beta_2^q(n, \varepsilon), \\ \sigma_2(\varepsilon) &= \inf \{ N \geq 1 : S_2(N, \varepsilon) \geq \delta_2^{-1} \varrho \}, \end{aligned}$$

where δ and δ_2 denote some fixed constants from the interval $(0, 1)$.

Estimation Procedure for the Case Θ_3''

Define by $\text{SEP3}(\varepsilon) = (T_3(\varepsilon), \vartheta_3^*(\varepsilon))$ the sequential estimation plan for $\vartheta \in \Theta_3''$ as

$$T_3(\varepsilon) = \tau_3(\sigma_3(\varepsilon), \varepsilon), \quad \vartheta_3^*(\varepsilon) = S_3^{-1}(\sigma_3(\varepsilon), \varepsilon) \sum_{n=1}^{\sigma_3(\varepsilon)} \beta_3^q(n, \varepsilon) \vartheta_3(n, \varepsilon).$$

Here we firstly choose an unboundedly increasing sequence of positive numbers $\nu_3(n, \varepsilon)$, satisfying the following conditions:

$$\nu_3(n, \varepsilon) = o(\varepsilon^{-1} c_n) \text{ as } n \rightarrow \infty \text{ or } \varepsilon \rightarrow 0$$

and define

$$\begin{aligned} \tau_3(n, \varepsilon) &= \inf \left\{ T > \nu_3(n, \varepsilon) : \left(\frac{1}{\varepsilon^{-1} c_n} \int_{\nu_3(n, \varepsilon)}^T Y_t^2 dt \right)^{q/2} \right. \\ &\quad \left. + \left(\left(\frac{2\alpha_3(n, \varepsilon)}{\ln \varepsilon^{-1} c_n} \right)^4 \frac{1}{\varepsilon^{-1} c_n} \int_{\nu_3(n, \varepsilon)}^T X^2(t) dt \right)^{q/2} = 1 \right\}, \end{aligned}$$

$$\begin{aligned} \alpha_3(n, \varepsilon) &= \ln \lambda_{\nu_3(n, \varepsilon)}, \\ \vartheta_3(n, \varepsilon) &= \hat{G}_{YX}^{-1}(\nu_3(n, \varepsilon), \tau_3(n, \varepsilon)) \hat{\Phi}_{YX}(\nu_3(n, \varepsilon), \tau_3(n, \varepsilon)), \\ \Psi_3(n, \varepsilon) &= \text{diag} \left\{ \varepsilon^{-1} c_n, [(2\alpha_3(n, \varepsilon))^{-1} \ln \varepsilon^{-1} c_n]^4 \varepsilon^{-1} c_n \right\}, \\ G_3(n, \varepsilon) &= (\varepsilon^{-1} c_n)^{-1/2} \Psi_3^{-1/2}(n, \varepsilon) \hat{G}_{YX}(\nu_3(n, \varepsilon), \tau_3(n, \varepsilon)), \\ \beta_3(n, \varepsilon) &= \|G_3^{-1}(n, \varepsilon)\|^{-1}, \\ S_3(N, \varepsilon) &= \sum_{n=1}^N \beta_3^q(n, \varepsilon), \\ \sigma_3(\varepsilon) &= \inf \{ N \geq 1 : S_3(N, \varepsilon) \geq \delta_3^{-1} \varrho \}, \end{aligned}$$

where $\delta_3 \in (0, 1)$ is some fixed constant.

Estimation Procedure for the Case Θ_4

Define by $\text{SEP}_4(\varepsilon) = (T_4(\varepsilon), \vartheta_4^*(\varepsilon))$ the sequential estimation plan of $\vartheta \in \Theta_4$ as

$$T_4(\varepsilon) = \bar{\tau}(\sigma_4(\varepsilon), \varepsilon), \quad \vartheta_4^*(\varepsilon) = S_4^{-1}(\sigma_4(\varepsilon), \varepsilon) \sum_{n=1}^{\sigma_4(\varepsilon)} \beta_4^q(n, \varepsilon) \vartheta_4(n, \varepsilon). \quad (9)$$

Here we firstly choose an unboundedly increasing sequence of positive numbers $\nu_4(n, \varepsilon)$, satisfying the conditions

$$\nu_4(n, \varepsilon) = o(\varepsilon^{-1} c_n), \quad \nu_4(n, \varepsilon) / \ln \varepsilon^{-1} c_n \rightarrow \infty \quad \text{as } n \rightarrow \infty \text{ or } \varepsilon \rightarrow 0,$$

for some known number $\gamma \in (0, 1)$ denote

$$\delta(n, \varepsilon) = \gamma \inf_{\nu_4(n, \varepsilon)/2 \leq T \leq \nu_4(n, \varepsilon)} \frac{1}{T} \int_0^T Y_t^2 dt,$$

$$\alpha_4(n, \varepsilon) = \ln \lambda_{\nu_4(n, \varepsilon)},$$

$$\tau_4(n, \varepsilon) = \inf \left\{ T > \nu_4(n, \varepsilon) : \int_{\nu_4(n, \varepsilon)}^T Y_t^2 dt = \delta(n, \varepsilon) \varepsilon^{-1} c_n \right\},$$

$$\tau_5(n, \varepsilon) = \inf \left\{ T > \nu_4(n, \varepsilon) : \int_{\nu_4(n, \varepsilon)}^T X^2(t) dt = e^{2\alpha_4(n, \varepsilon) \varepsilon^{-1} c_n} \right\},$$

$$\Phi_{YX}^*(n, \varepsilon) = \left(\int_{\nu_4(n, \varepsilon)}^{\tau_4(n, \varepsilon)} Y_t dX(t), \int_{\nu_4(n, \varepsilon)}^{\tau_5(n, \varepsilon)} X(t) dX(t) \right)^T,$$

$$G_{YX}^*(n, \varepsilon) = \begin{bmatrix} \int_{\nu_4(n, \varepsilon)}^{\tau_4(n, \varepsilon)} Y_t X(t) dt & \int_{\nu_4(n, \varepsilon)}^{\tau_4(n, \varepsilon)} Y_t X(t-1) dt \\ \int_{\nu_4(n, \varepsilon)}^{\tau_5(n, \varepsilon)} X^2(t) dt & \int_{\nu_4(n, \varepsilon)}^{\tau_5(n, \varepsilon)} X(t) X(t-1) dt \end{bmatrix},$$

$$\bar{\tau}(n, \varepsilon) = \max(\tau_4(n, \varepsilon), \tau_5(n, \varepsilon)),$$

$$\vartheta_4(n, \varepsilon) = (G_{YX}^*(n, \varepsilon))^{-1} \cdot \Phi_{YX}^*(n, \varepsilon),$$

$$\Psi_4(n, \varepsilon) = \text{diag} \left\{ \varepsilon^{-1} c_n, e^{2\alpha_4(n, \varepsilon) \varepsilon^{-1} c_n} \right\},$$

$$G_4(n, \varepsilon) = (\varepsilon^{-1} c_n)^{-1/2} \Psi_4^{-1/2}(n, \varepsilon) G_{YX}^*(n, \varepsilon),$$

$$\beta_4(n, \varepsilon) = \|G_4^{-1}(n, \varepsilon)\|^{-1},$$

$$S_4(N, \varepsilon) = \sum_{n=1}^N \beta_4^q(n, \varepsilon),$$

$$\sigma_4(\varepsilon) = \inf \{ N \geq 1 : S_4(N, \varepsilon) \geq \delta_4^{-1} \varrho \},$$

where $\delta_4 \in (0, 1)$ is arbitrary but fixed.

2.2 General Sequential Estimation Procedure

Because in general it is unknown to which region ϑ belongs to, we define the sequential plan $(T(\varepsilon), \vartheta(\varepsilon))$ of estimation $\vartheta \in \Theta$ as a combination of all constructed above estimators by the formulae

$$T(\varepsilon) = \min(T_1(\varepsilon), \dots, T_4(\varepsilon)),$$

$$\vartheta(\varepsilon) = \chi_1(\varepsilon)\vartheta_1^*(\varepsilon) + \dots + \chi_4(\varepsilon)\vartheta_4^*(\varepsilon),$$

where $\chi_i(\varepsilon) = \chi(T(\varepsilon) = T_i(\varepsilon))$, $i = \overline{1,4}$, ($\chi(a = b) = 1$, $a = b$; 0 , $a \neq b$).

The proof of the following theorem will be included in a forthcoming paper.

Theorem 1. *Assume that the underlying process $(X(t))$ satisfies the equations (5),(6) and for the numbers $\delta_1, \dots, \delta_4$ in the definitions (8)-(9) of sequential plans the condition*

$$\sum_{k=1}^4 \delta_k^{2/q} = 1$$

is fulfilled. Then for any $\varepsilon > 0$ and every $\vartheta \in \Theta$ the sequential estimation plans $(T(\varepsilon), \vartheta(\varepsilon))$ of ϑ are closed ($T(\varepsilon) < \infty$ $P_\vartheta - a.s.$). They possess the following properties:

1° For any $\varepsilon > 0$

$$\sup_{\Theta} E_\vartheta \|\vartheta(\varepsilon) - \vartheta\|_q^2 \leq \varepsilon;$$

2° The following relations hold with $P_\vartheta - probability one:$

(i) for $\vartheta \in \Theta_1$ (stationary case):

$$\overline{\lim}_{\varepsilon \rightarrow 0} \varepsilon T(\varepsilon) < \infty,$$

(ii) for $\vartheta \in \Theta_2$:

$$\overline{\lim}_{\varepsilon \rightarrow 0} [T(\varepsilon) - \frac{1}{2v_1} \ln \varepsilon^{-1}] < \infty,$$

(iii) for $\vartheta \in \Theta'_3$:

$$\overline{\lim}_{\varepsilon \rightarrow 0} [T(\varepsilon) - \frac{1}{2v_0} \ln \varepsilon^{-1}] < \infty,$$

(iv) for $\vartheta \in \Theta''_3$:

$$\overline{\lim}_{\varepsilon \rightarrow 0} [T(\varepsilon) - \frac{1}{v_0} \ln T(\varepsilon) - \frac{1}{2v_0} \ln \varepsilon^{-1}] < \infty,$$

(v) for $\vartheta \in \Theta_4$: exists some positive constant C , such that

$$\overline{\lim}_{\varepsilon \rightarrow 0} [T(\varepsilon) - C\varepsilon^{-1}] < \infty;$$

3° for $\vartheta \in \Theta$ the estimator $\vartheta(\varepsilon)$ is strongly consistent:

$$\lim_{\varepsilon \rightarrow 0} \vartheta(\varepsilon) = \vartheta \quad P_\vartheta - a.s.$$

Acknowledgement

This paper was supported by grants RFFI-DFG 02-01-04001, 05-01-04004.

References

1. A.A. Gushchin and U. Küchler: Asymptotic inference for a linear stochastic differential equation with time delay. *Bernoulli* **5**(6), 1999, 1059–1098.
2. A.A. Gushchin and U. Küchler: On parametric statistical models for stationary solutions of affine stochastic delay differential equations. *Math. Meth. Stat.* **12**, 2003, 31–61.
3. A.Yu. Kolesov, and Yu.S. Kolesov: Relaxation oscillations in mathematical models of ecology. In: *Proceedings of the Steklov Institute of Mathematics*, vol. 199, no. 1, E.F. Mishchenko (ed.), 1995.
4. U. Küchler, and Y. Kutoyants: Delay estimation for some stationary diffusion-type processes. *Scand. J. Statistics* **27**(3), 2000, 405–414.
5. U. Küchler and V. Vasiliev: On sequential parameter estimation for some linear stochastic differential equations with time delay. *Sequential Analysis* **20**(3), 2001, 117–146.
6. U. Küchler and V. Vasiliev: Sequential identification of linear dynamic systems with memory. *Statist. Inference for Stochastic Processes* **8**, 2005, 1–24.
7. R.S. Liptser and A.N. Shiryaev: *Statistics of Random Processes*. Springer, New York, 1977.
8. M.C. Mackey: Commodity price fluctuations: price dependent delays and nonlinearities as explanatory factors. *J. Econ. Theory* **48**(2), 1989, 497–509.

Scalar Periodic Complex Delay Differential Equations: Small Solutions and their Detection

Neville J. Ford and Patricia M. Lumb

Department of Mathematics, University of Chester, Chester CH1 4BJ, UK,
{n.j.ford,p.lumb}@chester.ac.uk

Summary. We consider the detection of the existence of *small* (super-exponentially decaying) solutions for the equation

$$x'(t) = b(t)x(t-1), \text{ with } b(t+1) = b(t). \quad (1)$$

where the function b is complex-valued. We present a numerical method which extends our previous methods for real-valued b and we compare the effectiveness of an alternative numerical scheme.

1 Introduction and Background

Detecting non-trivial small solutions to delay differential equations (solutions that are not identically zero but which satisfy, for every real k , $e^{kt}x(t) \rightarrow 0$ as $t \rightarrow \infty$) is a key objective for the mathematical analyst (see [1, 3, 5, 7, 10, 11, 12, 13, 14, 15, 16, 17]). When an equation does not admit small solutions the eigenvectors and generalised eigenvectors span the solution space (see [4], Chapter V).

The analytical detection of small solutions for general DDEs is difficult. This prompts us to seek their detection by numerical methods and we have shown this to be viable for simple equations. In earlier work (see for example, [6, 7], for details of the methodology and for the underpinning theoretical results) for real-valued b , we concluded that small solutions could be detected through examining the eigenspectra of simple numerical approximations. Experimentally, using comparisons with other simple rules, we concluded that the trapezium rule provided an excellent choice.

The new feature of this paper is the focus on DDEs with complex-valued b because here the detection of small solutions presents new challenges. The fundamental analytical theory is under-developed and therefore the numerical insights break new ground. The work of Guglielmi (see [9]) has highlighted (through the concept of τ -stability) that for certain complex-valued delay equations the backward Euler rule has better stability properties than the trapezium rule and so we reconsider whether the use of the trapezium rule is still appropriate.

2 Known Analytical Results for the Complex Case

The starting point for the detection of small solutions needs to be the formulation of some equations with known behaviour which can form the basis for our testing. With this in mind, we present here two results that provide a basis for identifying suitable test problems:

The following Theorems give (respectively) sufficient conditions for the *absence* and *presence* of small solutions for (1) with b complex-valued:

Theorem 1. (Theorem 4.7 in [17]) *If b is such that the real and imaginary parts of b have constant sign, then (1) has no small solutions.*

Theorem 2. (see [17, p. 504])

A sufficient condition for the presence of small solutions to (1), is that there exist θ_1, θ_2 with $-1 \leq \theta_1 < \theta_2 \leq 0$ such that $\int_{\theta_1}^{\theta_2} b(s)ds = 0$. This is equivalent to requiring the curve $\zeta(t) = \int_{-1}^t b(s)ds$ to have a self intersection.

Recently, Verduyn Lunel has given *necessary and sufficient* conditions for the existence of small solutions to (1). The conditions are given (see [18, Theorem 4.5]) as

$$\left| \int_{\sigma_1}^{\sigma_2} b(\sigma)d\sigma \right| \leq \left| \int_{-1}^0 b(\sigma)d\sigma \right|, \quad \text{for every } \sigma_1, \sigma_2: -1 \leq \sigma_1 < \sigma_2 \leq 0.$$

This full characterisation clarifies certain cases that were not covered before.

3 A Summary of our Methodology

We compare the eigenspectrum, arising from discretisation of (1) with a constant step-size $h = \frac{1}{N}$, with that from the autonomous problem

$$x'(t) = \hat{b}x(t-1), \quad \text{where } \hat{b} = \int_0^1 b(t)dt, \quad (2)$$

noting that in the absence of small solutions the dynamics of the solution sets of the two problems are equivalent. Discretisation of (1) results in a difference equation of the form $y_{n+1} = A(n)y_n$. Here $A(n+N) = A(n)$ so $y_{n+N} = Cy_n$ where $C = \prod_{i=1}^N A(N-i)$. In (2), $A(n) = A$. In our figures we choose to represent the eigenvalues of C by '+' and those of A^N by '*'. The idea is that we compare the eigenspectrum derived by discretising (1) with that obtained by discretising (2) and use this as the basis for comparing the dynamics of the solution sets.

In [6] (with b real-valued and using the trapezium rule) we were able to identify recognisable characteristic shapes for the eigenspectra and these helped with the classification. In the cases where there are small solutions (the right hand cases in Figure 1), the eigenspectrum derived from (1) contains additional loops that are not present in the eigenspectrum derived from (2). In the left hand graph, the eigenspectra for the two equations almost coincide.

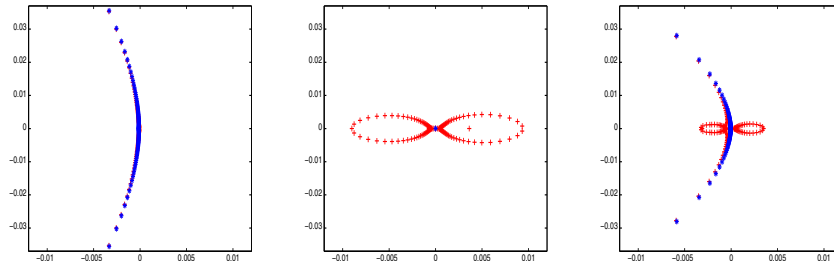


Fig. 1. Left: b does not change sign and there are no small solutions.
 Centre: b changes sign and $\int_0^1 b(t)dt = 0$ so almost all solutions are small.
 Right: b changes sign and $\int_0^1 b(t)dt \neq 0$ so the equation admits small solutions.

4 Numerical Results and their Interpretation

As we remarked in the introduction, it has been observed recently that the trapezium rule may become unstable for complex delay equations and this motivates us to present, as a double check, eigenspectra arising from the use of each of the two numerical methods, the trapezium rule (which is not τ -stable) and the backward Euler method (which is τ -stable, see [9]). In our illustrative examples we have chosen b to be a trigonometric function. However, our experiments have included a wide range of other function-types for b (see, for example, [12]). We define the solution map as in Section 3 and again compare the eigenspectra arising from the non-autonomous problem (1) and the autonomous problem (2).

In each of Figures 2 to 10 the left-hand diagram shows the eigenspectra arising from the trapezium rule and the right-hand diagram shows the eigenspectra arising from the backward Euler method.

In our examples we take $b(t) = \sin(2\pi t + d_1 + d_2 i) + c_1 + c_2 i$, where $c_1, c_2, d_1, d_2 \in \mathbb{R}$. We note that $\hat{b} = c_1 + c_2 i$.

We can rewrite $b(t)$ as

$$b(t) = \{\sin(2\pi t + d_1) \cosh(d_2) + c_1\} + i\{\cos(2\pi t + d_1) \sinh(d_2) + c_2\}.$$

First we present eigenspectra arising from problems that are known not to admit small solutions and begin our characterisation of the eigenspectra arising from (1) when b is complex-valued. If $|c_1| > \cosh(d_2)$ and $|c_2| > |\sinh(d_2)|$, then both the real and imaginary parts of b are of constant sign. Hence, by Theorem 1, we know that the equation does not admit small solutions and we expect the eigenspectra arising from the non-autonomous and autonomous problems to be very similar. Details of the examples included for this case are found in Table 1. The corresponding figures are indicated.

Example	c_1	c_2	d_1	d_2	$\cosh(d_2)$	$ \sinh(d_2) $	Figure	Small solutions
1	2	1	0.3	0.6	1.185	0.637	2	No
2	5	2.5	0.1	1.5	2.129	2.129	3	No
3	-1.5	0.2	1.6	-0.1	1.005	-0.100	4	No

Table 1. Details of examples where b does not change sign.

Of course, for complex-valued b , the trajectories are no longer symmetrical about the real axis. The figures we have here should be interpreted as indicating that no small solutions are present. The eigenvalues arising from use of the trapezium rule clearly lie on one asymptotic curve. However the backward Euler results are not so clear. In fact, this is consistent with the previous experiments we conducted in the real case (reported in [6]) where we saw similar deviation in the trajectories produced by the backward Euler scheme. This motivated us then to use the trapezium rule, and we have seen no evidence in our experiments here which would lead us to a different conclusion in the complex case.

The next examples satisfy the sufficient condition given in Theorem 2. Hence it is known that (1) will admit small solutions if we choose t_1, t_2 with $0 \leq t_1 < t_2 \leq 1$ such that

$$\int_{t_1}^{t_2} \{\sin(2\pi t + d_1 + d_2 i) + c_1 + c_2 i\} = 0.$$

We can show that this leads to

$$\frac{1}{\pi} \sin[\pi(t_1 + t_2) + d_1] \sin[\pi(t_2 - t_1)] \cosh(d_2) + c_1(t_2 - t_1) = 0 \tag{3}$$

and

$$\frac{1}{\pi} \cos[\pi(t_1 + t_2) + d_1] \sin[\pi(t_2 - t_1)] \sinh(d_2) + c_2(t_2 - t_1) = 0. \tag{4}$$

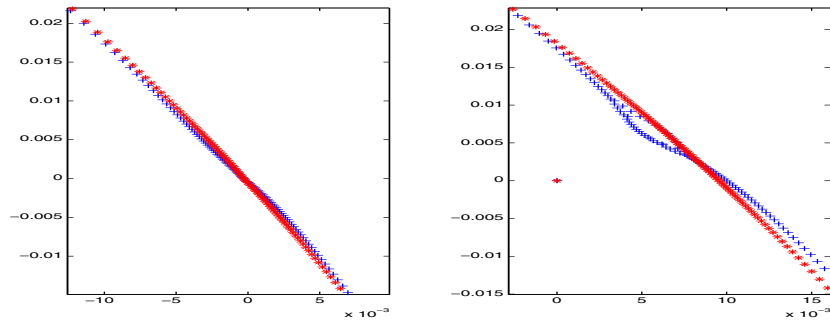


Fig. 2. Example 1 (Table 1). Left: Trapezium rule; right: Backward Euler.

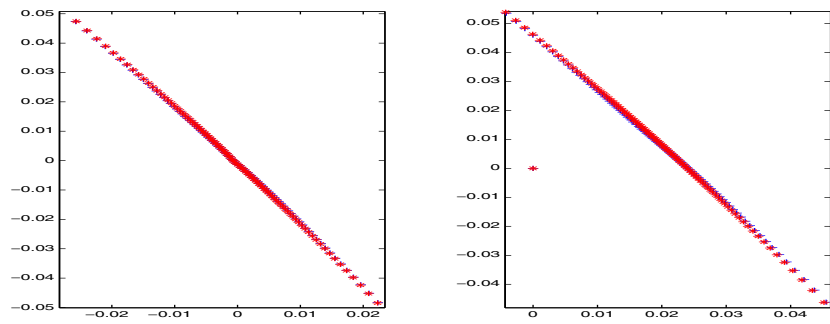


Fig. 3. Example 2 (Table 1) Left: Trapezium rule; right: Backward Euler.

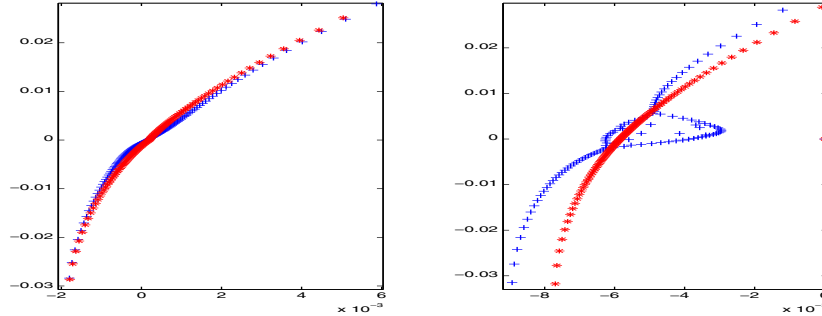


Fig. 4. Example 3 (Table 1) Left: Trapezium rule; right: Backward Euler.

We seek a solution in which $t_1 \neq t_2$. We use (3) and (4) to obtain

$$\frac{c_2}{c_1} = \frac{\tanh(d_2)}{\tan[\pi(t_1 + t_2) + d_1]}, \quad c_1 \neq 0, \quad \pi(t_1 + t_2) + d_1 \neq n\pi, n \in \mathbb{Z}, \quad (5)$$

and

$$\frac{\pi^2(t_2 - t_1)^2}{\sin^2[\pi(t_2 - t_1)]} \left\{ \frac{c_1^2}{\cosh^2(d_2)} + \frac{c_2^2}{\sinh^2(d_2)} \right\} = 1. \quad (6)$$

From equation (5) we see that

$$\pi(t_1 + t_2) + d_1 = n\pi + \tan^{-1} \left[\frac{c_1 \tanh(d_2)}{c_2} \right].$$

Equation (6) is of the form $\frac{\pi^2 x^2}{\sin^2(\pi x)} \{k\} = 1, x \neq 0$, where $x = t_2 - t_1$,

$$k = \frac{c_1^2}{\cosh^2(d_2)} + \frac{c_2^2}{\sinh^2(d_2)}.$$

Our analytical search for equations that admit small solutions reduces here to the following question. For a given problem can we find values of t_1 and t_2 such that both (5) and (6) are satisfied?

A visual inspection of the intersection of the curves $f_1(x) = k\pi^2 x^2$ and $f_2(x) = \sin^2(\pi x)$, combined with a search for the zeros of $f_1(x) = f_2(x)$ (using the Newton-Raphson method), enabled us to determine whether or not non-zero values of $(t_2 - t_1)$ satisfying (6) existed. Non-zero values of $(t_2 - t_1)$ exist if $0 < k < 1$. An infinite number of values of t_1 and t_2 are possible. We choose the value to give t_1 and t_2 in the required range.

Table 2 gives details of the equations being used for Figures 5 to 8. In Figure 5 an additional trajectory is observed for the non-autonomous problem. In Figure 6 the two trajectories are very different. The right-hand diagram of Figure 7 compares favourably with those produced using backward Euler when $b(t)$ is real and the equation admits small solutions (see [6]). The eigenspectra in Figure 8 resemble more closely those found in the real case (see Section 3).

Example	c_1	c_2	d_1	d_2	k	$(t_2 - t_1)$	$(t_1 + t_2)$	Figure	Small solutions
4	0.1	0.3	0.5	0.4	0.5420	0.4182	0.8809	5	Yes
5	0.3	0.4	0.1	2.5	0.0068	0.7062	0.8681	6	Yes
6	0.8	1.1	0.6	1.1	0.9082	0.1703	0.9678	7	Yes
7	0.6	0.01	0.2	0.1	0.3664	0.5243	1.3836	8	Yes

Table 2. Equations that satisfy the sufficient condition for small solutions to exist.

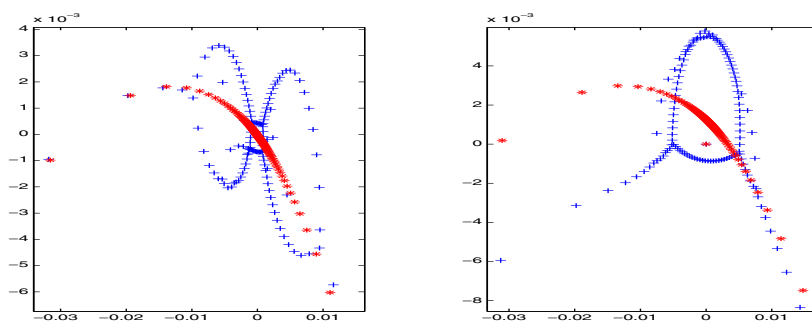


Fig. 5. Example 4 (Table 2).

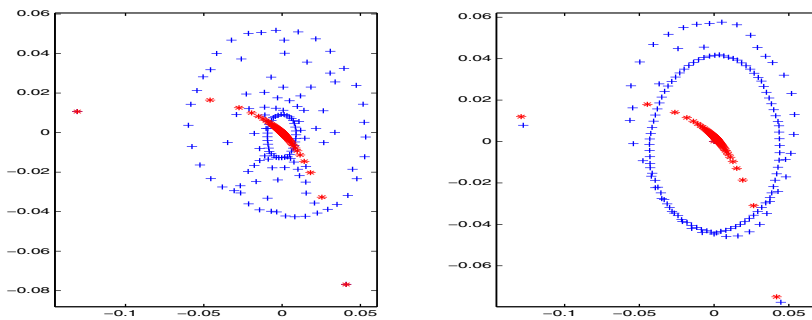


Fig. 6. Example 5 (Table 2).

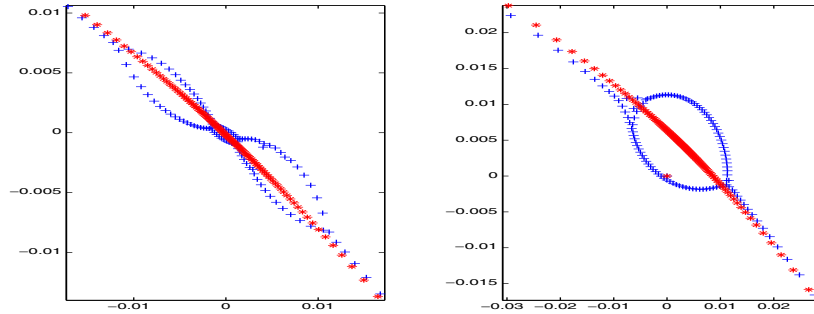


Fig. 7. Example 6 (Table 2).

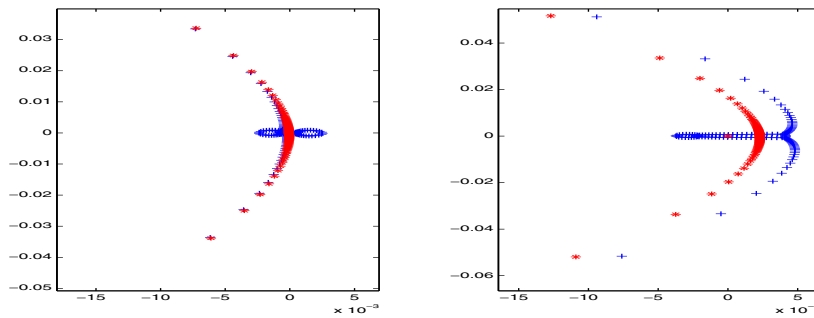


Fig. 8. Example 7 (Table 2).

Remark 1. If $d_2 = 0$ and $(t_2 - t_1) \neq 0$ then b is a real-valued function.

Remark 2. If $c_1 = 0$ and $t_1 \neq t_2$, then non-zero solutions to the equation $\pm \frac{\sinh(d_2)}{\pi} \sin(\pi x) + c_2 x = 0$, where $x = t_2 - t_1$ are needed to satisfy the sufficient condition for small solutions. A similar condition applies if $c_2 = 0$.

We can make the following observations:

1. The key theme in the detection of small solutions to complex delay equations remains the same as in the real case: in equations where small solutions occur, we detect an additional trajectory in the eigenspectrum compared to the autonomous case.
2. Figures 5 to 8 and our other experiments indicate that several characteristic shapes of eigenspectra now need to be interpreted as indicating the presence of small solutions to the equation.
3. We prefer the trapezium rule over the backward Euler rule because the eigenspectra for the two equations (1) and (2) obtained using the trapezium rule are more clearly different when the equation (1) does admit small solutions.

The examples we have considered so far are covered by the results of Theorems 1 and 2 but, to be really useful, the detection of small solutions needs to be possible even in the absence of theoretical analytical results. Therefore we undertook experimental work to determine whether or not certain equations that satisfied neither of the sufficient conditions given in Section 2 had small solutions. In other words, we used the numerical techniques we have developed to predict the presence or absence of small solutions.

We consider the examples in Table 3.

Example	c_1	c_2	d_1	d_2	$\cosh(d_2)$	$\sinh(d_2)$	Figure	Small solutions
8	0.3	0.4	0	0	1	0	9	Yes
9	0.4	value to give $k = 1$	1.3	0.1	1.0050	0.1002	10	Yes

Table 3. Equations were not covered by the previous theory but are now known to admit small solutions (according to the new theory).

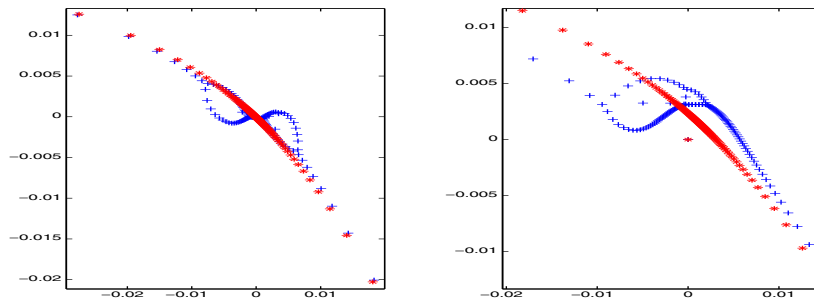


Fig. 9. Example 8 (Table 3).

Here we were able to predict the existence of small solutions using our numerical technique even though (at the time of the experiments) their presence could not be verified analytically. The very recent analytical work by Verduyn Lunel [18] enables the results of the numerical investigation to be verified retrospectively. In all cases we have investigated the numerical predictions have been confirmed by the new theory. We are able to conclude that

1. Our approach enables us to identify the presence or absence of small solutions even in cases that go beyond existing analytical results.
2. In the complex-valued case, there is more than one characteristic shape of eigen-spectrum that indicates the presence of small solutions so automation of the process will be difficult.

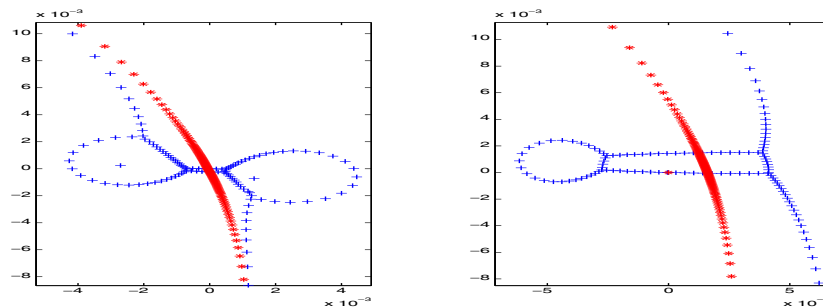


Fig. 10. Example 9 (Table 3).

3. We do not need to worry about instability of the trapezium rule for complex-valued equations. The trapezium rule appears to be at least as reliable as the backward Euler method.

References

1. D. Alboth: Individual asymptotics of C_0 -semigroups: lower bounds and small solutions. *J. Differential Equations* **143**, 1998, 221–242.
2. A. Bellen and M. Zennaro: *Numerical Methods for Delay Differential Equations*. Oxford University Press, 2003.
3. K.L. Cooke and S.M. Verduyn Lunel: Distributional and small solutions for linear time-dependent delay equations. *J. Differential and Integral Equations* **6**(5), 1993, 1101–1117.
4. O. Diekmann, S.A. van Gils, S.M. Verduyn Lunel, H.O. Walther: *Delay Equations: Functional, Complex and Nonlinear Analysis*. Springer, New York, 1995.
5. Y.A. Fiagbedzi: Characterization of small solutions in functional differential equations. *Appl. Math. Lett.* **10**, 1997, 97–102.
6. N.J. Ford and P.M. Lumb: Numerical approaches to delay equations with small solutions, In: *Proceedings of HERCMA*, E.A. Lipitakis (ed.), 2001, 101–108.
7. N.J. Ford and S.M. Verduyn Lunel: Characterising small solutions in delay differential equations through numerical approximations. *Applied Mathematics and Computation* **131**, 2002, 253–270.
8. N.J. Ford and S.M. Verduyn Lunel: Numerical approximation of delay differential equations with small solutions. *Proceedings of 16th IMACS World Congress on Scientific Computation, Applied Mathematics and Simulation, Lausanne 2000*, paper 173-3, New Brunswick, 2000.
9. N. Guglielmi: Delay dependent stability regions of θ -methods for delay differential equations. *IMA Journal of Numerical Analysis* **18**, 1998, 399–418.
10. J.K. Hale and S.M. Verduyn Lunel: *Introduction to Functional Differential Equations*. Springer, New York, 1993.
11. D. Henry: Small solutions of linear autonomous functional differential equations. *J. Differential Equations* **8**, 1970, 494–501.

12. P.M. Lumb: *Delay Differential Equations: Detection of Small Solutions*. Ph.D. thesis, Univ. Liverpool, UK, 2004. www.chester.ac.uk/math/pat.html.
13. S.M. Verduyn Lunel: Small solutions and completeness for linear functional and differential equations. In: *Oscillation and Dynamics in Delay Equations*, J.R. Graef and J.K. Hale (eds.), American Mathematical Society, 1992.
14. S.M. Verduyn Lunel: A sharp version of Henry's theorem on small solutions. *J. Differential Equations* **62**, 1986, 266–274.
15. S.M. Verduyn Lunel: Series expansions and small solutions for Volterra equations of convolution type. *J. Differential Equations* **85**, 1990, 17–53.
16. S.M. Verduyn Lunel: About completeness for a class of unbounded operators. *J. Differential Equations* **120**, 1995, 108–132.
17. S.M. Verduyn Lunel: Spectral theory for delay equations, In: *Systems, Approximation, Singular Integral Operators, and Related Topics*, A.A. Borichev and N.K. Nikolski (eds.), International Workshop on Operator Theory and Applications. *Operator Theory: Advances and Applications* **129**, 2001, 465–508.
18. S.M. Verduyn Lunel: New completeness and noncompleteness theorems for compact operators with applications. preprint MI-2005-10.
19. S.M. Verduyn Lunel: private communication.

Using Approximations to Lyapunov Exponents to Predict Changes in Dynamical Behaviour in Numerical Solutions to Stochastic Delay Differential Equations

Neville J. Ford and Stewart J. Norton

Department of Mathematics, University of Chester, Chester CH1 4BJ, UK,
{njford,s.norton}@chester.ac.uk

Summary. In this paper we explore the parameter values at which there are changes in qualitative behaviour of the numerical solutions to parameter-dependent linear stochastic delay differential equations with multiplicative noise. A possible tool in this analysis is the calculation of the approximate local Lyapunov exponents. We show that estimates for the maximal local Lyapunov exponent have predictable distributions dependent upon the parameter values and the fixed step length of the numerical method, and that changes in the qualitative behaviour of the solutions occur at parameter values that depend on the step length.

1 Introduction

The general form of stochastic delay differential equation that we consider takes the form

$$Y(t) = Y(t_0) + \int_{t_0}^t F(s, Y(s), Y(s - \tau))ds + \int_{t_0}^t G(s, Y(s), Y(s - \tau))dW(s), \quad (1)$$

with $Y(t) = \Phi(t)$ for $t \in [t_0 - \tau, t_0]$.

This equation is often written, in the Itô sense, in the shorthand form

$$\begin{aligned} dY(t) &= F(t, Y(t), Y(t - \tau))dt + G(t, Y(t), Y(t - \tau))dW(t), & t \geq t_0 \\ Y(t) &= \Phi(t), t \in [t_0 - \tau, t_0], \end{aligned}$$

where τ is the constant *time-lag* and $W(t)$ is a standard *Wiener* process. Following the terminology used in [1], F is called the *drift* term and G is the *diffusion* term. The analysis of equations of the general form (1) is still under development and there is comparatively little known about the qualitative behaviour of solutions of such a general equation as $t \rightarrow \infty$. For this reason, it is necessary to restrict our attention in the present paper to a simple linear test equation (2) below. We are restricting the equation to have instantaneous noise only. Despite its simplicity, the test equation

continues to present challenges both to classical and numerical analysis. In our test equation we take the time-lag $\tau = 1$.

$$\begin{aligned} dY(t) &= \lambda Y(t-1)dt + \mu Y(t)dW(t), \quad t \geq 0 \\ Y(t) &= t + \frac{1}{2}, \quad t \in [-1, 0], \lambda \in \mathbb{R}. \end{aligned} \quad (2)$$

Of particular interest to us is the investigation of the analogous behaviour in the stochastic equation of the bifurcation in the deterministic equation (3), where $\mu = 0$ in equation (2).

$$\begin{aligned} dY(t) &= \lambda Y(t-1)dt, \quad t \geq 0 \\ Y(t) &= t + \frac{1}{2}, \quad t \in [-1, 0]. \end{aligned} \quad (3)$$

Equation (3) is known to have a bifurcation at the parameter value $\lambda = -\frac{\pi}{2}$, (for example, see [6] p.17-19, or [4]), and Figure 1 illustrates this change in behaviour. For $\lambda > -\frac{\pi}{2}$ all possible solutions y satisfy $y(t) \rightarrow 0$ as $t \rightarrow \infty$ whereas for $\lambda < -\frac{\pi}{2}$ there can be solutions that become unbounded. Of course the particular solution in any specific case depends also on the starting function so this property of growing solutions may not always be seen for a specific starting function.

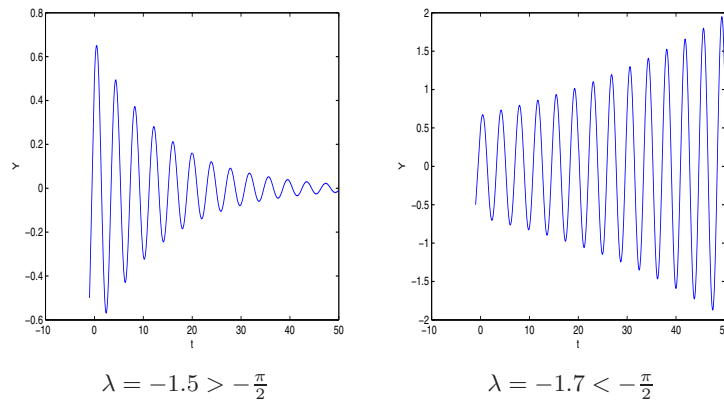


Fig. 1. Forward Euler, step size = 0.1 applied to (3).

A phenomenological approach was used in [7] to determine *by eye* the parameter values at which the behaviour of the linear deterministic equation (3) changes for the three most commonly used linear θ -methods, and the third order implicit Adams Moulton method. Figure 2 shows an intermediate state as we estimated the value of λ_{bif} to up to 6 decimal places. We also refer the reader to [7] for details of how the approach used here can be extended to other equations.

It was shown that the apparent bifurcation value of λ varied according to the numerical method and the step size h . The experiments showed that the errors in

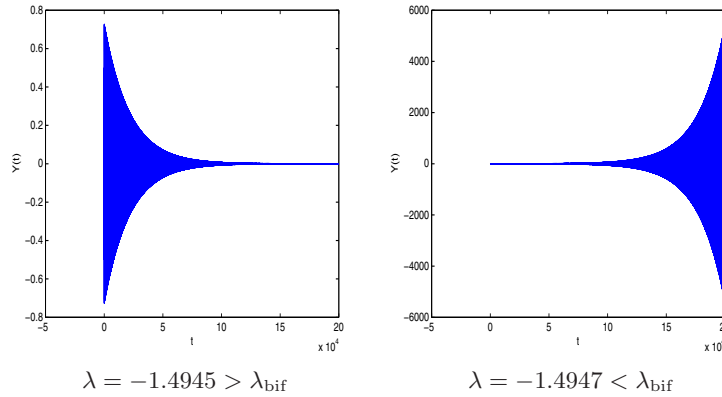


Fig. 2. Forward Euler, step size = 0.1 applied to (3).

the numerical values of λ at the apparent bifurcation from the theoretical value of $-\frac{\pi}{2}$ varies as h^n , where n is the order of the method. Also, for $\theta = 0$ and the implicit Adams Moulton method the apparent value of λ at which the change occurs was less than the theoretical value, whereas for $\theta = 0.5$ and $\theta = 1$ the value of λ was greater. For this deterministic equation, even more precise statements can be made about the bifurcation values of λ and we refer the reader, for example, to [2, 3].

A similar approach was taken with the stochastic equation (2), but it was evident that, for a given method and step size h , there was no single definite value for λ at which the behaviour changed. Using Matlab's random number generator it is possible to simulate the values of $dW(t)$ to simulate a single trajectory of the solution for a particular Brownian motion path. In fact, Matlab can repeat an identical Brownian motion path and this means that experiments are repeatable for different values of λ but with the same Brownian motion path. Figure 3 clearly shows that varying λ in this way produces different behaviour in a single trajectory. In addition, by repeating the experiment, we saw that the range of λ over which the change in behaviour occurred varied with different Brownian paths. As λ varies from -1.4925 to -1.4930 we can observe the solution becoming unbounded.

2 Dynamical Approach

The phenomenological approach used above has given us an insight into the changes in trajectories as the parameter λ changes and approaches $-\frac{\pi}{2}$. This approach has identified phenomenological or P-bifurcations.

The calculation of Lyapunov exponents and detecting the parameter values at which a Lyapunov exponent changes sign gives us a dynamical approach to seeking changes. This gives us the dynamical or D-bifurcations of the equations.

A linear stochastic delay equation has infinitely many Lyapunov exponents (see [1]) and for our approach we are interested in the principal (right most) Lyapunov exponent in the complex plane. We can define this value by

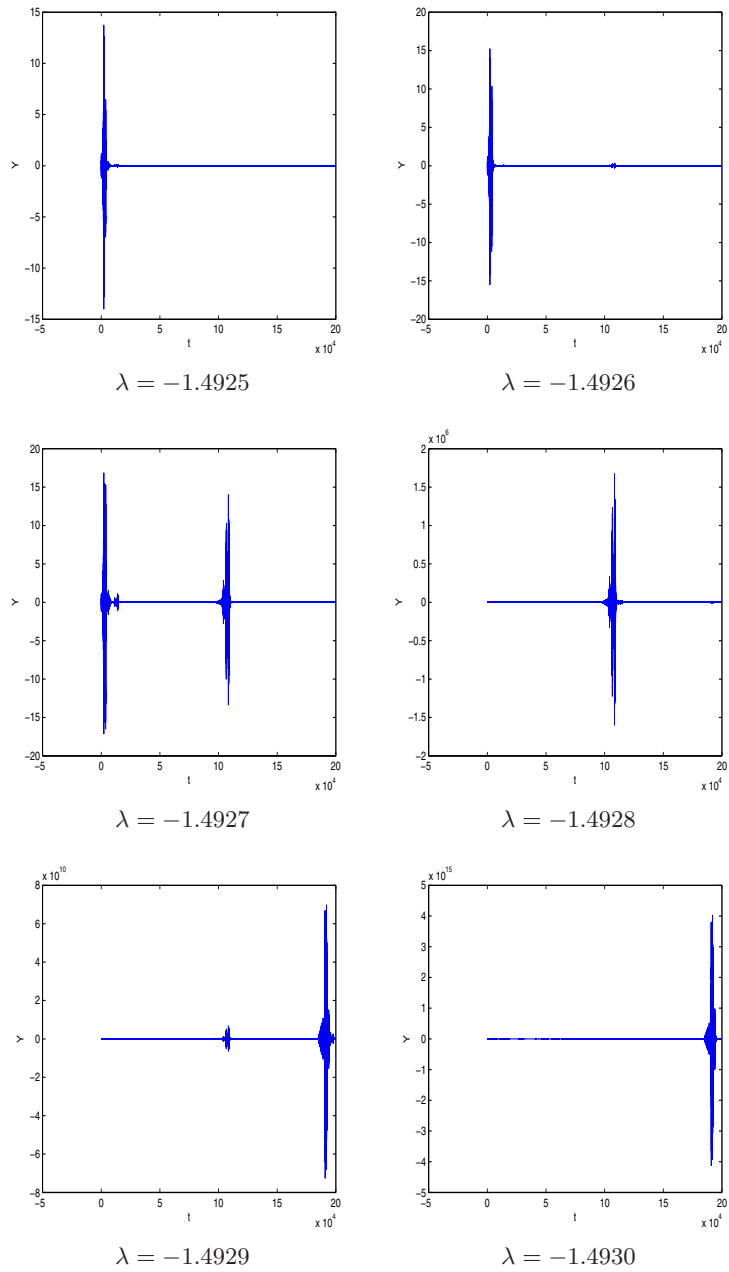


Fig. 3. Trajectories of equation (2) for $\mu = 0.1$ and stepsize $h = 0.1$, using the same Brownian path in each trajectory.

$$A = \lim_{t \rightarrow \infty} \sup E\left(\frac{1}{t} \log|Y(t)|\right).$$

We use the semi-implicit Euler method on equation (2), (see [5]), which is a stochastic version of the linear θ -method and leads to the numerical schemes

$$Y_{n+1} = Y_n + (1 - \theta)h\lambda Y_{n-N} + \theta h\lambda Y_{n+1-N} + \mu Y_n \Delta W_n,$$

where $Nh = 1$ and Y_{-N}, \dots, Y_0 are given by our initial function.

2.1 Methodology

For a range of values of λ close to $-\frac{\pi}{2}$ we used Matlab to simulate a large number of solution trajectories of our equation over the large interval $[0, T]$ for fixed values of μ, θ and step size h . We calculate $S = \sup_{[T-\epsilon, T]}(|Y(t)|)$ for each solution trajectory and calculate $L = \frac{\log(S)}{T}$ which might be taken as an estimate for the (local) Lyapunov exponent. We can now estimate the probability distribution of the values of L that we have found. It is important to note that in this paper we are not trying to find the best way to estimate a Lyapunov exponent but we are aiming to discover if L will give us information on the dynamical behaviour of each solution trajectory.

3 Experimental Results

For this paper we restrict the experiments to the method with $\theta = 0$, and $\mu = 0.1$. The results for other cases would be comparable. Preliminary experiments suggested that $T = 5000$ was sufficiently large to give consistent results without being so large that the experiments take excessive time. We set $\epsilon = 5$. For each λ , 500 trajectories were simulated and the 500 values of L were tabulated. We can construct histograms of the 500 values of L for $h = 0.1, \mu = 0.1$ and for λ close to the bifurcation value of -1.4927 suggested in Figure 3, and values either side of this.

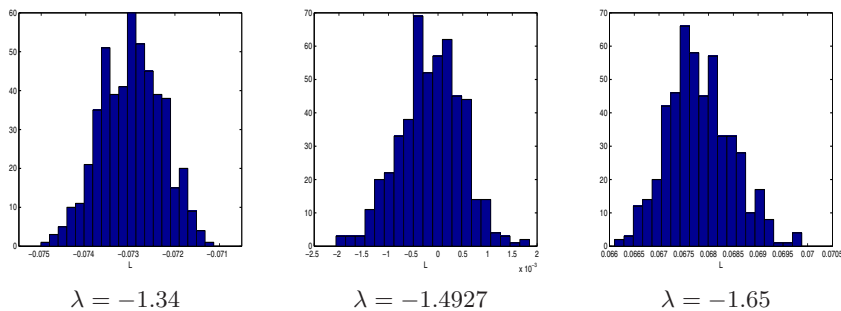


Fig. 4. Histogram of the 500 values of L for $\mu = 0.1$ and stepsize $h = 0.1$, using 500 fixed Brownian paths for direct comparisons.

Figure 4 shows that for $\lambda = -1.34$, for which every solution of equation (2) converges, all the values of L are negative. For $\lambda = -1.65$, for which every solution tends to infinity, all of the values of L are positive. At the parameter value $\lambda = -1.4927$ the phenomenological approach suggests that the behaviour appears to vary and for this value we can see from the figure that the range of values of L includes zero. In fact the figure indicates that the mean value of the set of 500 values is close to zero for this value of λ . The actual value of the mean of L is -0.000142 . Kolmogorov Smirnov tests on the histograms suggest that the distribution of the values of L are normal distributions in all three cases.

We can now consider the distribution of the mean value of L , L_{mean} , as we vary λ . Figure 5 shows that the graph of L_{mean} against λ produces what appears to be close to a straight line. However, a closer look shows that we have a slightly concave upwards curve. Regression analysis indicates that we have an excellent fit with a quadratic function and this curve has been added to the figure. The coefficient of determination, $R^2 = 1$, which confirms this excellent fit.

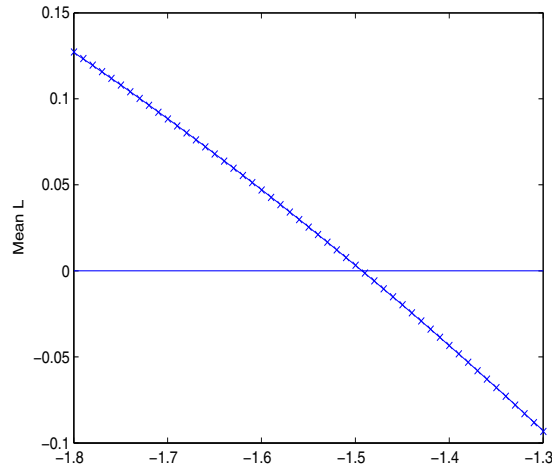


Fig. 5. Values of the L_{mean} as λ varies for $\mu = 0.1$, stepsize $h = 0.1$.

The equation of the curve shown in Figure 5 is

$$L_{\text{mean}} = -0.135577\lambda^2 - 0.860018\lambda - 0.981885.$$

Solving this for $L_{\text{mean}} = 0$ gives us a good estimate for the bifurcation value of λ , or certainly a good indication of the position of the interval over which this change occurs. This has been calculated as $\lambda = -1.4932$ which is consistent with the estimates possible using Figure 3.

We can also investigate how L_{mean} varies with h for a fixed λ . The value $\lambda = -1.49$ was chosen in the first instance. Seven values of h were used, 0.5, 0.25, 0.2,

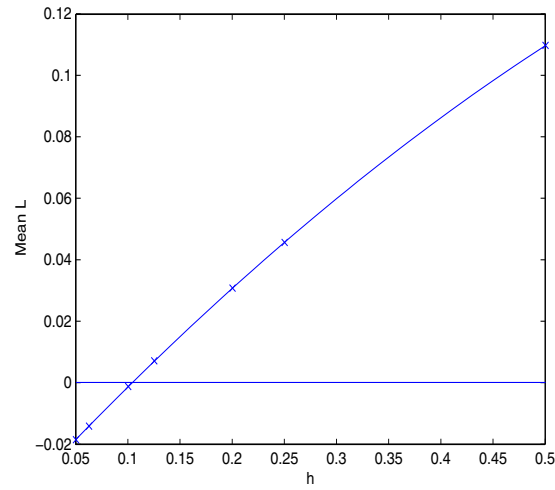


Fig. 6. Values of the mean of L as h varies for $\mu = 0.1$, $\lambda = -1.49$.

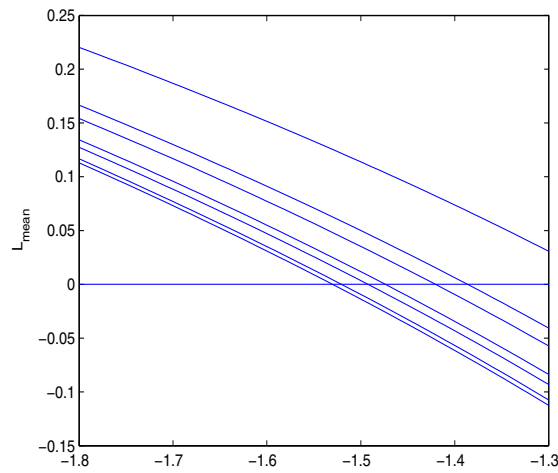


Fig. 7. Values of the mean of L as λ varies for $\mu = 0.1$
 From top to bottom, $h = 0.5, 0.25, 0.2, 0.125, 0.1, 0.0625, 0.05$.

0.125, 0.1, 0.0625 and 0.05. Figure 6 shows that the graph of L against h also produces what appears to be a slightly concave upwards curve. Regression analysis once again shows that a quadratic fit is excellent with a coefficient of determination of $R^2 = 1$, and the curve has been added to the figure.

The equation of the curve shown in figure 6 is

$$L_{\text{mean}} = -0.144113h^2 + 0.364118h - 0.036359.$$

Solving this for $L_{\text{mean}} = 0$ gives us a good estimate for the bifurcation value of the stepsize h for $\lambda = -1.49$. This has been calculated as $h = 0.175$. In other words, there is a critical step length at which the underlying dynamical behaviour of the equation will change.

We can plot the graphs of L_{mean} against λ with all seven of the chosen values of h . We can see from Figure 7 that we get seven almost parallel curves, and from the intersections with the line $L_{\text{mean}} = 0$, we can see how the stepsize moves the bifurcation value for λ .

We can finally see how L_{mean} varies with λ and h together. This can be seen in Figures 8,9.

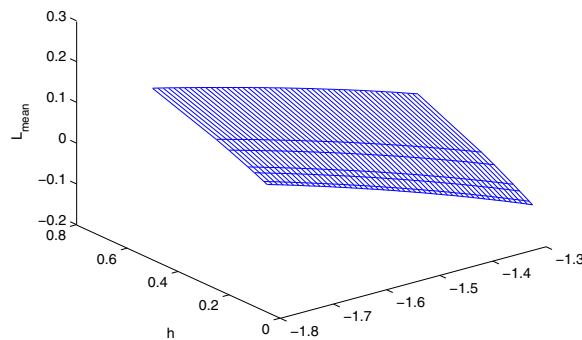


Fig. 8. L_{mean} against λ and h for $\mu = 0.1$.

Once again the regression equation has an excellent coefficient of determination, $R^2 = 0.998$. This equation is

$$L_{\text{mean}} = -0.131882\lambda^2 - 0.139499h^2 - 0.835240\lambda + 0.353116h - 0.986428.$$

We can use this equation to derive an expression for the bifurcation value λ in terms of h . This provides us with specific information about how the change in dynamical behaviour in the solution varies with the step size of the numerical scheme in use. In the present case it can be shown that

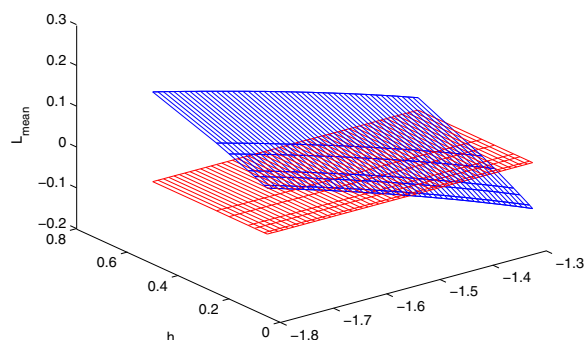


Fig. 9. L_{mean} against λ and h for $\mu = 0.1$, together with the plane $L_{\text{mean}} = 0$.

$$\lambda = -1.570421 - 0.838716h - 0.551686h^2 + \dots,$$

and that this expansion is valid for $-0.737 \leq h \leq 3.268$. It is clear that for small h , this expression represents a close $\mathcal{O}(h)$ approximation to the deterministic bifurcation value $-\frac{\pi}{2}$.

3.1 Conclusions

We have shown that the numerical solutions to equation (2) undergo changes of behaviour at particular values of the parameter λ . In addition, these values depend upon the stepsize h used (and also on the choice of numerical scheme but we have not had space to discuss this last point in detail in the current paper).

We have calculated L_{mean} , the mean of the Lyapunov exponent of 500 trajectories, and have shown that its distribution can be predicted as λ and h vary. The bifurcation values of the parameters estimated using the D-bifurcation method confirm (and make more precise) the predictions we were able to make using the P-bifurcation approach.

References

1. C.T.H. Baker, J.M. Ford, and N.J. Ford: Bifurcations in approximate solutions of stochastic delay differential equations. *International Journal of Bifurcation and Chaos* **14**(9), 2004, 2999–3021.
2. N.J. Ford and V. Wulf: Numerical Hopf bifurcation for a class of delay differential equations. *J. Comput. Appl. Math.* **115**, 2000, 601–616.
3. N.J. Ford and V. Wulf: How do numerical methods perform for delay differential equations undergoing a Hopf bifurcation? *J. Comput. Appl. Math.* **125**, 2000, 277–285.

4. K. Gopalsamy: *Stability and Oscillations in Delay Differential Equations of Population Dynamics*. Kluwer Academic, London, 1992.
5. D. Higham: Mean-square and asymptotic stability of the stochastic theta method. *SIAM J. Numer. Anal.* **38**, 2000 753–769.
6. J.D. Murray: *Mathematical Biology, 1: An Introduction*. 3rd edition, Springer, New York, 2002.
7. S.J. Norton and N.J. Ford: Predicting changes in dynamical behaviour in solutions to stochastic delay differential equations. *Communications on Pure and Applied Analysis*, to appear.

Superconvergence of Quadratic Spline Collocation for Volterra Integral Equations

Darja Saveljeva

Institute of Applied Mathematics, University of Tartu, Tartu 50409, Estonia,
darja.saveljeva@ut.ee

Summary. A collocation method with quadratic splines for Volterra integral equations is studied. Using special collocation points, error estimates at the collocation points are derived showing a more rapid convergence of order $\mathcal{O}(h^4)$ than the global uniform convergence of order $\mathcal{O}(h^3)$ in the interval of integration.

1 Introduction

One of the most practical methods for solving Volterra integral equations of the second kind is the polynomial spline collocation with step-by-step implementation. This method is known to be unstable for cubic and higher order smooth splines (see [5, 7, 11, 12]). In the case of quadratic splines of class C^1 the stability region consists only of one point [11]. In [13] one of the initial conditions, which are required by the standard quadratic spline collocation, is replaced by a not-a-knot boundary condition at the other end of the interval. These methods cannot now be implemented step-by-step and, in the case of linear integral equations, need the solution of a linear system which can be successfully done by Gaussian elimination. On the other hand, the nonlocal method with quadratic splines gives stability in the whole interval of collocation parameter.

The purpose of the present paper is to study the convergence rate of the nonlocal collocation method with quadratic splines at the collocation points for Volterra integral equations. The error analysis is based on a certain representation of quadratic splines and a general convergence theorem for operator equations. This research is closely related to the paper [13].

2 Description of the Method and Convergence Theorem

Consider the Volterra integral equation

$$y(t) = \int_0^t \mathcal{K}(t, s, y(s)) ds + f(t), \quad t \in [0, T], \quad (1)$$

where $f : [0, T] \rightarrow \mathbb{R}$ and $\mathcal{K} : R \times \mathbb{R} \rightarrow \mathbb{R}$ are given functions and the set R is defined by $R = \{(t, s) : 0 \leq s \leq t \leq T\}$.

A mesh $\Delta_N : 0 = t_0 < t_1 < \dots < t_N = T$ will be used representing the spline knots. As we consider the process $N \rightarrow \infty$, the knots t_i depend on N . Denote $h_i = t_i - t_{i-1}$. Then, for given collocation parameter $c \in (0, 1]$, define collocation points $\tau_i = t_{i-1} + ch_i, i = 1, \dots, N$. In order to determine the approximate solution u of the equation (1) as quadratic spline of class C^1 (denote this space by $S_2(\Delta_N)$), we impose the following collocation conditions

$$u(\tau_i) = \int_0^{\tau_i} \mathcal{K}(\tau_i, s, u(s))ds + f(\tau_i), \quad i = 1, \dots, N. \tag{2}$$

Since $\dim S_2(\Delta_N) = N + 2$ it is necessary to give two additional conditions which we choose

$$\begin{aligned} u(0) &= y(0), \\ u''(t_{N-1} - 0) &= u''(t_{N-1} + 0). \end{aligned} \tag{3}$$

We consider also the integral operator defined by

$$(Ku)(t) = \int_0^t \mathcal{K}(t, s, u(s))ds, \quad t \in [0, T].$$

Then the spline collocation problem (2), (3) is equivalent to the equation (see [13])

$$u = P_N Ku + P_N f, \quad u \in S_2(\Delta_N),$$

where the projection $P_N : C[0, T] \rightarrow C[0, T]$ is such that for any $v \in C[0, T]$ we have $P_N v \in S_2(\Delta_N)$ and

$$\begin{aligned} (P_N v)(0) &= v(0), \\ (P_N v)(\tau_i) &= v(\tau_i), \quad i = 1, \dots, N, \\ (P_N v)''(t_{N-1} - 0) &= (P_N v)''(t_{N-1} + 0). \end{aligned} \tag{4}$$

It was proved in [13] that, for any fixed $c \in (0, 1)$, in the case of quasi-uniform meshes, the projections P_N are uniformly bounded in the space $C[0, T]$. This allowed us to apply the classical convergence theorem for operator equations, which we are going to present, to show the convergence of the method.

Let E and F be Banach spaces, $\mathcal{L}(E, F)$ and $\mathcal{K}(E, F)$ spaces of linear continuous and compact operators. Suppose we have an equation

$$u = Ku + f \tag{5}$$

where $K \in \mathcal{K}(E, E)$ and $f \in E$. Let there be given a sequence of approximating operators $P_N \in \mathcal{L}(E, E), N = 1, 2, \dots$. Consider also equations

$$u_N = P_N K u_N + P_N f. \tag{6}$$

The following theorem may be called classical because it is one of the most important tools in the theory of approximate methods for integral equations (see [1, 3, 6]).

Theorem 1. *Suppose $u = Ku$ only if $u = 0$ and $P_N u \rightarrow u$ for all $u \in E$ as $N \rightarrow \infty$. Then equation (5) has the unique solution u^* , there is N_0 such that, for $N \geq N_0$, the equation (6) has the unique solution u_N^* . There are constants $C_1, C_2, C_3 > 0$ such that*

$$C_1 \|P_N u^* - u^*\| \leq \|u_N^* - u^*\| \leq C_2 \|P_N u^* - u^*\| \tag{7}$$

and

$$\|u_N^* - P_N u^*\| \leq C_3 \|K(P_N u^* - u^*)\|. \tag{8}$$

Note that this theorem can be deduced from more general ones [9, 14]. The reader can find the following notions, for instance, in [14].

The sequence of operators $A_N \in \mathcal{L}(E, F)$ is said to be *stably convergent* to the operator $A \in \mathcal{L}(E, F)$ if A_N converges to A pointwise (i.e., $A_N x \rightarrow Ax$ for all $x \in E$) and there is N_0 such that, for $N \geq N_0$, $A_N^{-1} \in \mathcal{L}(F, E)$ and $\|A_N^{-1}\| \leq \text{const}$. The sequence A_N is said to be *regularly convergent* to A if A_N converges to A pointwise and if x_N is bounded and $A_N x_N$ compact, then x_N is compact itself.

In the case $c = 1$ the sequence of projection operators P_N is unbounded. Nevertheless, the regular convergence of $I - P_N K$ to $I - K$ was proved (see [13]). This implies the two-sided error estimate (7) which guarantee the convergence for smooth solutions.

The rate of convergence of the method (2), (3) for linear equations is determined by the two-sided estimate (7). It is well known that quadratic spline interpolation projections P_N have the property $\|P_N u - u\| = O(h^3)$ for smooth functions u (see [8, 10]).

3 Superconvergence in the Case $c = 1/2$

In this section we show the superconvergence of the spline collocation method in collocation points for $c = 1/2$ and uniform mesh Δ_N , i.e., $h_i = h = T/N$, $i = 1, \dots, N$. We suppose also that the equation (1) is linear, i.e., $\mathcal{K}(t, s, u) = \mathcal{K}(t, s)$. As we have already mentioned, in this case projections P_N are uniformly bounded. Thus, Theorem 1 is applicable and the estimates (7) and (8) hold.

Using (4) and (8), we have for $\tau_i = t_{i-1} + h/2$

$$|u_N(\tau_i) - y(\tau_i)| = |u_N(\tau_i) - P_N y(\tau_i)| \leq \|u_N - P_N y\| \leq \text{const} \|K(P_N y - y)\|.$$

Therefore the rate of $\|K(P_N y - y)\|$ is the key problem in our investigation.

First of all we find a suitable representation of quadratic splines. Given any function $y \in C[0, T]$, let us consider $S = P_N y \in S_2(\Delta_N)$ determined by the conditions

$$\begin{aligned} S(0) &= y(0), \\ S(t_{i-1} + h/2) &= y(t_{i-1} + h/2), \quad i = 1, \dots, N, \\ S''(t_{N-1} - 0) &= S''(t_{N-1} + 0). \end{aligned}$$

Denote $S_{i-1/2} = S(t_{i-1} + h/2)$ and $m_i = S'(t_i)$. Using $t = t_{i-1} + \tau h$, we have the representation of S for $t \in [t_{i-1}, t_i]$

$$S(t) = S_{i-1/2} + \frac{h}{8}(2\tau - 1)((3 - 2\tau)m_{i-1} + (2\tau + 1)m_i).$$

The continuity of S in the knots gives

$$m_{i-1} + 6m_i + m_{i+1} = \frac{8}{h}(S_{i+1/2} - S_{i-1/2}), \quad i = 1, \dots, N - 1. \tag{9}$$

The initial condition $S(0) = y(0)$ adds the equation

$$3m_0 + m_1 = \frac{8}{h}(S_{1/2} - S_0) \tag{10}$$

and the not-a-knot requirement at t_{N-1} could be written in the form

$$m_{N-2} - 2m_{N-1} + m_N = 0. \tag{11}$$

The system of equations (10), (9), (11) has a unique solution. It will be calculated as $m_i = y'_i + \alpha_i h^2 y''_i + \beta_i$, $i = 0, \dots, N$, where $y'_i = y'(t_i)$ and $y''_i = y''(t_i)$. Suppose now and in the sequel that $y''' \in \text{Lip } 1$. Using a Taylor expansion in t_i , $i = 0, \dots, N$, we get

$$\begin{aligned} & \left(3\alpha_0 + \alpha_1 + \frac{1}{3}\right) h^2 y''_0 + \alpha_1 \mathcal{O}(h^3) + 3\beta_0 + \beta_1 = \mathcal{O}(h^3), \\ & \left(\alpha_{i-1} + 6\alpha_i + \alpha_{i+1} + \frac{2}{3}\right) h^2 y''_i + (\alpha_{i-1} + \alpha_{i+1}) \mathcal{O}(h^3) + \\ & \quad + \beta_{i-1} + 6\beta_i + \beta_{i+1} = \mathcal{O}(h^3), \quad i = 1, \dots, N - 1, \\ & (\alpha_{N-2} - \alpha_{N-1} + \alpha_N + 1) h^2 y''_N + (\alpha_{N-2} - 2\alpha_{N-1}) \mathcal{O}(h^3) + \\ & \quad + \beta_{N-2} - 2\beta_{N-1} + \beta_N = \mathcal{O}(h^3). \end{aligned}$$

Take $\alpha_i = -1/12$, $i = 0, \dots, N - 4$, and $\alpha_{N-3} = -67/840$, $\alpha_{N-2} = -11/105$, $\alpha_{N-1} = 1/24$, $\alpha_n = -341/420$, then β_i are uniquely defined and $\beta_i = \mathcal{O}(h^3)$, $i = 0, \dots, N$. Thus, for $t \in [t_{i-1}, t_i]$, we obtain the following expansions of the spline S

$$\begin{aligned} S(t) &= y(t) + y'''(t) \frac{h^3}{24} (-4\tau^3 + 6\tau^2 - 1) + \mathcal{O}(h^4), \quad i = 1, \dots, N - 4, \\ S(t) &= y(t) + y'''(t) \frac{h^3}{48} (1 - 2\tau) ((1 - 2\tau)^2 - 6(3 - 2\tau)\alpha_{i-1} - 6(3\tau + 1)\alpha_i) + \\ & \quad + \mathcal{O}(h^4), \quad i = N - 3, \dots, N. \end{aligned}$$

Then, for $t \in [t_{i-1}, t_i]$, $i = 1, \dots, N - 4$, we get

$$\begin{aligned} K(P_N y - y)(t) &= \int_0^t \mathcal{K}(t, s)(P_N y - y)(s) ds = \\ &= \frac{h^3}{24} \left(\sum_{k=1}^{i-1} \int_{t_{k-1}}^{t_k} \mathcal{K}(t, s) y'''(s) \varphi(\sigma) ds + \int_{t_{i-1}}^t \mathcal{K}(t, s) y'''(s) \varphi(\sigma) ds \right) + \mathcal{O}(h^4), \tag{12} \end{aligned}$$

where $\varphi(\tau) = -4\tau^3 + 6\tau^2 - 1$. The sum of integrals is of order $\mathcal{O}(h)$. Indeed, we have

$$\begin{aligned} \mathcal{K}(t, s) y'''(s) &= \mathcal{K}(t, t_{k-1} + \sigma h) y'''(t_{k-1} + \sigma h) = \\ &= \mathcal{K}(t, t_{k-1}) y'''(t_{k-1}) + \sigma h \left(\frac{\partial}{\partial s} \mathcal{K}(t, s) y'''(s) \right) \Big|_{s=\xi_k}, \quad \xi_k \in [t_{k-1}, t_k]. \end{aligned}$$

Then,

$$\int_{t_{k-1}}^{t_k} \mathcal{K}(t, s) y'''(s) \varphi(\sigma) ds = h \mathcal{K}(t, t_{k-1}) y'''(t_{k-1}) \int_0^1 \varphi(\sigma) d\sigma + h^2 \int_0^1 \left(\frac{\partial}{\partial s} \mathcal{K}(t, s) y'''(s) \right) \Big|_{s=\xi_k} \sigma \varphi(\sigma) d\sigma = \mathcal{O}(h^2),$$

as $\int_0^1 \varphi(\sigma) d\sigma = 0$ and using the assumption that $\mathcal{K}(t, s)$ is continuously differentiable with respect to s . The last integral in (12) can be estimated by **const** h . In the case $t \in [t_{k-1}, t_k]$, $k = N - 3, \dots, N$, there are a bounded number of integrals, each of order $\mathcal{O}(h)$. Hence, we have proved

Theorem 2. *Suppose that \mathcal{K} and $\partial\mathcal{K}/\partial s$ are continuous in $\{(t, s) \mid 0 \leq s \leq t \leq T\}$ and $y''' \in \text{Lip } 1$. Then, for $c = 1/2$, it holds*

$$\max_{1 \leq i \leq N} |u_N(t_{i-1} + h/2) - y(t_{i-1} + h/2)| = \mathcal{O}(h^4).$$

4 Superconvergence in the Case $c = 1$

According to [13] the sequence of projections P_N is not bounded when $c = 1$. Nevertheless, operators $I - P_N K$ converge regularly to $I - K$. In our case the regular and stable convergence coincide, so to prove the superconvergence we can use the modified estimate (8).

By definition of stable convergence the sequence of operators $(I - P_N K)^{-1}$ is bounded. Then, using (4), we have

$$\begin{aligned} |u_N(t_i) - y(t_i)| &= \|u_N - P_N y\| \leq \|(I - P_N K)^{-1}\| \|P_N K(P_N y - y)\| \leq \\ &\leq \mathbf{const} \|P_N K(P_N y - y)\|. \end{aligned}$$

In this section we shall show that $\|P_N K(P_N y - y)\| = \mathcal{O}(h^4)$.

First, as above, we are going to find an appropriate representation of the spline. Suppose the mesh Δ is uniform and $c = 1$. Given any function $y \in C[0, T]$, let us consider $S = P_N y \in S_2(\Delta_N)$ determined by the conditions

$$\begin{aligned} S(t_i) &= y(t_i), \quad i = 0, \dots, N, \\ S''(t_{N-1} - 0) &= S''(t_{N-1} + 0). \end{aligned}$$

Denote $S_i = S(t_i)$ and $S_{i-1/2} = S(t_{i-1} + h/2)$. Using $t = t_{i-1} + \tau h$, we get the representation of S for $t \in [t_{i-1}, t_i]$

$$S(t) = (1 - \tau)(1 - 2\tau)S_{i-1} + 4\tau(1 - \tau)S_{i-1/2} + \tau(2\tau - 1)S_i. \tag{13}$$

The continuity of S' in the knots t_i leads to the equations

$$S_{i-1} + 6S_i + S_{i+1} = 4(S_{i-1/2} + S_{i+1/2}), \quad i = 1, \dots, N - 1.$$

The not-a-knot boundary condition gives

$$S_N - S_{N-2} = 2(S_{N-1/2} - S_{N-3/2}).$$

Considering the values $S_i = y_i = y(t_i)$, $i = 0, \dots, N$, as known data, we have the linear system

$$\begin{pmatrix} 1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 1 & 0 & \cdots & 0 \\ & & \ddots & \ddots & & \\ 0 & \cdots & 0 & 1 & 1 \\ 0 & \cdots & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} S_{1/2} \\ S_{3/2} \\ \vdots \\ S_{N-3/2} \\ S_{N-1/2} \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{N-1} \\ d_N \end{pmatrix}, \tag{14}$$

where $d_i = (y_{i-1} + 6y_i + y_{i+1})/4$, $i = 1, \dots, N - 1$, and $d_N = (y_N - y_{N-2})/2$. However, the matrix of (14) is regular because its determinant is equal to 2. By direct calculation we obtain

$$\begin{aligned} S_{N-1/2} &= \frac{1}{8}(-y_{N-2} + 6y_{N-1} + 3y_N), \\ S_{N-3/2} &= \frac{1}{8}(3y_{N-2} + 6y_{N-1} - y_N), \\ S_{N-5/2} &= \frac{1}{8}(2y_{N-3} + 9y_{N-2} - 4y_{N-1} + y_N), \\ S_{k-1/2} &= \frac{1}{4}(y_{k-1} + 5y_k) - y_{k+1} + y_{k+2} - \dots \\ &\quad + \frac{(-1)^{N-k}}{8}(7y_{N-2} - 4y_{N-1} + y_N), \quad k = N - 3, \dots, 1. \end{aligned}$$

Consider the case when $N - k$ is even. Suppose now that $y''' \in \text{Lip } 1$. Using Simpson's rule, i.e.,

$$\begin{aligned} \frac{h}{3}(y_{k-1} + 4y_k + 2y_{k+1} + 4y_{k+2} + \dots + 4y_{N-2} + y_{N-1}) &= \\ = \int_{t_{k-1}}^{t_{N-1}} y(t)dt + \frac{h^4}{180}(y'''_{N-1} - y'''_{k-1}) + \mathcal{O}(h^5), \\ \frac{h}{3}(y_k + 4y_{k+1} + 2y_{k+2} + 4y_{k+3} + \dots + 4y_{N-1} + y_N) &= \\ = \int_{t_k}^{t_N} y(t)dt + \frac{h^4}{180}(y'''_N - y'''_k) + \mathcal{O}(h^5), \end{aligned}$$

we get

$$\begin{aligned} S_{k-1/2} &= \frac{3}{2h} \left(\int_{t_{k-1}}^{t_k} y(t)dt + \frac{h^4}{180}(y'''_k - y'''_{k-1}) \right) - \frac{1}{4}(y_{k-1} + y_k) - \\ &\quad - \frac{3}{2h} \left(\int_{t_{N-1}}^{t_N} y(t)dt - \frac{h^4}{180}(y'''_{N-1} - y'''_N) \right) - \frac{1}{8}y_{N-2} + y_{N-1} + \frac{5}{8}y_N. \end{aligned}$$

Performing a Taylor expansion in $t_{k-1/2}$ and t_N for the first and second rows of the above formula, respectively, we obtain

$$S_{k-1/2} = y_{k-1/2} + \frac{h^3}{16}y'''_N + \mathcal{O}(h^4). \tag{15}$$

Likewise, for $N - k$ odd, we get

$$S_{k-1/2} = y_{k-1/2} - \frac{h^3}{16} y_N''' + \mathcal{O}(h^4). \tag{16}$$

Now, substitute (15) or (16) for $S_{i-1/2}$ in (13) and use a Taylor expansion, which gives for $t = t_{i-1} + \tau h$

$$S(t) = y(t) - \frac{h^3}{12} y'''(t) \varphi(\tau) + (-1)^{N-i} \frac{h^3}{4} y'''(t_N) \phi(\tau) + \mathcal{O}(h^4), \tag{17}$$

where $\varphi(\tau) = \tau(1 - \tau)(1 - 2\tau)$ and $\phi(\tau) = \tau(1 - \tau)$.

Similarly to the case $c = 1/2$, we can show that $K(P_N y - y)(t) = \mathcal{O}(h^4)$. Namely, for $t \in [t_{i-1}, t_i]$, $i = 1, \dots, N$, we have

$$\begin{aligned} K(P_N y - y)(t) &= \int_0^t \mathcal{K}(t, s)(P_N y - y)(s) ds = \\ &= -\frac{h^3}{12} \int_0^t \mathcal{K}(t, s) y'''(s) \varphi(\sigma) ds + \frac{h^3}{4} y_N''' \int_0^t (-1)^{N-i} \mathcal{K}(t, s) \phi(\sigma) ds + \mathcal{O}(h^4). \end{aligned}$$

Using the same technique as in Section 3 and taking into account $\int_0^1 \varphi(\sigma) d\sigma = 0$, we get that the first integral is of order $\mathcal{O}(h)$. The second integral is also of order $\mathcal{O}(h)$, as $(-1)^{N-i} \int_{t_{i-1}}^{t_i} \phi(\sigma) ds + (-1)^{N-i-1} \int_{t_i}^{t_{i+1}} \phi(\sigma) ds = 0$.

Finally, apply the operator P_N to $K(P_N y - y)$. Assume that \mathcal{K} is continuous and three times continuously differentiable with respect to the first variable on $\{(t, s) : 0 \leq s \leq t \leq T\}$ and the function $t \mapsto \mathcal{K}(t, t)$ is two times continuously differentiable on $[0, T]$. Then it can be easily checked that $(K(P_N y - y))'''(t)$ is of order $\mathcal{O}(h)$. Thus, using (17), we have proved $\|P_N K(P_N y - y)\| = \mathcal{O}(h^4)$ and

Theorem 3. *Suppose that \mathcal{K} , $\partial\mathcal{K}/\partial s$, $\partial\mathcal{K}/\partial t$, $\partial^2\mathcal{K}/\partial t^2$ and $\partial^3\mathcal{K}/\partial t^3$ are continuous on $\{(t, s) : 0 \leq s \leq t \leq T\}$. Suppose also the function $t \mapsto \mathcal{K}(t, t)$ is twice continuously differentiable on $[0, T]$ and $y''' \in \text{Lip } 1$. Then,*

$$\max_{0 \leq i \leq N} |u_N(t_i) - y(t_i)| = \mathcal{O}(h^4)$$

in the case of uniform mesh.

5 Numerical Tests

In numerical tests we chose the test equation

$$y(t) = \lambda \int_0^t y(s) ds + f(t), \quad t \in [0, 1],$$

which has the exact solution $y(t) = (\sin t + \cos t + e^t)/2$. We also implemented the method for the equation in the linear case with $\mathcal{K}(t, s) = t - s$ and $f(t) = \sin t$ whose exact solution is $y(t) = (2 \sin t + e^t - e^{-t})/4$ on the interval $[0, 1]$. This equation is used in [2, 4, 13]. We calculated the error at the collocation points, i.e., $\max_{1 \leq i \leq N} |u_N(t_{i-1} + ch) - y(t_{i-1} + ch)|$.

For $c = 1$ and $c = 1/2$, the numerical experiments confirm the convergence rate $\mathcal{O}(h^4)$ predicted by theory. The results also show the superconvergence for $c = \mathcal{O}(h^2)$ for the test equations. It leads us to assume that superconvergence holds for the more general case. We state as an open problem that the superconvergence of the spline collocation method (2), (3) holds for collocation points with $c = \mathcal{O}(h^2)$.

Numerical results for $y(t) = \lambda \int_0^t y(s)ds + f(t)$:

$$\lambda = -2, f(t) = (3 \sin t - \cos t + 3e^t)/2$$

N	4	16	64	256
$c = 1$	$2.11 \cdot 10^{-4}$	$9.75 \cdot 10^{-7}$	$3.93 \cdot 10^{-9}$	$1.55 \cdot 10^{-11}$
$c = 0.5$	$1.59 \cdot 10^{-4}$	$9.33 \cdot 10^{-7}$	$3.99 \cdot 10^{-9}$	$1.59 \cdot 10^{-11}$
$c = N^{-2}$	$5.72 \cdot 10^{-5}$	$1.71 \cdot 10^{-7}$	$5.32 \cdot 10^{-10}$	$1.95 \cdot 10^{-12}$

$$\lambda = -1, f(t) = \sin t + e^t$$

N	4	16	64	256
$c = 1$	$1.16 \cdot 10^{-4}$	$6.25 \cdot 10^{-7}$	$2.62 \cdot 10^{-9}$	$1.04 \cdot 10^{-11}$
$c = 0.5$	$8.04 \cdot 10^{-5}$	$4.53 \cdot 10^{-7}$	$1.92 \cdot 10^{-9}$	$7.63 \cdot 10^{-12}$
$c = N^{-2}$	$3.56 \cdot 10^{-5}$	$1.15 \cdot 10^{-7}$	$3.66 \cdot 10^{-10}$	$1.34 \cdot 10^{-12}$

$$\lambda = 1, f(t) = \cos t$$

N	4	16	64	256
$c = 1$	$2.78 \cdot 10^{-4}$	$1.79 \cdot 10^{-6}$	$7.87 \cdot 10^{-9}$	$3.16 \cdot 10^{-11}$
$c = 0.5$	$7.29 \cdot 10^{-5}$	$3.68 \cdot 10^{-7}$	$1.52 \cdot 10^{-9}$	$5.98 \cdot 10^{-12}$
$c = N^{-2}$	$6.09 \cdot 10^{-5}$	$2.47 \cdot 10^{-7}$	$8.57 \cdot 10^{-10}$	$3.19 \cdot 10^{-12}$

$$\lambda = 2, f(t) = (-\sin t + 3 \cos t - e^t)/2$$

N	4	16	64	256
$c = 1$	$1.76 \cdot 10^{-3}$	$1.09 \cdot 10^{-5}$	$5.16 \cdot 10^{-8}$	$2.11 \cdot 10^{-10}$
$c = 0.5$	$1.19 \cdot 10^{-4}$	$6.87 \cdot 10^{-7}$	$3.11 \cdot 10^{-9}$	$1.26 \cdot 10^{-11}$
$c = N^{-2}$	$1.69 \cdot 10^{-4}$	$8.03 \cdot 10^{-7}$	$2.95 \cdot 10^{-9}$	$1.12 \cdot 10^{-11}$

Numerical results for $y(t) = \int_0^t (t-s)y(s)ds + f(t)$

N	4	16	64	256
$c = 1$	$2.39 \cdot 10^{-5}$	$1.53 \cdot 10^{-7}$	$1.03 \cdot 10^{-10}$	$2.67 \cdot 10^{-12}$
$c = 0.5$	$8.39 \cdot 10^{-7}$	$7.67 \cdot 10^{-9}$	$3.59 \cdot 10^{-11}$	$1.47 \cdot 10^{-13}$
$c = N^{-2}$	$1.99 \cdot 10^{-6}$	$1.17 \cdot 10^{-8}$	$4.46 \cdot 10^{-11}$	$1.71 \cdot 10^{-13}$

Acknowledgement

This work was supported by the Estonian Science Foundation grant 6704.

References

1. K.E. Atkinson: *The Numerical Solution of Integral Equations of the Second Kind*. Cambridge University Press, 1997.
2. C.T.H. Baker: *The Numerical Treatment of Integral Equations*. Clarendon Press, Oxford, 1977.
3. H. Brunner: *Collocation Methods for Volterra Integral and Related Functional Differential Equations*. Cambridge University Press, Cambridge, 2004.
4. H. Brunner and P.J. van der Houwen: *The Numerical Solution of Volterra Equations*. North-Holland, Amsterdam, 1986.
5. M.A.E. El Tom: On the numerical stability of spline function approximations to solutions of Volterra integral equations of the second kind. *BIT* **14**, 1974, 136–143.
6. W. Hackbusch: *Integral Equations: Theory and Numerical Treatment*. Birkhäuser, Basel, 1995.
7. H.-S. Hung: *The Numerical Solution of Differential and Integral Equations by Spline Functions*. MRC Technical report no. 1053, University of Wisconsin, Madison, 1970.
8. W.J. Kammerer, G.W. Reddien, and R.S. Varga: Quadratic interpolatory splines. *Numer. Math.* **22**, 1974, 241–259.
9. M.A. Krasnoselskii, G.M. Vainikko, P.P. Zabreiko, Y.B. Rutitskii, and V.Ya. Stecenko: *Approximate Solution of Operator Equations*. Wolters-Noordhoff, Groningen, 1972.
10. H. Mettke, E. Pfeifer, and E. Neuman: Quadratic spline interpolation with coinciding interpolation and spline grids. *J. Comp. Appl. Math.* **8**, 1982, 57–62.
11. P. Oja: Stability of the spline collocation method for Volterra integral equations. *J. Integral Equations Appl.* **13**, 2001, 141–155.
12. P. Oja: Stability of collocation by smooth splines for Volterra integral equations. In: *Mathematical Methods for Curves and Surfaces*, T. Lyche and L.L. Schumaker (eds.), Vanderbilt University Press, Nashville, 2001, 405–412.
13. P. Oja and D. Saveljeva: Quadratic spline collocation for Volterra integral equations. *Z. Anal. Anwendungen* **23**, 2004, 833–854.
14. G. Vainikko: *Functionalanalysis der Diskretisierungsmethoden*, Teubner, Leipzig, 1976.

Part VI

**Special Functions and Approximation on
Manifolds**

Asymptotic Approximations to Truncation Errors of Series Representations for Special Functions

Ernst Joachim Weniger

Institute for Physical and Theoretical Chemistry, University of Regensburg,
D-93040 Regensburg, Germany, joachim.weniger@chemie.uni-regensburg.de

Summary. Asymptotic approximations ($n \rightarrow \infty$) to the truncation errors $r_n = -\sum_{\nu=n+1}^{\infty} a_{\nu}$ of infinite series $\sum_{\nu=0}^{\infty} a_{\nu}$ for special functions are constructed by solving a system of linear equations. The linear equations follow from an approximative solution of the inhomogeneous difference equation $\Delta r_n = a_{n+1}$. In the case of the remainder of the Dirichlet series for the Riemann zeta function, the linear equations can be solved in closed form, reproducing the corresponding Euler-Maclaurin formula. In the case of the other series considered – the Gaussian hypergeometric series ${}_2F_1(a, b; c; z)$ and the divergent asymptotic inverse power series for the exponential integral $E_1(z)$ – the corresponding linear equations are solved symbolically with the help of Maple. The practical usefulness of the new formalism is demonstrated by some numerical examples.

1 Introduction

A large part of special function theory had been developed already in the 19th century. Thus, it is tempting to believe that our knowledge about special functions is essentially complete and that no significant new developments are to be expected. However, up to the middle of the 20th century, research on special functions had emphasized analytical results, whereas the efficient and reliable evaluation of most special functions had been – and to some extent still is – a more or less unsolved problem.

Due to the impact of computers on mathematics, the situation has changed substantially. We witness a revival of interest in special functions. The general availability of electronic computers in combination with the development of powerful computer algebra systems like Maple or Mathematica opened up many new applications, and it also created a great demand for efficient and reliable computational schemes (see for example [10, 14, 23] or [21, Section 13] and references therein).

Most special functions are defined via infinite series. Examples are the Dirichlet series for the Riemann zeta function,

$$\zeta(s) = \sum_{\nu=0}^{\infty} (\nu+1)^{-s}. \quad (1)$$

which converges for $\Re(s) > 1$, or the Gaussian hypergeometric series

$${}_2F_1(a, b; c; z) = \sum_{\nu=0}^{\infty} \frac{(a)_{\nu}(b)_{\nu}}{(c)_{\nu}\nu!} z^{\nu}, \quad (2)$$

which converges for $|z| < 1$.

The definition of special functions via infinite series is to some extent highly advantageous since it greatly facilitates analytical manipulations. However, from a purely numerical point of view, infinite series representations are at best a mixed blessing. For example, the Dirichlet series (1) converges for $\Re(s) > 1$, but is notorious for extremely slow convergence if $\Re(s)$ is only slightly larger than one. Similarly, the Gaussian hypergeometric series (2) converges only for $|z| < 1$, but the corresponding Gaussian hypergeometric function is a multivalued function defined in the whole complex plane with branch points at $z = 1$ and ∞ . A different computational problem occurs in the case of the asymptotic series for the exponential integral:

$$z e^z E_1(z) \sim \sum_{m=0}^{\infty} (-1/z)^m m! = {}_2F_0(1, 1; -1/z),$$

$$z \rightarrow \infty, \quad |\arg(z)| < 3\pi/2. \quad (3)$$

This series is probably the most simple example of a large class of series that diverge for every finite argument z and that are only asymptotic in the sense of Poincaré as $z \rightarrow \infty$. In contrast, the exponential integral $E_1(z)$, which has a cut along the negative real axis, is defined in the whole complex plane.

Problems with slow convergence or divergence were encountered already in the early days of calculus. Thus, numerical techniques for the acceleration of convergence or the summation of divergent series are almost as old as calculus. According to Knopp [12, p. 249], the first systematic work in this direction can be found in Stirling's book [20], which was published already in 1730 (recently, Tweddle [22] published a new annotated translation), and in 1755 Euler [9] published the series transformation which now bears his name. For a survey of the historical development, I recommend one book and two articles by Brezinski [4, 5, 6].

The convergence and divergence problems mentioned above can be formalized as follows: Let us assume that the partial sums $s_n = \sum_{\nu=0}^n a_{\nu}$ of a convergent or divergent but summable series form a sequence $\{s_n\}_{n=0}^{\infty}$ whose elements can be partitioned into a (generalized) limit s and a remainder or truncation error r_n according to

$$s_n = s + r_n, \quad n \in \mathbb{N}_0.$$

This implies

$$r_n = - \sum_{\nu=n+1}^{\infty} a_{\nu}, \quad n \in \mathbb{N}_0.$$

At least in principle, a convergent infinite series can be evaluated by adding up the terms successively until the remainders become negligible. This approach has two obvious shortcomings. Firstly, convergence can be so slow that it is uneconomical or practically impossible to achieve sufficient accuracy. Secondly, this approach does not work in the case of a divergent but summable series because increasing the index n normally only aggravates divergence.

As a principal alternative, we can try to compute a sufficiently accurate approximation \bar{r}_n to the truncation error r_n . If this is possible, \bar{r}_n can be eliminated from s_n , yielding a (much) better approximation $s_n - \bar{r}_n$ to the (generalized) limit s than s_n itself.

This approach looks very appealing since it is in principle remarkably powerful. In addition, it can avoid the troublesome asymptotic regime of large indices n , and it also works in the case of divergent but summable sequences and series. Unfortunately, it is by no means easy to obtain sufficiently accurate approximations \bar{r}_n to truncation errors r_n . The straightforward computation of r_n by adding up the terms does not gain anything.

The Euler-Maclaurin formula, which is discussed in Section 2, is a principal analytical tool that produces asymptotic approximations to truncation errors of monotone series in terms of integrals plus correction terms. Unfortunately, it is not always possible to apply the Euler-Maclaurin formula. Given a reasonably well behaved integrand, it is straightforward to compute a sum of integrand values plus derivatives of the integrand. But for a given series term a_n , it may be prohibitively difficult to differentiate and integrate it with respect to the index n .

In Section 3, an alternative approach for the construction of asymptotic approximations ($n \rightarrow \infty$) to the truncation errors r_n of infinite series is proposed that is based on the solution of a system of linear equations. The linear equations exist under very mild conditions: It is only necessary that the ratio a_{n+2}/a_{n+1} or similar ratios of series terms possesses an asymptotic expansion in terms of inverse powers $1/(n + \alpha)$ with $\alpha > 0$. Moreover, it is also fairly easy to solve these linear equations since they have a triangular structure.

The asymptotic nature of the approximants makes it difficult to use them also for small indices n , although this would be highly desirable. In Section 4, it is mentioned briefly that factorial series and Padé approximants can be helpful in this respect since they can accomplish a numerical analytic continuation.

In Section 5, the formalism proposed in this article is applied to the truncation error of the Dirichlet series for the Riemann zeta function. It is shown that the linear equations can in this case be reduced to a well known recurrence formula of the Bernoulli numbers. Accordingly, the terms of the corresponding Euler-Maclaurin formula are exactly reproduced.

In Section 6, the Gaussian hypergeometric series ${}_2F_1(a, b; c; z)$ is treated. Since the terms of this series depend on three parameters and one argument, a closed form solution of the linear equations seems to be out of reach. Instead, approximations are computed symbolically with the help of the computer algebra system Maple. The practical usefulness of these approximations is demonstrated by some numerical examples.

In Section 7, the divergent asymptotic inverse power series for the exponential integral $E_1(z)$ is treated. Again, the linear equations are solved symbolically with the help of Maple, and the practical usefulness of these solutions is demonstrated by some numerical examples.

2 The Euler-Maclaurin Formula

The derivation of the Euler-Maclaurin formula is based on the assumption that $g(x)$ is a smooth and slowly varying function. Then, $\int_M^N g(x)dx$ with $M, N \in \mathbb{Z}$ can be

approximated by the finite sum $\frac{1}{2}g(M)+g(M+1)+\dots+g(N-1)+\frac{1}{2}g(N)$. This finite sum can also be interpreted as a trapezoidal quadrature rule. In the years between 1730 and 1740, Euler and Maclaurin derived independently correction terms to this quadrature rule, which ultimately yielded what we now call the Euler-Maclaurin formula (see for example [21, Eq. (1.20)]):

$$\sum_{\nu=M}^N g(\nu) = \int_M^N g(x) dx + \frac{1}{2} [g(M) + g(N)] + \sum_{j=1}^k \frac{B_{2j}}{(2j)!} [g^{(2j-1)}(N) - g^{(2j-1)}(M)] + R_k(g), \quad (4a)$$

$$R_k(g) = -\frac{1}{(2k)!} \int_M^N B_{2k}(x - [x]) g^{(2k)}(x) dx. \quad (4b)$$

Here, $g^{(m)}(x)$ is the m -th derivative, $[x]$ is the integral part of x , $B_m(x)$ is a Bernoulli polynomial defined by the generating function $te^{xt}/(e^t - 1) = \sum_{n=0}^{\infty} B_n(x)t^n/n!$, and $B_m = B_m(0)$ is a Bernoulli number.

It is not a priori clear whether the integral $R_k(g)$ in (4b) vanishes as $k \rightarrow \infty$ for a given function $g(x)$. Thus, the Euler-Maclaurin formula may lead to an asymptotic expansion that ultimately diverges. In this article, it is always assumed that the Euler-Maclaurin formula and related expansions are only asymptotic in the sense of Poincaré.

Although originally used to express the in the early 18th century still unfamiliar integral in terms more elementary quantities, the Euler-Maclaurin formula is now often used to approximate the truncation error $r_n = -\sum_{\nu=n+1}^{\infty} a_{\nu}$ of a slowly convergent monotone series by an integral plus correction terms. The power and the usefulness of this approach can be demonstrated convincingly via the Dirichlet series (1) for the Riemann zeta function.

The terms $(\nu + 1)^{-s}$ of the Dirichlet series (1) are obviously smooth and slowly varying functions of the index ν , and they can be differentiated and integrated easily. Thus, the application of the Euler-Maclaurin formula (4) with $M = n + 1$ and $N = \infty$ to the truncation error of the Dirichlet series yields:

$$-\sum_{\nu=n+1}^{\infty} (\nu + 1)^{-s} = -\frac{(n + 2)^{1-s}}{s - 1} - \frac{1}{2} (n + 2)^{-s} - \sum_{j=1}^k \frac{(s)_{2j-1} B_{2j}}{(2j)!} (n + 2)^{-s-2j+1} + R_k(n, s), \quad (5a)$$

$$R_k(n, s) = \frac{(s)_{2k}}{(2k)!} \int_{n+1}^{\infty} \frac{B_{2k}(x - [x])}{(x + 1)^{s+2k}} dx. \quad (5b)$$

Here, $(s)_m = s(s + 1) \cdots (s + m - 1) = \Gamma(s + m)/\Gamma(s)$ with $s \in \mathbb{C}$ and $m \in \mathbb{N}_0$ is a Pochhammer symbol.

In [3, Tables 8.7 and 8.8, p. 380] and in [27, Section 2] it was shown that a few terms of the sum in (5a) suffice for a convenient and reliable computation of $\zeta(s)$ with $s = 1.1$ and $s = 1.01$, respectively. For these arguments, the Dirichlet series for $\zeta(s)$ converges so slowly that it is practically impossible to evaluate it by adding up its terms.

In order to understand better its nature, the Euler-Maclaurin formula (4) is rewritten in a more suggestive form. Let us set $M = n + 1$ and $N = \infty$, and let us also assume $\lim_{N \rightarrow \infty} g(N) = \lim_{N \rightarrow \infty} g'(N) = \lim_{N \rightarrow \infty} g''(N) = \dots = 0$. With the help of $B_0 = 1$, $B_1 = -1/2$, and $B_{2n+1} = 0$ with $n \in \mathbb{N}$ (see for example [21, p. 3]), we obtain:

$$- \sum_{\nu=n+1}^{\infty} g(\nu) = -B_0 \int_{n+1}^{\infty} g(x) dx + \sum_{\mu=1}^m \frac{(-1)^{\mu-1} B_{\mu}}{\mu!} g^{(\mu-1)}(\nu) + R_m(g), \quad (6a)$$

$$R_m(g) = \frac{(-1)^m}{(m)!} \int_{n+1}^{\infty} B_m(x - [x]) g^{(m)}(x) dx, \quad m \in \mathbb{N}. \quad (6b)$$

In the same way, we obtain for the Euler-Maclaurin approximation (5) to the truncation error of the Dirichlet series:

$$- \sum_{\nu=n+1}^{\infty} (\nu + 1)^{-s} = \sum_{\mu=0}^m \frac{(-1)^{\mu-1} (s)_{\mu-1} B_{\mu}}{\mu!} (n + 2)^{1-s-\mu} + R_m(n, s), \quad (7a)$$

$$R_m(n, s) = \frac{(-1)^m (s)_m}{(m)!} \int_{n+1}^{\infty} \frac{B_m(x - [x])}{(1 + x)^{s+m}} dx, \quad m \in \mathbb{N}. \quad (7b)$$

The reformulated Euler-Maclaurin approximation (7) looks suspiciously like a truncated expansion of the truncation error in terms of the asymptotic sequence $\{(n + 2)^{-\mu}\}_{\mu=0}^{\infty}$ of inverse powers. An analogous interpretation of the reformulated Euler-Maclaurin formula (6) is possible if we assume that the quantities $\int_{n+1}^{\infty} g(x) dx, g(n), g'(n), g''(n), \dots$ form an asymptotic sequence $\{\mathcal{G}_{\mu}(n)\}_{\mu=0}^{\infty}$ as $n \rightarrow \infty$ according to

$$\mathcal{G}_0(n) = \int_{n+1}^{\infty} g(x) dx,$$

$$\mathcal{G}_{\mu}(n) = g^{(\mu-1)}(n), \quad \mu \in \mathbb{N}.$$

The expansion of the truncation error $-\sum_{\nu=n+1}^{\infty} g(\nu)$ in terms of the asymptotic sequence $\{\mathcal{G}_{\mu}(n)\}_{\mu=0}^{\infty}$ according to (6) has the undeniable advantage that the expansion coefficients do not depend on the terms $g(\nu)$ and are explicitly known. The only remaining computational problem is the determination of the leading elements of the asymptotic sequence $\{\mathcal{G}_{\mu}(n)\}_{\mu=0}^{\infty}$. In the case of the Dirichlet series (1), this is trivially simple. Unfortunately, the terms of most series expansions for special functions are (much) more complicated than the terms of the Dirichlet series (1). In those less fortunate cases, it can be extremely difficult to do the necessary differentiations and integrations. Thus, the construction of the asymptotic sequence $\{\mathcal{G}_{\mu}(n)\}_{\mu=0}^{\infty}$ may turn out to be an unsurmountable problem.

3 Asymptotic Approximations to Truncation Errors

Let us assume that we want to construct an asymptotic expansion of a special function $f(z)$ as $z \rightarrow \infty$. First, we have to find a suitable asymptotic sequence $\{\varphi_j(z)\}_{j=0}^{\infty}$. Obviously, $\{\varphi_j(z)\}_{j=0}^{\infty}$ must be able to model the essential features of

$f(z)$ as $z \rightarrow \infty$. On the other hand, $\{\varphi_j(z)\}_{j=0}^\infty$ should also be sufficiently simple in order to facilitate the necessary analytical manipulations. In that respect, the most convenient asymptotic sequence is the sequence $\{z^{-j}\}_{j=0}^\infty$ of inverse powers, and it is also the one which is used almost exclusively in special function theory. An obvious example is the asymptotic series (3).

The behavior of most special functions as $z \rightarrow \infty$ is incompatible with an expansion in terms of inverse powers. Therefore, an indirect approach has to be pursued: Let us assume that for a given $f(z)$ one can find some $g(z)$ such that $f(z)/g(z)$ admits an asymptotic expansion in terms of inverse powers:

$$f(z)/g(z) \sim \sum_{j=0}^{\infty} c_j/z^j, \quad z \rightarrow \infty. \quad (8)$$

Although $f(z)$ cannot be expanded in terms of inverse powers $\{z^{-j}\}_{j=0}^\infty$, it can be expanded in terms of the asymptotic sequence $\{g(z)/z^j\}_{j=0}^\infty$. The asymptotic series (3) is of the form of (8) with $f(z) = E_1(z)$ and $g(z) = \exp(-z)/z$.

It is the central hypothesis of this article that such an indirect approach is useful for the construction of asymptotic approximations to remainders $r_n = -\sum_{\nu=n+1}^\infty a_\nu$ of infinite series as $n \rightarrow \infty$. Thus, instead of trying to use the technically difficult Euler-Maclaurin formula (4), we should try to find some ρ_n such that the ratio r_n/ρ_n admits an asymptotic expansion as $n \rightarrow \infty$ in terms of inverse powers $\{(n+\alpha)^{-j}\}_{j=0}^\infty$ with $\alpha > 0$.

A natural candidate for ρ_n is the first term a_{n+1} neglected in the partial sum $s_n = \sum_{\nu=0}^n a_\nu$, but in some cases it is better to choose instead $\rho_n = a_n$ or $\rho_n = (n+\alpha)a_{n+1}$ with $\alpha > 0$. Moreover, the terms a_{n+1} and the remainders r_n of an infinite series are connected by the inhomogeneous difference equation

$$\Delta r_n = r_{n+1} - r_n = a_{n+1}, \quad n \in \mathbb{N}_0. \quad (9)$$

In Jagerman's book [11, Chapter 3 and 4], solutions to difference equations of that kind are called Nörlund sums.

If we knew how to solve (9) efficiently and reliably for essentially arbitrary inhomogeneities a_{n+1} , all problems related to the evaluation of infinite series would in principle be solved. Unfortunately, this is not the case. Nevertheless, we can use (9) to construct the leading terms of an asymptotic expansion of r_n/a_{n+1} or of related expressions in terms of inverse powers.

For that purpose, we make the following ansatz:

$$r_n^{(m)} = -a_{n+1} \sum_{\mu=0}^m \frac{\gamma_\mu^m}{(n+\alpha)^\mu}, \quad n \in \mathbb{N}_0, \quad m \in \mathbb{N}, \quad \alpha > 0. \quad (10)$$

This ansatz, which is inspired by the theory of converging factors [1, 16] and by a truncation error estimate for Levin's sequence transformation [13] proposed by Smith and Ford [19, Eq. (2.5)] (see also [24, Section 7.3] or [26, Section IV]), is not completely general and has to be modified slightly both in the case of the Dirichlet series (1) for the Riemann zeta function, which is discussed in Section 5, and in the case of the divergent asymptotic series (3) for the exponential integral, which is discussed in Section 7. Moreover, the ansatz (10) does not cover the series expansions of all special functions of interest. For example, in [25] a power series expansion for the digamma function $\psi(z)$ was analyzed whose truncation errors cannot be

approximated by a truncated power series of the type of (10). Nevertheless, the examples considered in this article should suffice to convince even a sceptical reader that the ansatz (10) is indeed computationally useful.

We cannot expect that the ansatz (10) satisfies the the inhomogeneous difference equation (9) exactly. However, we can choose the unspecified coefficients $\gamma_\mu^{(m)}$ in (10) in such a way that only a higher order error remains:

$$\frac{r_{n+1}^{(m)} - r_n^{(m)}}{a_{n+1}} = \sum_{\mu=0}^m \frac{\gamma_\mu^m}{(n + \alpha)^\mu} - \frac{a_{n+2}}{a_{n+1}} \sum_{\mu=0}^m \frac{\gamma_\mu^m}{(n + \alpha + 1)^\mu} \tag{11}$$

$$= 1 + \mathcal{O}(n^{-m-1}), \quad n \rightarrow \infty. \tag{12}$$

The approach of this article depends crucially on the assumption that the ratio a_{n+2}/a_{n+1} can be expressed as an (asymptotic) power series in $1/(n + \alpha)$. If this is the case, then the right-hand side of (11) can be expanded in powers of $1/(n + \alpha)$ and we obtain:

$$\frac{r_{n+1}^{(m)} - r_n^{(m)}}{a_{n+1}} = \sum_{\mu=0}^m \frac{\mathcal{C}_\mu^{(m)}}{(n + \alpha)^\mu} + \mathcal{O}(n^{-m-1}), \quad n \rightarrow \infty.$$

Now, (12) implies that we have solve the following system of linear equations:

$$\mathcal{C}_\mu^{(m)} = \delta_{\mu 0}, \quad 0 \leq \mu \leq m. \tag{13}$$

Since $\mathcal{C}_\mu^{(m)}$ with $0 \leq \mu \leq m$ contains only the unspecified coefficients $\gamma_0^{(m)}, \dots, \gamma_\mu^{(m)}$ but not $\gamma_{\mu+1}^{(m)}, \dots, \gamma_m^{(m)}$, the linear system (13) has a triangular structure and the unspecified coefficients $\gamma_0^{(m)}, \dots, \gamma_m^{(m)}$ can be determined by solving successively the equations $\mathcal{C}_0^{(m)} = 1, \mathcal{C}_1^{(m)} = 0, \dots, \mathcal{C}_m^{(m)} = 0$.

Another important aspect is that the linear equations (13) do not depend explicitly on m , which implies that the coefficients $\gamma_\mu^{(m)}$ in (10) also do not depend explicitly on m . Accordingly, the superscript m of both $\mathcal{C}_\mu^{(m)}$ and $\gamma_\mu^{(m)}$ is superfluous and will be dropped in the following Sections.

4 Numerical Analytic Continuation

Divergent asymptotic series of the type of (8) can be extremely useful computationally: For sufficiently large arguments z , truncated expansions of that kind are able to provide (very) accurate approximations to the corresponding special functions, in particular if the series is truncated in the vicinity of the minimal term. If, however, the argument z is small, truncated expansions of that kind produce only relatively poor or even completely nonsensical results.

We can expect that our asymptotic expansions in powers of $1/(n + \alpha)$ have similar properties. Thus, we can be confident that they produce (very) good results for sufficiently large indices n , but it would be overly optimistic to assume that these expressions necessarily produce good results in the nonasymptotic regime of moderately large or even small indices n .

Asymptotic approximants can often be constructed (much) more easily than other approximants that are valid in a wider domain. Thus, it is desirable to use

asymptotic approximants also outside the asymptotic domain. This means that we would like to use our asymptotic approximants also for small indices n in order to avoid the computationally problematic asymptotic regime of large indices. Obviously, this is intrinsically contradictory. We also must find a way of extracting additional information from the terms of a truncated divergent inverse power series expansion.

Often, this can be accomplished at low computational cost by converting an inverse power series $\sum_{n=0}^{\infty} c_n/z^n$ to a factorial series $\sum_{n=0}^{\infty} \tilde{c}_n/(z)_n$. Factorial series, which had already been known to Stirling [22, p. 6], frequently have superior convergence properties. An example is the incomplete gamma function $\Gamma(a, z)$, which possesses a divergent asymptotic series of the type of (8) [8, Eq. (6) on p. 135] and also a convergent factorial series [8, Eq. (1) on p. 139]. Accordingly, the otherwise so convenient inverse powers are not necessarily the computationally most effective asymptotic sequence.

The transformation of an inverse power series to a factorial series can be accomplished with the help of the Stirling numbers of the first kind which are normally defined via the expansion $(z-n+1)_n = \sum_{\nu=0}^n \mathbf{S}^{(1)}(n, \nu)z^\nu$ of a Pochhammer symbol in terms of powers. As already known to Stirling (see for example [22, p. 29] or [17, Eq. (6) on p. 78]), the Stirling numbers of the first kind occur also in the factorial series expansion of an inverse power:

$$\frac{1}{z^{k+1}} = \sum_{\kappa=0}^{\infty} \frac{(-1)^\kappa \mathbf{S}^{(1)}(k+\kappa, k)}{(z)_{k+\kappa+1}}, \quad k \in \mathbb{N}_0. \tag{14}$$

This infinite generating function can also be derived by exploiting the well known recurrence relationships of the Stirling numbers.

With the help of (14), the following transformation formula can be derived easily:

$$\sum_{n=0}^{\infty} \frac{c_n}{z^n} = c_0 + \frac{c_1}{(z)_1} + \sum_{k=2}^{\infty} \frac{(-1)^k}{(z)_k} \sum_{\kappa=1}^k (-1)^\kappa \mathbf{S}^{(1)}(k-1, \kappa-1) c_\kappa. \tag{15}$$

Let us now assume that the coefficients γ_μ with $0 \leq \mu \leq m$ of a truncated expansion of r_n/a_{n+1} in powers of $1/(n+\alpha)$ according to (10) are known. Then, (15) implies that we can use the transformation scheme

$$\tilde{\gamma}_\mu = \begin{cases} \gamma_\mu, & \mu = 0, 1, \\ \sum_{\nu=1}^{\mu} (-1)^{\mu+\nu} \mathbf{S}^{(1)}(\mu-1, \nu-1) \gamma_\nu, & \mu \geq 2, \end{cases}$$

to obtain instead of (10) the truncated factorial series

$$\tilde{r}_n^{(m)} = -a_{n+1} \left[\sum_{\mu=0}^m \frac{\tilde{\gamma}_\mu}{(n+\alpha)_\mu} + \mathcal{O}(n^{-m-1}) \right], \quad n \rightarrow \infty.$$

Padé approximants, which convert the partial sums of a formal power series to a doubly indexed sequence of rational functions, can also be quite helpful. They are now used almost routinely in applied mathematics and theoretical physics to overcome convergence problems with power series (see for example the monograph by Baker and Graves-Morris [2] and references therein). The ansatz (10) produces a

truncated series expansion of r_n/a_{n+1} in powers of $1/(n+\alpha)$, which can be converted to a Padé approximant, i.e., to a rational function in $1/(n+\alpha)$.

The numerical results presented in Sections 6 and 7 show that the conversion to factorial series and Padé approximants improves the accuracy of our asymptotic approximants, in particular for small indices n .

5 The Dirichlet Series for the Riemann Zeta Function

In this Section, an asymptotic approximation to the truncation error of the Dirichlet series for the Riemann zeta function is constructed by suitably adapting the approach described in Section 3. In the case of the Dirichlet series (1), we have:

$$\begin{aligned}
 s_n &= \sum_{\nu=0}^n (\nu+1)^{-s}, \\
 r_n &= - \sum_{\nu=n+1}^{\infty} (\nu+1)^{-s} = -(n+2)^{-s} \sum_{\nu=0}^{\infty} \left(1 + \frac{\nu}{n+2}\right)^{-s}, \\
 \Delta r_n &= (n+2)^{-s}.
 \end{aligned} \tag{16}$$

It is an obvious idea to express the infinite series on the right-hand side of (16) as a power series in $1/(n+2)$. If $\nu < n+2$, we can use the binomial series $(1+z)^a = {}_1F_0(-a; -z) = \sum_{m=0}^{\infty} \binom{a}{m} z^m$ [15, p. 38], which converges for $|z| < 1$. We thus obtain

$$[1 + \nu/(n+2)]^{-s} = \sum_{m=0}^{\infty} \frac{(s)_m}{m} [-\nu/(n+2)]^m.$$

The infinite series converges if $\nu/(n+2) < 1$. Thus, an expansion of the right-hand side of (16) in powers of $1/(n+2)$ can only be asymptotic as $n \rightarrow \infty$. Nevertheless, a suitably truncated expansion suffices for our purposes.

In the case of the Dirichlet series for the Riemann zeta function, we cannot use ansatz (10). This follows at once from the relationship

$$\Delta n^\alpha = (n+1)^\alpha - n^\alpha = \alpha n^{\alpha-1} + \mathcal{O}(n^{\alpha-2}), \quad n \rightarrow \infty.$$

Thus, we make the following ansatz, which takes into account the specific features of the Dirichlet series (1):

$$r_n^{(m)} = -(n+2)^{1-s} \sum_{\mu=0}^m \frac{\gamma_\mu}{(n+2)^\mu}, \quad m \in \mathbb{N}, \quad n \in \mathbb{N}_0. \tag{17}$$

This ansatz is inspired by the truncation error estimate for Levin’s u transformation [13] (see also [24, Section 7.3] or [26, Section IV]).

As in Section 3, the unspecified coefficients γ_μ are chosen in such a way that only a higher order error remains:

$$\Delta r_n^{(m)} = (n+2)^{-s} [1 + \mathcal{O}(n^{-m-1})], \quad n \rightarrow \infty.$$

For that purpose, we write:

$$\begin{aligned} \frac{r_{n+1}^{(m)} - r_n^{(m)}}{(n+2)^{1-s}} &= \sum_{\mu=0}^m \frac{\gamma_\mu}{(n+2)^\mu} - \left[\frac{n+3}{n+2} \right]^{1-s} \sum_{\mu=0}^m \frac{\gamma_\mu}{(n+3)^\mu} \\ &= \sum_{\mu=0}^m \frac{\gamma_\mu}{(n+2)^\mu} \{1 - [1 + 1/(n+2)]^{1-s-\mu}\}. \end{aligned} \tag{18}$$

With the help of the binomial series [15, p. 38], we obtain:

$$1 - [1 + 1/(n+2)]^{1-s-\mu} = \sum_{\lambda=0}^{\infty} \frac{(s + \mu - 1)_{\lambda+1}}{(\lambda + 1)!} \frac{(-1)^\lambda}{(n+2)^{\lambda+1}}. \tag{19}$$

Inserting (19) into (18) yields:

$$\begin{aligned} &\sum_{\mu=0}^m \frac{\gamma_\mu}{(n+2)^\mu} \{1 - [1 + 1/(n+2)]^{1-s-\mu}\} \\ &= \sum_{\mu=0}^m \frac{\gamma_\mu}{(n+2)^\mu} \sum_{\lambda=0}^{\infty} \frac{(s + \mu - 1)_{\lambda+1}}{(\lambda + 1)!} \frac{(-1)^\lambda}{(n+2)^{\lambda+1}} \\ &= - \sum_{\nu=0}^{\infty} (n+2)^{-\nu-1} \sum_{\lambda=0}^{\min(\nu, m)} \frac{(1 - s - \nu)_{\lambda+1} \gamma_{\nu-\lambda}}{(\lambda + 1)!}. \end{aligned}$$

Thus, we obtain the following truncated asymptotic expansion:

$$\begin{aligned} \frac{r_{n+1}^{(m)} - r_n^{(m)}}{(n+2)^{-s}} &= - \sum_{\mu=0}^m (n+2)^{-\mu} \\ &\times \sum_{\lambda=0}^{\mu} \frac{(1 - s - \mu)_{\lambda+1} \gamma_{\mu-\lambda}}{(\lambda + 1)!} + \mathcal{O}(n^{-m-1}), \quad n \rightarrow \infty. \end{aligned}$$

The unspecified coefficients γ_μ have to be determined by solving the following system of linear equations, whose triangular structure is obvious:

$$\sum_{\lambda=0}^{\mu} \frac{(1 - s - \mu)_{\lambda+1} \gamma_{\mu-\lambda}}{(\lambda + 1)!} = \delta_{\mu 0}, \quad 0 \leq \mu \leq m. \tag{20}$$

For a more detailed analysis of the linear system (20), let us define β_μ via

$$\gamma_\mu = (-1)^\mu \frac{(s)_{\mu-1}}{\mu!} \beta_\mu, \quad \mu \in \mathbb{N}_0. \tag{21}$$

Inserting (21) into (20) yields:

$$\sum_{\lambda=0}^{\mu} \frac{(1 - s - \mu)_{\lambda+1}}{(\lambda + 1)!} \frac{(-1)^{\mu-\lambda} (s)_{\mu-\lambda-1}}{(\mu - \lambda)!} \beta_{\mu-\lambda} = \delta_{\mu 0}, \quad \mu \in \mathbb{N}_0.$$

Next, we use $(1 - s - \mu)_{\lambda+1} = (-1)^{\lambda+1} (s)_{\lambda+1}$ and obtain

$$\begin{aligned} \sum_{\lambda=0}^{\mu} \frac{(1-s-\mu)_{\lambda+1} \gamma_{\mu-\lambda}}{(\lambda+1)!} &= (-1)^{\mu+1} (s)_{\mu} \sum_{\lambda=0}^{\mu} \frac{\beta_{\mu-\lambda}}{(\lambda+1)! (\mu-\lambda)!} \\ &= \frac{(-1)^{\mu+1} (s)_{\mu}}{(\mu+1)!} \sum_{\sigma=0}^{\mu} \frac{(\mu+1)!}{(\mu-\sigma+1)! \sigma!} \beta_{\sigma} \\ &= \frac{(-1)^{\mu+1} (s)_{\mu}}{(\mu+1)!} \sum_{\sigma=0}^{\mu} \binom{\mu+1}{\sigma} \beta_{\sigma} = \delta_{\mu 0}, \quad \mu \in \mathbb{N}_0. \end{aligned}$$

Thus, the linear system (20) is equivalent to the well known recurrence formula

$$\sum_{\nu=0}^n \binom{n+1}{\nu} B_{\nu} = 0, \quad n \in \mathbb{N}, \tag{22}$$

of the Bernoulli numbers (see for example [21, Eq. (1.11)]) together with the initial condition $B_0 = 1$. Thus, the ansatz (17) reproduces the finite sum (5a) or (7a) of the Euler-Maclaurin formula for the truncation error of the Dirichlet series, which is not really surprising since asymptotic series are unique, if they exist. Only the integral (5b) or (7b) cannot be reproduced in this way.

6 The Gaussian Hypergeometric Series

The simplicity of the terms of the Dirichlet series (1) facilitates the derivation of explicit asymptotic approximations to truncation errors by solving a system of linear equations in closed form. A much more demanding test for the feasibility of the new formalism is the Gaussian hypergeometric series (2), which depends on three parameters a, b , and c , and one argument z .

Due to the complexity of the terms of the Gaussian hypergeometric series (2), there is little hope in obtaining explicit analytical solutions to the linear equations. From a pragmatist's point of view, it is therefore recommendable to use computer algebra systems like Maple and Mathematica and let the computer do the work.

In the case of a nonterminating Gaussian hypergeometric series, we have:

$$\begin{aligned} s_n(z) &= \sum_{\nu=0}^n \frac{(a)_{\nu} (b)_{\nu}}{(c)_{\nu} \nu!} z^{\nu}, \\ r_n(z) &= - \sum_{\nu=n+1}^{\infty} \frac{(a)_{\nu} (b)_{\nu}}{(c)_{\nu} \nu!} z^{\nu} \\ &= - \frac{(a)_{n+1} (b)_{n+1}}{(c)_{n+1} (n+1)!} z^{n+1} \sum_{\nu=0}^{\infty} \frac{(a+n+1)_{\nu} (b+n+1)_{\nu}}{(c+n+1)_{\nu} (n+2)_{\nu}} z^{\nu}, \tag{23} \\ \Delta r_n(z) &= \frac{(a)_{n+1} (b)_{n+1}}{(c)_{n+1} (n+1)!} z^{n+1}. \end{aligned}$$

Since $[(a+n+1)_{\nu} (b+n+1)_{\nu}] / [(c+n+1)_{\nu} (n+2)_{\nu}]$ can be expressed as a power series in $1/(n+1)$, the following ansatz make sense:

$$r_n^{(m)}(z) = -\frac{(a)_{n+1}(b)_{n+1}}{(c)_{n+1}(n+1)!} z^{n+1} \sum_{\mu=0}^m \frac{\gamma_\mu}{(n+1)^\mu}, \quad (24)$$

$$m \in \mathbb{N}, \quad n \in \mathbb{N}_0, \quad |z| < 1.$$

Again, we choose the unspecified coefficients γ_μ in (24) in such a way that only a higher order error error remains:

$$\begin{aligned} \Delta r_n^{(m)}(z) &= r_{n+1}^{(m)}(z) - r_n^{(m)}(z) \\ &= \frac{(a)_{n+1}(b)_{n+1}}{(c)_{n+1}(n+1)!} z^{n+1} [1 + \mathcal{O}(n^{-m-1})], \quad n \rightarrow \infty. \end{aligned}$$

This convergence condition can be reformulated as follows:

$$\begin{aligned} &\frac{r_{n+1}^{(m)}(z) - r_n^{(m)}(z)}{[(a)_{n+1}(b)_{n+1}z^{n+1}]/[(c)_{n+1}(n+1)!]} \\ &= \sum_{\mu=0}^m \frac{\gamma_\mu}{(n+1)^\mu} - \frac{(a+n+1)(b+n+1)}{(c+n+1)(n+2)} z \sum_{\mu=0}^m \frac{\gamma_\mu}{(n+2)^\mu} \quad (25) \\ &= 1 + \mathcal{O}(n^{-m-1}), \quad n \rightarrow \infty. \end{aligned}$$

Now, we only have to do an asymptotic expansion of (25) in terms of the asymptotic sequence $\{1/(n+1)^j\}_{j=0}^\infty$ as $n \rightarrow \infty$. This yields:

$$\begin{aligned} &\frac{r_{n+1}^{(m)}(z) - r_n^{(m)}(z)}{[(a)_{n+1}(b)_{n+1}z^{n+1}]/[(c)_{n+1}(n+1)!]} \\ &= \sum_{\mu=0}^m \frac{\mathcal{C}_\mu}{(n+1)^\mu} + \mathcal{O}(n^{-m-1}), \quad n \rightarrow \infty. \quad (26) \end{aligned}$$

We then obtain the following system of coupled linear equations in the unspecified coefficients γ_μ with $0 \leq \mu \leq m$:

$$\mathcal{C}_\mu = \delta_{\mu 0}, \quad 0 \leq \mu \leq m. \quad (27)$$

As discussed in Section 3, a coefficient \mathcal{C}_μ with $0 \leq \mu \leq m$ contains only the unspecified coefficients $\gamma_0, \dots, \gamma_\mu$ but not $\gamma_{\mu+1} \dots, \gamma_m$. Thus, the symbolic solution of these linear equations for a Gaussian hypergeometric function ${}_2F_1(a, b; c; z)$ with unspecified parameters a, b , and c and unspecified argument z is not particularly difficult for a computer algebra system, since the unspecified coefficient γ_μ can be determined successively. The following linear equations were constructed with the help of Maple 8:

$$\mathcal{C}_0 = (1-z)\gamma_0 = 1, \quad (28a)$$

$$\mathcal{C}_1 = (c-a-b+1)z\gamma_0 + (1-z)\gamma_1 = 0, \quad (28b)$$

$$\begin{aligned} \mathcal{C}_2 &= [(c-b+1)a + (c+1)b - 1 - c - c^2]z\gamma_0 \\ &\quad + (c+2-b-a)z\gamma_1 + (1-z)\gamma_2 = 0, \quad (28c) \end{aligned}$$

$$\begin{aligned} \mathcal{C}_3 &= \{[(c+1)b - 1 - c - c^2]a - (1+c+c^2)b + c^3 + c^2 + c + 1\}z\gamma_0 \\ &\quad + [(c+2-b)a + (c+2)b - 3 - c^2 - 2c]z\gamma_1 \\ &\quad + (3-b-a+c)z\gamma_2 + (1-z)\gamma_3 = 0. \quad (28d) \end{aligned}$$

This example shows that the complexity of the coefficients C_μ in (26) increases so rapidly with increasing index μ that a solution of the linear equations (27) becomes soon unmanageable for humans. This is also confirmed by the following solutions of (28) obtained symbolically with the help of Maple 8:

$$(z - 1) \gamma_0 = 1, \tag{29a}$$

$$(z - 1)^2 \gamma_1 = z(a + b - c - 1), \tag{29b}$$

$$(z - 1)^3 \gamma_2 = z \{ [a^2 + (b - c - 2)a + b^2 - (c + 2)b + 1 + 2c] z + (b - c - 1)a - (c + 1)b + 1 + c + c^2 \}, \tag{29c}$$

$$(z - 1)^4 \gamma_3 = z \{ [a^3 + (b - c - 3)a^2 + (b^2 - (c + 3)b + 3 + 3c)a + b^3 - (c + 3)b^2 + (3 + 3c)b - 1 - 3c] z^2 + [(2b - 2c - 3)a^2 + (2b^2 - (4c + 8)b + 2c^2 + 7 + 8c)a - (2c + 3)b^2 + (2c^2 + 7 + 8c)b - 4 - 5c^2 - 7c] z + [-(c + 1)b + 1 + c^2 + c] a + (1 + c^2 + c)b - 1 - c^2 - c - c^3 \}. \tag{29d}$$

The solutions (29), which are rational in z , demonstrate quite clearly a principal weakness of symbolic computing. Typically, the results are complicated and poorly structured algebraic expressions, and it is normally very difficult to gain further insight from them. Nevertheless, symbolic solutions of the linear equations (27) are computationally very useful.

For the Gaussian hypergeometric series with $a = 1/3$, $b = 7/5$, $c = 9/2$, and $z = -0.85$, Maple 8 produced for $m = 8$ and $n = 1$ the following results:

$$\begin{aligned} r_1 &= -0.016\ 412\ 471, \\ a_2 [4/4] &= -0.016\ 410\ 482, \\ a_2 \tilde{r}_1^{(8)} &= -0.016\ 414\ 203, \\ a_2 r_1^{(8)} &= -0.004\ 008\ 195. \end{aligned}$$

It is in my opinion quite remarkable that for $n = 1$, which is very far away from the asymptotic regime, at least the Padé approximant $a_2[4/4]$ and the truncated factorial series $a_2 \tilde{r}_1^{(8)}$ agree remarkably well with the “exact” truncation error r_1 . In contrast, the truncated inverse power series $a_2 r_1^{(8)}$ produces a relatively poor result. For $n = 10$, which possibly already belongs to the asymptotic regime, Maple 8 produced the following results:

$$\begin{aligned} r_{10} &= 0.000\ 031\ 925\ 482, \\ a_{11} [4/4] &= 0.000\ 031\ 925\ 482, \\ a_{11} \tilde{r}_{10}^{(8)} &= 0.000\ 031\ 925\ 483, \\ a_{11} r_{10}^{(8)} &= 0.000\ 031\ 925\ 471. \end{aligned}$$

Finally, let me emphasize that the formalism of this article is not limited to a Gaussian hypergeometric series (2), but works just as well in the case of a generalized hypergeometric series ${}_{p+1}F_p(\alpha_1, \dots, \alpha_{p+1}; \beta_1, \dots, \beta_p; z)$.

7 The Asymptotic Series for the Exponential Integral

The divergent asymptotic series (3) for the exponential integral $E_1(z)$ is probably the most simple model for many other factorially divergent asymptotic inverse power series occurring in special function theory. Well known examples are the asymptotic series for the modified Bessel function $K_\nu(z)$, the complementary error function $\operatorname{erfc}(z)$, the incomplete gamma function $\Gamma(a, z)$, or the Whittaker function $W_{\kappa, \mu}(z)$. Moreover, factorial divergence is also the rule rather than the exception among the perturbation expansions of quantum physics (see [26] for a condensed review of the relevant literature).

The exponential integral $E_1(z)$ can also be expressed as a Stieltjes integral:

$$z e^z E_1(z) = \int_0^\infty \frac{e^{-t} dt}{1 + t/z}. \tag{32}$$

If $z < 0$, this integral has to be interpreted as a principal value integral.

In the case of a factorially divergent inverse power series, it is of little use to represent the truncation error $r_n(z)$ by a power series as in (23). If, however, we use $\sum_{\nu=0}^n x^\nu = [1 - x^{n+1}]/[1 - x]$ in (32), we immediately obtain:

$$\begin{aligned} s_n(z) &= \sum_{\nu=0}^n (-1/z)^\nu \nu!, \\ r_n(z) &= -(-z)^{-n-1} \int_0^\infty \frac{t^{n+1} e^{-t} dt}{1 + t/z}, \\ \Delta r_n(z) &= (-1/z)^{n+1} (n+1)!. \end{aligned}$$

Because of the factorial growth of the coefficients in (3), it is advantageous to use instead of (10) the following ansatz:

$$r_n^{(m)}(z) = -(-1/z)^n n! \sum_{\mu=0}^m \frac{\gamma_\mu}{(n+1)^\mu}, \quad m \in \mathbb{N}, \quad n \in \mathbb{N}_0. \tag{33}$$

Again, we choose the unspecified coefficients γ_μ in (33) in such a way that only a higher order error remains:

$$\Delta r_n^{(m)}(z) = (-1/z)^{n+1} (n+1)! [1 + \mathcal{O}(n^{-m-1})], \quad n \rightarrow \infty.$$

This convergence condition can be reformulated as follows:

$$\begin{aligned} \frac{r_{n+1}^{(m)}(z) - r_n^{(m)}(z)}{(-1/z)^{n+1} (n+1)!} &= \frac{-z}{n+1} \sum_{\mu=0}^m \frac{\gamma_\mu}{(n+1)^\mu} - \sum_{\mu=0}^m \frac{\gamma_\mu}{(n+2)^\mu} \\ &= 1 + \mathcal{O}(n^{-m-1}), \quad n \rightarrow \infty. \end{aligned} \tag{34}$$

Next, we do an asymptotic expansion of the right-hand side of (34) in terms of the asymptotic sequence $\{1/(n+1)^j\}_{j=0}^\infty$ as $n \rightarrow \infty$. This yields:

$$\frac{r_{n+1}^{(m)}(z) - r_n^{(m)}(z)}{(-1/z)^{n+1} (n+1)!} = \sum_{\mu=0}^m \frac{C_\mu}{(n+1)^\mu} + \mathcal{O}(n^{-m-1}), \quad n \rightarrow \infty.$$

Again, we have to solve the following system of linear equations:

$$C_\mu = \delta_{\mu 0}, \quad 0 \leq \mu \leq m.$$

The following linear equations were constructed with the help of Maple 8:

$$C_0 = -\gamma_0 = 1, \tag{35a}$$

$$C_1 = -z\gamma_0 - \gamma_1 = 0, \tag{35b}$$

$$C_2 = (1 - z)\gamma_1 - \gamma_2 = 0, \tag{35c}$$

$$C_3 = -\gamma_1 + (2 - z)\gamma_2 - \gamma_3 = 0, \tag{35d}$$

$$C_4 = \gamma_1 - 3\gamma_2 + (3 - z)\gamma_3 - \gamma_4 = 0. \tag{35e}$$

If we compare the complexity of the equations (35) with those of (28), we see that in the case of the asymptotic series (3) for the exponential integral there may be a chance of finding explicit expressions for the coefficients γ_μ . At least, the solutions of the linear system (35) obtained symbolically with the help of Maple 8 look comparatively simple:

$$\gamma_0 = -1, \tag{36a}$$

$$\gamma_1 = z, \tag{36b}$$

$$\gamma_2 = -(z - 1)z, \tag{36c}$$

$$\gamma_3 = (z^2 - 3z + 1)z, \tag{36d}$$

$$\gamma_4 = -(z^3 - 6z^2 + 7z - 1)z. \tag{36e}$$

Of course, this requires further investigations.

The relative simplicity of the coefficients in (36) offers other perspectives. For example, the [2/2] Padé approximant to the truncated power series in (33) is compact enough to be printed without problems:

$$[2/2] = \frac{-1 + \frac{3-z}{n+1} - \frac{2}{(n+1)^2}}{1 + \frac{2z-3}{n+1} + \frac{z^2-2z+2}{(n+1)^2}} = \frac{n^2 - n + zn + z}{n^2 - n + 2zn + z^2}.$$

Padé approximants to the truncated power series in (33) seem to be a new class of approximants that are rational in both n and z .

For the asymptotic series (3) for $E_1(z)$ with $z = 5$, Maple 8 produced for $m = 16$ and $n = 2$ the following results:

$$\begin{aligned} r_2 &= 0.027\,889, \\ a_2 [8/8] &= 0.027\,965, \\ a_2 \tilde{r}_2^{(16)} &= 0.028\,358, \\ a_2 r_2^{(16)} &= -177.788. \end{aligned}$$

The Padé approximant $a_2 [8/8]$ and the truncated factorial series $a_2 \tilde{r}_2^{(16)}$ agree well with the “exact” truncation error r_2 , but the truncated inverse power series $a_2 r_2^{(16)}$ is way off. For $n = 10$, all results agree reasonably well

$$\begin{aligned}
r_{10} &= 0.250\,470\,879, \\
a_{10} [8/8] &= 0.250\,470\,882, \\
a_{10} \tilde{r}_{10}^{(16)} &= 0.250\,470\,902, \\
a_{10} r_{10}^{(16)} &= 0.250\,470\,221.
\end{aligned}$$

8 Conclusions and Outlook

A new formalism is proposed that permits the construction of asymptotic approximations to truncation errors $r_n = -\sum_{\nu=n+1}^{\infty} a_{\nu}$ of infinite series for special functions by solving a system of linear equations. Approximations to truncation errors of monotone series can be obtained via the Euler-Maclaurin formula. The formalism proposed here is, however, based on different assumptions and can be applied even if the terms of the series have a comparatively complicated structure. In addition, the new formalism works also in the case of alternating and even divergent series.

Structurally, the asymptotic approximations of this article resemble the asymptotic inverse power series for special functions as $z \rightarrow \infty$, since they are not expansions of r_n , but rather expansions of ratios like r_n/a_{n+1} , r_n/a_n , or $r_n/[(n+\alpha)a_{n+1}]$ with $\alpha > 0$. This is consequential, because it makes it possible to use the convenient asymptotic sequence $\{1/(n+\alpha)^j\}_{j=0}^{\infty}$ of inverse powers. This greatly facilitates the necessary analytical manipulations and ultimately leads to comparatively simple systems of linear equations.

As shown in Section 5, the new formalism reproduces in the case of the Dirichlet series (1) for the Riemann zeta function the expressions (5a) or (7a) that follow from the Euler-Maclaurin formula. The linear equations (20) are equivalent to the recurrence formula (22) of the Bernoulli numbers. Thus, only the integral (5b) or (7b) cannot be obtained in this way.

Much more demanding is the Gaussian hypergeometric series (2), which is discussed in Section 6. The terms of this series depend on three in general complex parameters a , b , and c and one argument z . Accordingly, there is little hope that we might succeed in finding an explicit solution to the linear equations. However, all linear equations considered in this article have a triangular structure. Consequently, it is relatively easy to construct solutions symbolically with the help of a computer algebra system like Maple. The numerical results presented in Section 6 also indicate that the formalism proposed in this article is indeed computationally useful.

As a further example, the divergent asymptotic series (3) for the exponential integral $E_1(z)$ is considered in Section 7. The linear equations are again solved symbolically by Maple. Numerical results are also presented. This example is important since it shows that the new formalism works also in the case of factorially divergent series. The Euler-Maclaurin formula can only handle convergent monotone series.

Although the preliminary results look encouraging, a definite assessment of the usefulness of the new formalism for the computation of special functions is not yet possible. This requires much more data. Consequently, the new formalism should be applied to other series expansions for special functions and the performance of the resulting approximations should be analyzed and compared with other computational approaches.

I suspect that in most cases it will be necessary to solve the linear equations symbolically with the help of a computer algebra system like Maple. Nevertheless, it

cannot be ruled out that at least for some special functions with sufficiently simple series expansions explicit analytical solutions to the linear equations can be found.

Effective numerical analytic continuation methods are of considerable relevance for the new formalism which produces asymptotic approximations. We cannot tacitly assume that these approximations provide good results outside the asymptotic regime, although it would be highly desirable to use them also for small indices. In Section 4, only factorial series and Padé approximants are mentioned, although many other numerical techniques are known that can accomplish such an analytic continuation. Good candidates are sequence transformations which are often more effective than the better known Padé approximants. Details can be found in books by Brezinski and Redivo Zaglia [7], Sidi [18], or Wimp [28], or in a review by the present author [24].

References

1. J.R. Airey: The “converging factor” in asymptotic series and the calculation of Bessel, Laguerre and other functions. *Philos. Mag.* **24**, 1937, 521–552.
2. G.A. Baker, Jr. and P. Graves-Morris: *Padé Approximants*. 2nd edition, Cambridge University Press, Cambridge, UK, 1996.
3. C.M. Bender and S.A. Orszag: *Advanced Mathematical Methods for Scientists and Engineers*. McGraw-Hill, New York, 1978.
4. C. Brezinski: *History of Continued Fractions and Padé Approximants*. Springer, Berlin, 1991.
5. C. Brezinski: Extrapolation algorithms and Padé approximations: a historical survey. *Appl. Numer. Math.* **20**, 1996, 299–318.
6. C. Brezinski: Convergence acceleration during the 20th century. *J. Comput. Appl. Math.* **122**, 2000, 1–21. Reprinted in: *Numerical Analysis 2000, Vol. 2: Interpolation and Extrapolation*, C. Brezinski (ed.), Elsevier, Amsterdam, 2000, 1–21.
7. C. Brezinski and M. Redivo Zaglia: *Extrapolation Methods*. North-Holland, Amsterdam, 1991.
8. A. Erdélyi, W. Magnus, F. Oberhettinger, and F.G. Tricomi: *Higher Transcendental Functions*. Vol. II, McGraw-Hill, New York, 1953.
9. L. Euler: *Institutiones calculi differentialis cum eius usu in analysi finitorum ac doctrina serium. Pars II.1. De transformatione serium*, Academia Imperialis Scientiarum Petropolitana, St. Petersburg (1755). Reprinted as vol. X of *Leonardi Euleri Opera Omnia, Seria Prima*. Teubner, Leipzig and Berlin, 1913.
10. A. Gil, J. Segura, and N. Temme: Computing special functions by using quadrature rules. *Numer. Algor.* **33**, 2003, 265–275.
11. D.L. Jagerman: *Difference Equations with Applications to Queues*. Marcel Dekker, New York, 2000.
12. K. Knopp: *Theorie und Anwendung der unendlichen Reihen*. Springer, Berlin, 1964.
13. D. Levin: Development of non-linear transformations for improving convergence of sequences. *Int. J. Comput. Math. B* **3**, 1973, 371–388.
14. D.W. Lozier and F.W. Olver: Numerical evaluation of special functions. In: *Mathematics of Computation 1943-1993: A Half-Century of Computational Mathematics*, W. Gautschi (ed.), vol. 48 of *Proc. Symp. Appl. Math.*, American Mathematical Society, Providence, 1994, 79–125.

15. W. Magnus, F. Oberhettinger, and R.P. Soni: *Formulas and Theorems for the Special Functions of Mathematical Physics*. Springer, New York, 1966.
16. J.C.P. Miller: A method for the determination of converging factors, applied to the asymptotic expansions for the parabolic cylinder function. *Proc. Cambridge Phil. Soc.* **48**, 1952, 243–254.
17. N. Nielsen: *Die Gammafunktion*. Chelsea, New York, 1965. Originally published by Teubner, Leipzig and Berlin, 1906.
18. A. Sidi: *Practical Extrapolation Methods*. Cambridge University Press, Cambridge, UK, 2003.
19. D.A. Smith and W.F. Ford: Acceleration of linear and logarithmic convergence. *SIAM J. Numer. Anal.* **16**, 1979, 223–240.
20. J. Stirling: *Methodus differentialis sive tractatus de summatione et interpolatione serium infinitarum*, London, 1730. English translation by F. Holliday: *The Differential Method, or, a Treatise Concerning the Summation and Interpolation of Infinite Series*, London, 1749.
21. N.M. Temme: *Special Functions – An Introduction to the Classical Functions of Mathematical Physics*. Wiley, New York, 1996.
22. I. Tweddle: *James Stirling’s Methodus Differentialis: An Annotated Translation of Stirling’s Text*. Springer, London, 2003.
23. C.G. van der Laan and N.M. Temme: *Calculation of Special Functions: The Gamma Function, the Exponential Integrals and Error-Like Functions*. Centrum voor Wiskunde en Informatica, Amsterdam, 1980.
24. E.J. Weniger: Nonlinear sequence transformations for the acceleration of convergence and the summation of divergent series. *Comput. Phys. Rep.* **10**, 1989, 189–371. Los Alamos Preprint math-ph/0306302 (<http://arXiv.org>).
25. E.J. Weniger: A rational approximant for the digamma function. *Numer. Algor.* **33**, 2003, 499–507.
26. E.J. Weniger: Mathematical properties of a new Levin-type sequence transformation introduced by Čížek, Zamastil, and Skála. I. Algebraic theory, *J. Math. Phys.* **45**, 2004, 1209–1246.
27. E.J. Weniger and B. Kirtman: Extrapolation methods for improving the convergence of oligomer calculations to the infinite chain limit of quasi-onedimensional stereoregular polymers. *Comput. Math. Applic.* **45**, 2003, 189–215.
28. J. Wimp: *Sequence Transformations and Their Applications*. Academic Press, New York, 1981.

Strictly Positive Definite Functions on Generalized Motion Groups

Wolfgang zu Castell and Frank Filbir

Institute of Biomathematics and Biometry, GSF - National Research Center for Environment and Health, D-85764 Neuherberg, Germany
{castell,filbir}@gsf.de

Summary. Strictly positive definite functions are used as basis functions for approximation methods in various contexts. Using an interpretation of Bochner's theorem from abstract harmonic analysis we give a sufficient condition for strictly positive definite functions on generalized motion groups. As an example we consider reflection invariant functions on Euclidean spaces.

1 Introduction

Interpolation of scattered data based on positive definite functions has become a well-established method in applied mathematics. There are two independent approaches, leading to the same type of interpolation method. One comes from interpolation of spatial data with an isotropic and stationary random field model, known as *kriging* (cf. [11]). The second approach is based on what is known as *radial basis function interpolation* in the Euclidean space \mathbb{R}^d (cf. [3]). In the meantime a lot of results have been extended to the sphere \mathcal{S}^{d-1} and, in some extend, to more abstract spaces (e.g. [1, 7, 16]).

To be able to unify different approaches, let us for the moment assume that X is a topological space and $\phi : X \times X \rightarrow \mathbb{C}$ a complex-valued, continuous function. Then ϕ is called *positive definite* on X , if for any set of finitely many pairwise distinct points $x_1, \dots, x_n \in X$ and arbitrary complex coefficients $c_1, \dots, c_n \in \mathbb{C}$, the inequality

$$\sum_{j,k=1}^n c_j \bar{c}_k \phi(x_j, x_k) \geq 0 \tag{1}$$

holds true. Observe that our definition includes continuity of the function.

In practical applications it is usually not enough to assume the *basis function* ϕ to be positive definite. One rather needs the matrix

$$(\phi(x_j, x_k))_{j,k=1}^n$$

to be non-singular. This is guaranteed if the inequality (1) holds true in the strict sense for all non-zero coefficients c_1, \dots, c_n . Functions with this property are called *strictly positive definite*.

While characterizations for positive definite functions are known in many cases, necessary and sufficient conditions for strictly positive definite functions are rare. Recently, characterizations for the sphere have been given by Chen, Menegatto & Sun [7], and for ridge functions on real and complex inner product spaces by Pinkus [12, 13]. Sufficient conditions for several settings can be found for example in [1, 5, 6, 12].

Positive definite functions also play a fundamental role in the analysis of the structure of convolution algebras. Bochner's theorem, characterizing positive definite functions on \mathbb{R}^d as Fourier transforms of non-negative Borel measures has a generalization for locally compact Abelian groups and for Gelfand pairs. There is a nice theory of positive definite functions in abstract spaces. To understand strictly positive definiteness, one has to analyze the Bochner measure in more detail. Therefore, to analyze strictly positive definite functions, one not only needs to follow structural arguments, but also needs to take into account of the properties of the spaces the Bochner measure lives. This can hardly be done in the general context of locally compact groups since the topology on the group is explicitly involved.

In this paper we will present a sufficient condition for strictly positive definite functions on generalized motion groups. It is the analogue of a well-known condition for strictly positive definiteness on Euclidean spaces and the sphere (cf. [3, 6, 16]). As an example, we will give an application for strictly positive definite, reflection invariant functions on \mathbb{R}^d . Although, one can easily derive a stronger condition in the latter setting, our main aim of the present paper is to make the abstract setting accessible for researchers mainly interested in applications.

To keep the paper self-contained, we recall some basic facts of harmonic analysis on Gelfand pairs in Section 2. A more detailed exposition can be found in [2, 8, 9]. In Section 3, we will concentrate on generalized motion groups, stating and proving the main result of the paper. In the final section we will consider reflection invariant functions on Euclidean spaces.

2 Interpolation of Scattered Data

Let X be a locally compact Hausdorff space and $\mathcal{X} = \{x_1, \dots, x_N\}$ a set of pairwise distinct points in X . Given values f_1, \dots, f_N of an (unknown) function $f : X \rightarrow \mathbb{C}$ at the points in \mathcal{X} we want to recover f from this data. The *basis function method* uses a positive definite function $\phi : X \times X \rightarrow \mathbb{C}$ to set up the model

$$s_f(y) = \sum_{j=1}^N a_j \phi(y, x_j), \quad y \in X,$$

defined by the interpolation conditions $s_f(x_j) = f_j$, $1 \leq j \leq N$. To ensure that the collocation matrix $A = (\phi(x_j, x_k))_{j,k=1}^N$ is invertible, we want the function ϕ to be strictly positive definite. Usually, one further assumes the function ϕ to carry some symmetry properties. Typical choices for basis function models are functions of the form

- $\phi(|x - y|)$, $x, y \in \mathbb{R}^d$ (*radial basis functions*),
- $\phi(x^t y)$, $x, y \in \mathcal{S}^{d-1}$ (*zonal basis functions*),

- $\phi(x^{-1}y)$, $x, y \in G$, where G is a locally compact Abelian group.

To introduce symmetry into the basis function method let us assume there is a transformation group T acting on X . If X is a finite-dimensional vector space, T can be realized as matrix group acting on X via matrix-vector multiplication.

The action of T on X can naturally be extended to functions on X . A function $f : X \rightarrow \mathbb{C}$ is then called T -*(left)-invariant* if for all $x \in X$

$$f(\tau x) = f(x), \quad \forall \tau \in T.$$

In the above examples the symmetry groups are SO_d , SO_{d-1} , and $\{e\}$, i.e., the trivial subgroup of G , respectively. From now on let X be a locally compact group G . In this case, one can take T to be a subgroup of G which naturally acts on G by left-multiplication. Similarly, one can define the action of T on G by right-multiplication. A function which is invariant under both of these actions is called T -*biinvariant*, i.e., $f(\tau_1 x \tau_2) = f(x)$, for all $\tau_1, \tau_2 \in T$.

Note that although G needs not to be Abelian, it follows from the fact that positive definite functions on locally compact groups are Hermitian that positive definite, T -left-invariant functions on G are T -biinvariant.

The class of positive definite functions is closed under addition and multiplication with non-negative constants. It therefore has the structure of a cone. The cone is closed in the topology of pointwise convergence. Further on, every positive definite function ϕ on a group G is bounded, since $|\phi(x)| \leq \phi(e)$, for all $x \in G$.

The examples given so far are all coming from Gelfand pairs. Let us briefly recall the definition. Let G be a locally compact group and $f, g \in C_c(G)$ continuous functions on G with compact support. The *convolution* of f and g is then defined as

$$f * g(x) = \int_G f(y)g(y^{-1}x) dy, \quad x \in G,$$

where dy denotes the (left-) Haar measure on G . With this operation, the space $L^1(G)$ becomes a convolution algebra.

Let K be a compact subgroup of G and $L^1(G, K)$ the set of K -biinvariant functions in $L^1(G)$. Since the convolution of K -biinvariant functions is again biinvariant — this follows from the translation invariance of the Haar measure — $L^1(G, K)$ actually is a Banach subalgebra of $L^1(G)$. If this subalgebra is commutative, (G, K) is called a *Gelfand pair*.

A very important role in the analysis on Gelfand pairs is played by *spherical functions*. These are continuous K -biinvariant functions φ on G for which the linear functional

$$f \mapsto \chi_\varphi(f) = \int_G f(x)\varphi(x^{-1}) dx$$

defines a multiplicative functional on the space $L^1(G, K)$, i.e.,

$$\chi_\varphi(f * g) = \chi_\varphi(f)\chi_\varphi(g), \quad f, g \in L^1(G, K).$$

We denote the set of spherical functions by $(G, K)^\wedge$ and the subset of positive definite, spherical functions by $(G, K)_+^\wedge$.

We are now able to state Bochner's theorem characterizing positive definite, K -biinvariant functions.

Theorem 1. (cf. [9]) *Let (G, K) be a Gelfand pair and ϕ a continuous K -biinvariant function on G . Then ϕ is positive definite if and only if there is a bounded, non-negative Borel measure μ on $(G, K)_+^\wedge$ such that*

$$\phi(x) = \int_{(G, K)_+^\wedge} \varphi(x) d\mu(\varphi), \quad x \in G. \tag{2}$$

For simplicity, we call the measure μ *Bochner measure* associated with the positive definite function ϕ .

To give some examples, let us recall the well-known characterizations of positive definite functions given by Schoenberg [14, 15].

Example 1. The spherical functions for the Gelfand pair (M_d, SO_d) , where M_d is the group of affine transformations of \mathbb{R}^d , are given by the *spherical Bessel functions*

$$\mathcal{J}_{\frac{d-2}{2}}(\rho \cdot) = \Gamma\left(\frac{d}{2}\right) \left(\frac{\rho \cdot}{2}\right)^{-\frac{d-2}{2}} J_{\frac{d-2}{2}}(\rho \cdot), \quad \rho \in \mathbb{R}_+.$$

Therefore, the set $(G, K)_+^\wedge$ can be identified with the positive real line. Further note that SO_d -biinvariant functions on M_d can be identified with radial functions on \mathbb{R}^d . Bochner’s theorem for this case then reads as

Schoenberg [14]: *Let ϕ be a continuous, radial function on \mathbb{R}^d . ϕ is positive definite on \mathbb{R}^d if and only if there is a bounded, non-negative Borel measure μ on \mathbb{R}_+ such that*

$$\phi(t) = \int_{\mathbb{R}_+} \mathcal{J}_{\frac{d-2}{2}}(tu) d\mu(u), \quad t \in \mathbb{R}_+. \tag{3}$$

Example 2. Zonal functions on the sphere \mathcal{S}^{d-1} can be interpreted as SO_{d-1} -biinvariant functions on SO_d . The spherical functions for the Gelfand pair (SO_d, SO_{d-1}) are given by Gegenbauer polynomials $C_n^{\frac{d-2}{2}}$ on $[-1, 1]$, with parameter $\frac{d-2}{2}$. Note that SO_d is compact as is the double coset space $SO_d//SO_{d-1}$. In this case the set $(G, K)^\wedge$ is discrete. The measure μ in (2) is therefore supported in the set of non-negative integers.

Schoenberg [15]: *Let ϕ be a continuous, zonal function on $\mathcal{S}^{d-1} \times \mathcal{S}^{d-1}$. ϕ is positive definite on \mathcal{S}^{d-1} if and only if there are non-negative coefficients $(a_n)_{n \in \mathbb{N}}$ such that*

$$\phi(t) = \sum_{n=0}^{\infty} a_n C_n^{\frac{d-2}{2}}(t), \quad t \in [-1, 1].$$

Let us now further specialize the group G . The motivation for our specialization is the group of motions of \mathbb{R}^d , i.e., the group M_d . Let therefore A be a locally compact Abelian group and T a compact subgroup of the group of automorphisms of A . Then T acts on A via automorphisms, i.e., $T \times A \rightarrow A, (\tau, a) \mapsto a^\tau$. The *semi-direct product* of T and A , denoted by $T \rtimes A$, is the group defined by the group operations

$$\begin{aligned} (\tau, a)(\sigma, b) &= (\tau\sigma, a + b^\tau), & \tau, \sigma \in T, a, b \in A, \\ (\tau, a)^{-1} &= (\tau^{-1}, -a^{\tau^{-1}}), & \tau \in T, a \in A. \end{aligned}$$

The neutral element in $T \times A$ is given by the pair $(e, 0)$, where e and 0 denote the neutral elements in T and A , respectively. The groups T and A are naturally embedded in $T \times A$ via the mappings $T \rightarrow T \times A, \tau \mapsto (\tau, 0)$, and $A \rightarrow T \times A, a \mapsto (e, a)$. Even more, the subgroup A is a normal subgroup of $T \times A$. Groups of this type are called *generalized motion groups*.

Whenever there is no danger of confusion we will simply write $\tau \in T \times A$ and $a \in T \times A$, keeping the natural embedding in mind. Note that every element $(\tau, a) \in T \times A$ can be written as product $(\tau, a) = a\tau$, but since $T \times A$ is in general not Abelian, this product does not commute.

Recall that $(T \times A, T)$ is a Gelfand pair (cf. [8, (22.6.3), Ex. 3]). The harmonic analysis on $(T \times A, T)$ is governed by the harmonic analysis on A . Let α be a *character* of the Abelian group A , i.e., α defines a continuous homomorphism from A into the multiplicative group \mathbb{C}^* . Observe that this implies $\alpha(a^{-1}) = \overline{\alpha(a)}$, $a \in A$.

In [8, (22.6.12)] it is shown that the spherical functions for the pair $(T \times A, T)$ are given by the functions

$$\varphi(x) = \int_T \int_T \tilde{\alpha}_\varphi(\tau x \tau^{-1}) d\tau = \int_T \alpha_\varphi(a_x^\tau) d\tau, \quad x = (\tau_x, a_x) \in T \times A, \quad (4)$$

where $\tilde{\alpha}((\tau, a)) = \alpha(a)$ denotes the lifting of the function α to the group $T \times A$. Hereby, $d\tau$ denotes the left- and right-invariant Haar measure on the compact group T . Since the value of $\tilde{\alpha}(x)$ is independent of τ_x , where $x = (\tau_x, a_x) \in T \times A$, $\tilde{\alpha}$ naturally is T -right-invariant. Averaging the function over T thus defines a T -biinvariant function on $T \times A$.

It is straight forward to show that the sets $(T \times A)^\wedge$ and $(T \times A)_+^\wedge$ are equal in this case, i.e., every spherical function is positive definite. Theorem 1 then states

Corollary 1. *Let T, A be as above and ϕ be a continuous, T -biinvariant function on $T \times A$. Then ϕ is positive definite if and only if there is a bounded, non-negative Borel measure μ on $(T \times A, T)^\wedge$ such that*

$$\phi(x) = \int_{(T \times A, T)^\wedge} \varphi(x) d\mu(\varphi), \quad x \in T \times A. \quad (5)$$

Again, equation (3) is a special cases of (5), where $T = SO_d$ acts on the Abelian group \mathbb{R}^d . From the construction of the semi-direct product it is clear that $(T \times A)/T$ can be identified with A . The double coset space $M_d//SO_d$ can in this special case be identified with the positive real line. One can show that every spherical function carries a non-negative real parameter, i.e., the dual $(M_d, SO_d)^\wedge$ can also be identified with the positive real line. We therefore have in the case of radial functions on \mathbb{R}^d that

$$(M_d, SO_d)^\wedge \cong \mathbb{R}_+ \cong M_d//SO_d.$$

3 Strictly Positive Definite Functions on Semi-Direct Products

As mentioned in the introduction we have to analyze the measure μ in (5) in more detail. In order to do so, let us first come back to the space $L^1(T \times A)$.

Weil's formula for this case states that

$$\int_{T \times A} f(x) dx = \int_A \int_T f(a_x \tau_x) d\tau_x da_x,$$

where $x = (\tau_x, a_x) \in T \times A$ and $dx, d\tau_x,$ and da_x denote the Haar measure on $T \times A, T,$ and $A,$ respectively. Note that the argument of the function in the integral on the right hand side has to be interpreted as product in $T \times A.$

If $f \in L^1(T \times A, T)$ the integral over the group T is equal to one and we can apply Weil's formula again to obtain

$$\begin{aligned} \int_{T \times A} f(x) dx &= \int_A f(a_x) da_x = \int_{(T \times A) // T} \int_T f(\tau a_x) d\tau d(Ta_x T) \\ &= \int_{(T \times A) // T} f(a_x) d(Ta_x T), \end{aligned}$$

where the latter integral is over the set of all double cosets of the type $TaT, a \in A,$ endowed with the quotient topology.

Since the algebra $L^1(T \times A, T)$ is a commutative Banach algebra there is a Fourier transform on the space $L^1(T \times A, T),$ called *Gelfand transform.* It is defined for $f \in L^1(T \times A, T)$ by

$$\widehat{f}(\varphi) = \int_{T \times A} f(x) \overline{\varphi(x)} dx, \quad \varphi \in (T \times A, T)^\wedge.$$

The mapping $\widehat{\cdot} : L^1(T \times A, T) \rightarrow C_0((T \times A, T)^\wedge)$ is an algebra homomorphism, since $\widehat{f * g} = \widehat{f} \widehat{g},$ where $f, g \in L^1(T \times A, T).$

Using the measure $d(TaT)$ on the set of double cosets, the Gelfand transform can be extended to the space $L^2(T \times A, T)$ analogously as for the classical Fourier transform. There is a measure π on $(T \times A, T)^\wedge$ such that for all $f \in L^2(T \times A, T)$ the *Plancherel theorem* (cf. [8, (22.7.4)]) holds, i.e.,

$$\int_{(T \times A) // T} |f(a)|^2 d(TaT) = \int_{(T \times A, T)^\wedge} |\widehat{f}(\varphi)|^2 d\pi(\varphi).$$

Clearly, the support of π is a subset of $(T \times A, T)^\wedge.$ But in contrast to classical harmonic analysis it can indeed be a proper subset in the case of Gefand pairs. Nevertheless, for generalized motion groups, the measure π is the projection of the Plancherel measure on the dual group \widehat{A} of the Abelian component A and, thus, the support of π is the full set $(T \times A, T)^\wedge$ (cf. [2, Sec. 3.2]). In analogy to the group case, we call π *Plancherel measure.*

We can decompose the Bochner measure μ in (5) with respect to the Plancherel measure π into

$$\mu = \mu_{ac} + \mu_d + \mu_{sc}, \tag{6}$$

where μ_{ac} is absolutely continuous w.r.t. π, μ_d is the discrete part of the singular part of $\mu,$ while μ_{sc} is the continuous part of the singular part of $\mu.$ Since μ_{ac} is absolutely continuous w.r.t. $\pi,$ there is a function $h_\mu \in L^1((T \times A, T)^\wedge, \pi),$ such that for all measurable sets E we have

$$\mu_{ac}(E) = \int_E h_\mu d\pi. \tag{7}$$

We are now able to state the main theorem.

Theorem 2. *Let A and T be as above and ϕ be a continuous positive definite, T -bilinear function on $T \times A$. Let further $h_\mu \in L^1((T \times A, T)^\wedge, \pi)$ be the representative of the absolutely continuous part of the Bochner measure μ associated with ϕ . If the function h_μ is strictly positive for all φ in the support of π , then ϕ is strictly positive definite.*

Proof. Assume there is a set of n points x_1, \dots, x_n points in $T \times A$ and non-zero coefficients $c_1, \dots, c_n \in \mathbb{C}$, such that

$$\sum_{j,k=1}^n c_j \bar{c}_k \phi(x_j^{-1} x_k) = 0.$$

Without loss of generality we can assume the points x_1, \dots, x_n to lie in distinct double cosets, i.e., $Tx_jT \neq Tx_kT$ for $j \neq k$. From Corollary 1 and (4) it then follows — using $\alpha_\varphi(a^{-1}) = \alpha_\varphi(a)$ and the notation $x_k = (\tau_k, a_k)$ — that

$$\int_{(T \times A, T)^\wedge} \int_T \left| \sum_{k=1}^n \bar{c}_k \alpha_\varphi(a_k^\tau) \right|^2 d\tau d\mu(\varphi) = 0.$$

Decomposing the measure μ according to (6) we can conclude that the following integral equals zero

$$\int_{(T \times A, T)^\wedge} \int_T \left| \sum_{k=1}^n \bar{c}_k \alpha_\varphi(a_k^\tau) \right|^2 d\tau h_\mu(\varphi) d\pi(\varphi).$$

Since $h_\mu > 0$ on the support of π and the latter equals $(T \times A, T)^\wedge$, we have that

$$\sum_{k=1}^n \bar{c}_k \alpha(a_k) = 0,$$

for all α in \hat{A} . It is a consequence of the Gelfand-Raikov theorem that the point evaluation functionals are linearly independent on the dual group \hat{A} , thus $c_1 = \dots = c_n = 0$, which is a contradiction. \square

Remark 1. The proof shows that the criterion for strictly positive definiteness is based on a statement about linear independence of point evaluation functionals on the space of characters \hat{A} . Knowing more about the space \hat{A} allows to derive stronger conditions. For example, if this space is parameterized by a subset Ω of \mathbb{C}^d with non-empty interior, and if the characters are analytic functions on Ω , then it is enough to assume that h_μ is strictly positive on an open subset of Ω to ensure linear independence. Thus, the question arises, how small sets can be such that the characters α are linearly independent as functions on such sets (see for example [5]). Analyticity of the characters in a domain $\Omega \subset \mathbb{C}^d$ is clearly given in the following example.

4 Reflection Invariant Functions

We now want to apply the abstract theory in a concrete example. Our aim is to derive a sufficient condition for positive definite, reflection invariant functions on

\mathbb{R}^d . We will first recall some basic facts on reflection groups. For a more detailed treatment the reader is referred to the book by Humphreys [10].

Given a vector $v \in \mathbb{R}^d$, a mapping

$$\sigma_v : \mathbb{R}^d \rightarrow \mathbb{R}^d, \quad x \mapsto x - 2 \frac{x^t v}{v^t v} v$$

is called a *reflection*. Geometrically speaking the mapping reflects the space \mathbb{R}^d along the hyperplane defined by the normal vector $v \in \mathbb{R}^d$. A group of matrices generated by a finite set of reflections is called (*finite*) *reflection group*.

Let W_d denote a finite reflection group on \mathbb{R}^d . The action of the group can naturally be extended to an action on functions on \mathbb{R}^d via $\sigma_v f(x) = f(x^{\sigma_v})$, $x \in \mathbb{R}^d$. A function f is then called *reflection invariant*, or W_d -*invariant*, if $\sigma_v f = f$ for all $\sigma_v \in W_d$. Whenever there is no danger of confusion we will drop the index v indicating the reflecting hyperplane.

The set of reflecting hyperplanes associated with a given reflection group W_d decomposes the space \mathbb{R}^d into a set of finitely many, connected cones. Let us fix one of these cones and denote its closure by \mathcal{W} . Then every point in \mathbb{R}^d is the unique image under a suitable reflection of a point in \mathcal{W} . The cone \mathcal{W} is called a *fundamental domain* for the action of W_d on \mathbb{R}^d . It is a minimal set in the sense that no point in \mathcal{W} is the image of another point in \mathcal{W} under a reflection in W_d .

Imitating the group theoretic interpretation of radial functions on \mathbb{R}^d , we will identify reflection invariant functions on \mathbb{R}^d with W_d -biinvariant functions on the semi-direct product $W_d \ltimes \mathbb{R}^d$. Since \mathbb{R}^d is an Abelian normal subgroup of $W_d \ltimes \mathbb{R}^d$ the latter is a motion group in the generalized sense.

Since W_d is compact and \mathbb{R}^d is Abelian, the pair $(W_d \ltimes \mathbb{R}^d, W_d)$ is a Gelfand pair and the set of spherical functions is exactly given by the set of functions

$$J_{W_d}(\mathbf{v}; \mathbf{x}) = \frac{1}{|W_d|} \sum_{\sigma \in W_d} e^{i\mathbf{v}^t(\mathbf{x}^\sigma)} \quad \mathbf{x}, \mathbf{v} \in \mathcal{W}.$$

Corollary 1 then immediately leads to a characterization of positive definite, reflection invariant functions on \mathbb{R}^d .

Theorem 3. (cf. [4]) *Let W_d be a finite reflection group and ϕ be a continuous, W_d -invariant function on \mathbb{R}^d . ϕ is positive definite if and only if there is a bounded, non-negative Borel measure μ on \mathcal{W} such that*

$$\phi(\mathbf{x}) = \int_{\mathcal{W}} J_{W_d}(\mathbf{v}; \mathbf{x}) d\mu(\mathbf{v}), \quad \mathbf{v} \in \mathcal{W}.$$

The Plancherel measure π in this case reduces to the Lebesgue measure on \mathcal{W} . Applying Theorem 2 then allows to formulate a sufficient condition for strictly positive definite functions on \mathcal{W} .

Corollary 2. *Let W_d be a finite reflection group and ϕ be a continuous, positive definite, W_d -invariant function on \mathbb{R}^d and μ be the associated Bochner measure on \mathcal{W} . If the function h_μ defined by equation (7) satisfies $h_\mu(\mathbf{v}) > 0$ for all $\mathbf{v} \in \mathcal{W}$, the function ϕ is strictly positive definite on \mathcal{W} .*

Theorem 2 is the generalization of the fact that a positive definite function on \mathbb{R}^d with strictly positive Fourier transform is strictly positive definite. A similar result on compact groups has been given by Allali and Przebinda [1] using representation theory.

References

1. M. Allali and T. Przebinda: Strictly positive definite functions on a compact group. *Proc. Amer. Math. Soc.* **129**, 2001, 1459–1462.
2. P. Bougerol: Un mini-cours sur les couples de Guelfand. *Publ. du Lab. de Statist. et Probab.*, No. 01-83, Université Paul Sabatier, Toulouse, 1983.
3. M.D. Buhmann: *Radial Basis Functions: Theory and Implementations*. Cambridge Monographs on Applied and Computational Mathematics **12**, Cambridge University Press, Cambridge, 2003.
4. W. zu Castell: Interpolation with reflection invariant positive definite functions. In: *Approximation Theory XI: Gatlinburg 2004*, C.K. Chui, M. Neamtu, and L.L. Schumaker (eds.), Nashboro Press, Brentwood, 2005, 105–120.
5. W. zu Castell, F. Filbir, and R. Szwarc: Strictly positive definite functions in \mathbb{R}^d . *J. Approx. Theory* **137**, 2005, 277–280.
6. K.-F. Chang: Strictly positive definite functions. *J. Approx. Theory* **87**, 1996, 148–158.
7. D. Chen, V.A. Menegatto, and X. Sun: A necessary and sufficient condition for strictly positive definite functions on spheres. *Proc. Amer. Math. Soc.* **131**, 2003, 2733–2740.
8. Dieudonné: *Éléments d'Analyse*, Vol. 5/6, Gauthier-Villars, Paris, 1975.
9. J. Faraut: Analyse harmonique sur les espaces hyperboliques. In: *Analyse Harmonique*, Les Cours du C.I.M.P.A., J.L. Clerc, P. Eymard, J. Faraut, M. Raïs, and R. Takahasi (eds.), Centre International de Mathématiques Pures et Appliquées, Nice, 1983, 315–446.
10. J.E. Humphreys: *Reflection Groups and Coxeter Groups*. Cambridge Studies in Advanced Mathematics **29**, Cambridge University Press, Cambridge, 1997.
11. G. Matheron: *Les Variables Régionalisées et leur Estimation*. Masson, Paris, 1965.
12. A. Pinkus: Strictly positive definite functions on a real inner product space. *Adv. Comp. Math.* **20**, 2004, 263–271.
13. A. Pinkus: Strictly Hermitian positive definite functions. *Journal d'Analyse Math.* **94**, 2004, 293–318.
14. I.J. Schoenberg: Metric spaces and completely monotone functions. *Ann. Math.* **39**(2), 1938, 811–841.
15. I.J. Schoenberg: Positive definite functions on spheres. *Duke Math. J.* **9**, 1942, 96–108.
16. Y. Xu and E.W. Cheney: Strictly positive definite functions on spheres. *Proc. Amer. Math. Soc.* **116**, 1992, 977–981.

Energy Estimates and the Weyl Criterion on Compact Homogeneous Manifolds

Steven B. Damelin¹, Jeremy Levesley², and Xingping Sun³

¹ Institute for Mathematics and its Applications, University of Minnesota, Minneapolis, MN 55455, U.S.A., damelin@ima.umn.edu

² Department of Mathematics, University of Leicester, Leicester LE1 7RH, UK, j11@mcs.le.ac.uk

³ Department of Mathematics, Missouri State University, Springfield, MO 65897, U.S.A., XSun@MissouriState.edu

Summary. The purpose of this paper is to demonstrate that a number of results concerning approximation, integration, and uniform distribution on spheres can be generalised to a much wider range of compact homogeneous manifolds. The essential ingredient is that certain types of invariant kernels on the manifold (the generalisation of zonal kernels on the sphere or radial kernels in euclidean space) have a spectral decomposition in terms of projection kernels onto invariant polynomial subspaces. In particular, we establish a Weyl's criterion on such manifolds and announce a discrepancy estimate that generalises some pertinent results of Damelin and Grabner.

1 Introduction

Let M be a $d \geq 1$ dimensional homogeneous space of a compact Lie group G embedded in \mathbb{R}^{d+r} for some $r \geq 0$. Then (see [6]), we may assume that $G \subset O(d+r)$, the orthogonal group on \mathbb{R}^{d+r} . Thus $M = \{gp : g \in G\}$ where $p \in M$ is a non-zero vector in \mathbb{R}^{d+r} . For technical reasons, we will assume that M is *reflexive*. That is, for any given $x, y \in M$, there exists $g \in G$ such that $gx = y$ and $gy = x$.

Let $d(x, y)$ be the geodesic distance between $x, y \in M$ induced by the embedding of M in \mathbb{R}^{d+r} (see [5] for details). On the spheres, this corresponds to the usual geodesic distance. A real valued function $\kappa(x, y)$ defined on $M \times M$ is called a positive definite kernel on M , if for every nonempty finite subset $Y \subset M$, and arbitrary real numbers $c_y, y \in Y$, we have

$$\sum_{x \in Y} \sum_{y \in Y} c_x c_y \kappa(x, y) \geq 0.$$

If the above inequality becomes strict whenever the points y are distinct, and not all the c_y are zero, then the kernel κ is called strictly positive definite. A kernel κ is called G -invariant if $\kappa(gx, gy) = \kappa(x, y)$ for all $x, y \in M$ and $g \in G$. For example, if

$M := S^d$, the d dimensional sphere realized as a subset of \mathbb{R}^{d+1} and $G := O(d+1)$, then all the G -invariant kernels have the form $\phi(xy)$, where $\phi : [-1, 1] \rightarrow \mathbb{R}$, and where xy denotes the usual inner product of x and y . A kernel of the form $\phi(xy)$ is often called a zonal kernel on the sphere in the literature.

Let μ be a G -invariant measure on M (which may be taken as an appropriately normalized ‘surface’ measure). Then, for two functions $f, g : M \rightarrow \mathbb{R}$, we define an inner product with respect to μ :

$$[f, g] = [f, g]_\mu := \int_M fg d\mu$$

and let $L_2(M)_\mu$ denote the space of all square integrable functions from M into \mathbb{R} with respect to the above inner product. In the usual way, we identify all functions as being equal in $L_2(M)_\mu$, if they are equal almost everywhere with respect to the measure μ .

Let $n \geq 0$ and P_n be the space of polynomials in $d+r$ variables of degree n restricted on M . Here, multiplication is taken pointwise on \mathbb{R}^{d+r} . The *harmonic polynomials* of degree n on M are $H_n := P_n \cap P_{n-1}^\perp$. We may always (uniquely) decompose H_n into irreducible G -invariant subspaces $H_{n,k}$, $k = 1, \dots, \nu_n$. Indeed, the uniqueness of the decomposition follows from the minimality of the G -invariant space, since a different decomposition would give subspaces contained in minimal ones leading to a contradiction.

Any G -invariant kernel κ , has an associated integral operator which we define by

$$T_\kappa f(x) = \int_M \kappa(x, y) f(y) d\mu(y).$$

Now, for $n \geq 0, k \geq 1$, let $Y_{n,k}^1, \dots, Y_{n,k}^{d_{n,k}}$ be any orthonormal basis for $H_{n,k}$, and set

$$Q_{n,k}(x, y) := \sum_{j=1}^{d_{n,k}} Y_{n,k}^j(x) Y_{n,k}^j(y).$$

Then $Q_{n,k}$ is the unique G -invariant kernel for the orthogonal projection $T_{Q_{n,k}}$ of $L_2(M)_\mu$ onto $H_{n,k}$ acting as

$$T_{Q_{n,k}} f(x) = \int_M Q_{n,k}(x, y) f(y) d\mu(y).$$

The symmetry of $Q_{n,k}$ in x and y implies that it is positive definite on M . In fact, for every nonempty finite subset $Y \subset M$, and arbitrary real numbers $c_y, y \in Y$, we have

$$\begin{aligned} \sum_{x \in Y} \sum_{y \in Y} c_x c_y Q_{n,k}(x, y) &= \sum_{j=1}^{d_{n,k}} \left(\sum_{x \in Y} c_x Y_{n,k}^j(x) \right) \left(\sum_{y \in Y} c_y Y_{n,k}^j(y) \right) \\ &= \sum_{j=1}^{d_{n,k}} \left(\sum_{x \in Y} c_x Y_{n,k}^j(x) \right)^2 \\ &\geq 0. \end{aligned}$$

We summarise a few basic facts about G -invariant kernels in the following lemma:

Lemma 1. *Let y, z be fixed points in M . Then*

- (a) $\int_M Q_{n,k}(y, x)Q_{n,k}(x, z)d\mu(x) = Q_{n,k}(y, z)$.
- (b) For all $x \in M$, we have $Q_{n,k}(x, x) = d_{n,k}$.
- (c) If κ is a G -invariant kernel, then for all pairs of $(x, y) \in M \times M$, we have $\kappa(x, y) = \kappa(y, x)$.
- (d) For all $(x, y) \in M \times M$, we have $|Q_{n,k}(x, y)| \leq Q_{n,k}(x, x)$.

Proof. Part (a) follows directly from the fact that $Q_{n,k}$ is the projection kernel from $L_2(M)_\mu$ onto $H_{n,k}$.

Part (b) is a consequence of the equation

$$Q_{n,k}(x, x) := \sum_{j=1}^{d_{n,k}} Y_{n,k}^j(x)Y_{n,k}^j(x).$$

Indeed, since $Q_{n,k}$ is G -invariant, $Q_{n,k}(x, x)$ is a constant function of x for all $x \in M$. Integrating the last equation over M and using the orthonormality of the $Y_{n,k}^j$, we then arrive at the desired result.

The proof of part (c) needs the reflexivity of M . Indeed, pick a $g \in G$ so that $gx = y$ and $gy = x$. Then

$$\kappa(x, y) = \kappa(gy, gx) = \kappa(y, x)$$

using the G -invariance of κ .

Part (d) follows from a standard positive definiteness argument. Indeed, for each fixed pair $(x, y) \in M \times M$, the positive definiteness of the kernel $Q_{n,k}$ implies that the matrix

$$\begin{pmatrix} Q_{n,k}(x, x) & Q_{n,k}(x, y) \\ Q_{n,k}(y, x) & Q_{n,k}(y, y) \end{pmatrix}$$

is nonnegative definite, which further implies that

$$(Q_{n,k}(x, x))(Q_{n,k}(y, y)) - (Q_{n,k}(x, y))(Q_{n,k}(y, x)) \geq 0.$$

Since $Q_{n,k}(x, x) = Q_{n,k}(y, y)$, by part (b), and $Q_{n,k}(x, y) = Q_{n,k}(y, x)$ by part (c), we have the desired inequality. \square

An important consequence of the development above is that each irreducible subspace is generated by the translates of a fixed element. For this result on the sphere S^d , see, for instance, [1].

Proposition 1. *Let $Y \in H_{n,k}$, $Y \neq 0$. Then $H_{n,k} = \text{span}\{Y(g) : g \in G\}$.*

Proof. It is clear that $V = \text{span}\{Y(g) : g \in G\}$ is a G -invariant subspace of $H_{n,k}$, and since Y is not zero this is a non-trivial subspace. But $H_{n,k}$ is irreducible, so that V cannot be a proper subspace of $H_{n,k}$. Thus $V = H_{n,k}$. \square

Lemma 2. *Let κ_1 and κ_2 be continuous G -invariant kernels. If M is a reflexive space, $T_{\kappa_1}T_{\kappa_2} = T_{\kappa_2}T_{\kappa_1}$.*

Proof. Let $f \in L_2(M)_\mu$. Then

$$\begin{aligned} [T_{\kappa_1} T_{\kappa_2} f](x) &= \int_M \kappa_1(x, y) \left\{ \int_M \kappa_2(y, z) f(z) d\mu(z) \right\} d\mu(y) \\ &= \int_M f(z) \left\{ \int_M \kappa_1(x, y) \kappa_2(y, z) d\mu(y) \right\} d\mu(z). \end{aligned}$$

Since the manifold is reflexive, there is a $g \in G$ which interchanges x and z . Thus,

$$\int_M \kappa_1(x, y) \kappa_2(y, z) d\mu(y) = \int_M \kappa_1(z, y) \kappa_2(y, x) d\mu(y),$$

so that

$$\begin{aligned} [T_{\kappa_1} T_{\kappa_2} f](x) &= \int_M f(z) \left\{ \int_M \kappa_1(z, y) \kappa_2(y, x) d\mu(y) \right\} d\mu(z) \\ &= \int_M \kappa_2(x, y) \left\{ \int_M \kappa_1(y, z) f(z) d\mu(z) \right\} d\mu(y) \\ &= [T_{\kappa_2} T_{\kappa_1} f](x), \end{aligned}$$

where the penultimate step uses Lemma 1 (c). The changes of order of integration are easy to justify since the kernels are continuous and $f \in L_2(M)_\mu$. \square

We are now able to show that a G -invariant kernel has a spectral decomposition in terms of projection kernels onto invariant polynomial subspaces. This is contained in the following theorem.

Theorem 1. *If M is a reflexive manifold, then any G -invariant kernel κ has the spectral decomposition*

$$\kappa(x, y) = \sum_{n=0}^{\infty} \sum_{k=1}^{\nu_n} a_{n,k}(\kappa) Q_{n,k}(x, y),$$

where

$$a_{n,k}(\kappa) = \frac{1}{d_{n,k}} \int_M \kappa(x, y) Q_{n,k}(x, y) d\mu(y), \quad n \geq 0, k \geq 1.$$

Here the convergence is in the topology of $L_2(M)_\mu$.

Proof. If $Y \in H_{n,k}$ then $T_{Q_{n,k}} Y = Y$. Thus

$$\begin{aligned} T_\kappa Y &= T_\kappa (T_{Q_{n,k}} Y) \\ &= T_{Q_{n,k}} (T_\kappa Y) \in H_{n,k}, \end{aligned}$$

since $T_{Q_{n,k}}$ is the orthogonal projection onto $H_{n,k}$. Here we have used Lemma 2.

Since T_κ is a symmetric operator, it can be represented on the finite dimensional subspace by a symmetric matrix. Either this matrix is the zero matrix, in which case all the pertinent $a_{n,k}(\kappa)$ are zero, or T_κ has a non-trivial range. Since the matrix is symmetric, it must have a non-zero real eigenvalue. Let γ be a nonzero eigenvalue of the matrix, and let Y be an associated eigenvector, i.e., $T_\kappa Y = \gamma Y$. This implies that, for any fixed $g \in G$, $Y(g \cdot)$ is also an eigenvector. In fact, we have

$$\begin{aligned} [T_\kappa Y(g\cdot)](x) &= \int_M \kappa(x, y)Y(gy)d\mu(y) \\ &= \int_M \kappa(x, g^{-1}y)Y(y)d\mu(g^{-1}y) \\ &= \int_M \kappa(gx, y)Y(y)d\mu(y), \end{aligned}$$

using the G -invariance of both κ and μ . But Y is an eigenvector of T_κ , so that

$$[T_\kappa Y(g\cdot)](x) = \gamma Y(gx).$$

Now, using Proposition 1 we see that $H_{n,k}$ is an eigenspace for T_κ with single eigenvalue γ . We can compute γ by evaluating T_κ on $Q_{n,k}(\cdot, y)$ for a fixed y :

$$\int_M \kappa(z, x)Q_{n,k}(x, y)d\mu(x) = \gamma Q_{n,k}(z, y).$$

Setting $z = y$ and using Lemma 1 (b) we have

$$\gamma = \frac{1}{d_{n,k}} \int_M \kappa(y, x)Q_{n,k}(x, y)d\mu(x),$$

and the appropriate form for γ follows using the symmetry of G -invariant kernels (Lemma 1 (c)). \square

2 Weyl’s Criterion

Weyl’s criterion concerns uniformly distributed sequences $\{x_l : l \in \mathbb{N}\} \subset M$. These are sequences for which

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{l=1}^N \delta_{x_l}$$

(δ_x is the point evaluation functional at x) converge weakly to the measure μ . In this section we provide alternative characterisations for uniformly distributed sequences. The equivalence of the above definition to that of part (a) of the following theorem follows from standard arguments (see Kuipers and Niederreiter [4]).

In this section, we assume that $a_{n,k}(\kappa) > 0$ for all n, k , and

$$\sum_{n=0}^{\infty} \sum_{k=1}^{\nu_n} d_{n,k} a_{n,k}(\kappa) < \infty. \tag{1}$$

Thus κ is bounded and continuous on $M \times M$. More importantly for our purpose in this section, κ is strictly positive definite on M . We will prove the equivalence of two characterisations of uniform distribution of points on M . Our main result of this section is as follows.

Theorem 2. *The following two criteria of a uniformly distributed sequence on M are equivalent.*

(a) A sequence $\{x_l : l \in \mathbb{N}\}$ is uniformly distributed on M if and only if

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{l=1}^N Y_{n,k}^j(x_l) = 0$$

for all $n \geq 0$ and $1 \leq k \leq \nu_n, 1 \leq j \leq d_{n,k}$.

(b) Let κ be a strictly positive definite G -invariant kernel on M . A sequence $\{x_l : l \in \mathbb{N}\}$ is uniformly distributed on M if and only if

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{l=1}^N \kappa(x_l, y) = a_{0,0}(\kappa),$$

holds true uniformly for $y \in M$.

Proof. Using the series expansion for κ we have for any $y \in M$,

$$\frac{1}{N} \sum_{l=1}^N \kappa(x_l, y) = \sum_{n=0}^{\infty} \sum_{k=1}^{\nu_n} a_{n,k}(\kappa) \sum_{j=1}^{d_{n,k}} Y_{n,k}^j(y) \left(\frac{1}{N} \sum_{l=1}^N Y_{n,k}^j(x_l) \right). \tag{2}$$

Suppose $\{x_l : l \in \mathbb{N}\}$ is uniformly distributed by criterion (a). Using Lemma 1, part (d), we can dominate the right hand side of the last equation by

$$\sum_{n=0}^{\infty} \sum_{k=1}^{\nu_n} a_{n,k}(\kappa) \frac{1}{N} \sum_{l=1}^N |Q_{n,k}(x_l, y)| \leq \sum_{n=0}^{\infty} \sum_{k=1}^{\nu_n} d_{n,k} a_{n,k}(\kappa).$$

The right hand side of the inequality is bounded from equation (1). This allows us to use the dominated convergence theorem to pass the limit in N through the sum to get

$$\begin{aligned} & \lim_{N \rightarrow \infty} \sum_{n=1}^{\infty} \sum_{k=1}^{\nu_n} a_{n,k}(\kappa) \sum_{j=1}^{d_{n,k}} Y_{n,k}^j(y) \left(\frac{1}{N} \sum_{l=1}^N Y_{n,k}^j(x_l) \right) \\ &= \sum_{n=1}^{\infty} \sum_{k=1}^{\nu_n} a_{n,k}(\kappa) \sum_{j=1}^{d_{n,k}} Y_{n,k}^j(y) \lim_{N \rightarrow \infty} \left(\frac{1}{N} \sum_{l=1}^N Y_{n,k}^j(x_l) \right) \\ &= 0, \end{aligned}$$

by assumption. Thus

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{l=1}^m \kappa(x_l, y) = a_{0,0}(\kappa)$$

uniformly for each y by (1), and the sequence $\{x_l : l \in \mathbb{N}\}$ is thus uniformly distributed by criterion (b).

Conversely suppose that $\{x_l : l \in \mathbb{N}\}$ is uniformly distributed by criterion (b). Then, as in equation (2), we have

$$\frac{1}{N^2} \sum_{m=1}^N \sum_{l=1}^N \kappa(x_m, x_l) = \sum_{n=0}^{\infty} \sum_{k=1}^{\nu_n} a_{n,k}(\kappa) \sum_{j=1}^{d_{n,k}} \left(\frac{1}{N} \sum_{l=1}^N Y_{n,k}^j(x_l) \right)^2.$$

Now, for each x_m , by hypothesis

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{l=1}^N \phi(x_m, x_l) = \int_M \phi(x_m, x) d\mu(x) = a_{0,0}(\kappa).$$

Thus,

$$\lim_{N \rightarrow \infty} \frac{1}{N^2} \sum_{m=1}^N \sum_{l=1}^N \phi(x_l, x_j) = \int_M \phi(x, x_j) d\mu(x) = a_{0,0}(\kappa).$$

Therefore

$$\lim_{N \rightarrow \infty} \sum_{n=1}^{\infty} \sum_{k=1}^{\nu_n} a_{n,k}(\kappa) \sum_{j=1}^{d_{n,k}} \left(\frac{1}{N} \sum_{l=1}^N Y_{n,k}^j(x_l) \right)^2 = 0,$$

and since $a_{n,k}(\kappa) > 0$, $n \in \mathbb{N}$ and $1 \leq k \leq \nu_n$, it must be that

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{l=1}^N Y_{n,k}^j(x_l) = 0,$$

so that $\{x_l : l \in \mathbb{N}\}$ is uniformly distributed by (a). \square

We note that criterion (a) is called Weyl’s criterion in the literature.

3 Energy on Manifolds

In this section, we work with kernels κ that satisfy the following two conditions:

1. There exists a positive constant C , independent of x , such that

$$\int_M |\kappa(x, y)| d\mu(y) \leq C.$$

2. For each non-trivial continuous function ϕ on M , we have

$$\int_M \int_M \kappa(x, y) \phi(x) \phi(y) d\mu(x) d\mu(y) > 0.$$

We will call a kernel κ satisfying the above two conditions *admissible*. The archetype for admissible kernels is the *Riesz kernel*

$$\kappa(x, y) = \|x - y\|^{-s}, \quad 0 < s < d + r, \quad x, y \in M,$$

where $\|\cdot\|$ is the Euclidean norm in \mathbb{R}^{d+r} .

We are interested in studying errors of numerical integration of continuous functions $f : M \rightarrow \mathbb{R}$ over a set $Z \subset M$ of cardinality $N \geq 1$. In particular, we seek a generalization of results of Damelin and Grabner in [2]. More precisely, given an admissible kernel κ and such a point set Z , we define the discrete energy

$$E_\kappa(Z) = \frac{1}{N^2} \sum_{\substack{y, z \in Z \\ y \neq z}} \kappa(y, z)$$

and for the normalised G -invariant measure μ on M , denote by

$$R(f, Z, \mu) := \left| \int_M f d\mu - \frac{1}{N} \sum_{y \in Z} f(y) \right|$$

the error of numerical integration of f with respect to μ over M .

For an admissible kernel κ and probability measure ν on M , we define the energy integral

$$\mathcal{E}_\kappa(\nu) = \int_M \int_M \kappa(x, y) d\nu(x) d\nu(y).$$

We have

Lemma 3. *The energy integral $\mathcal{E}_\kappa(\nu)$ is uniquely minimised by the normalized G -invariant measure μ .*

Proof. Since κ satisfies condition 2 we have $a_{n,k} > 0$, $\mathcal{E}_\kappa(\nu) \geq 0$ for every Borel probability measure ν . Also, a simple computation shows that $\mathcal{E}_\kappa(\mu) = a_{0,0}(\kappa)$.

Next, for an arbitrary probability measure σ on M , we use Lemma 1, part (d) to write down

$$\begin{aligned} \mathcal{E}_\kappa(\sigma) &= \int_M \int_M \left\{ \sum_{n=0}^{\infty} \sum_{k=1}^{\nu_n} a_{n,k}(\kappa) Q_{n,k}(x, z) \right\} d\sigma(x) d\sigma(z) \\ &= a_{0,0}(\kappa) + \sum_{n=1}^{\infty} \sum_{k=1}^{\nu_n} a_{n,k}(\kappa) \int_M \int_M Q_{n,k}(x, z) d\sigma(x) d\sigma(z) \\ &= a_{0,0}(\kappa) + \sum_{n=1}^{\infty} \sum_{k=1}^{\nu_n} a_{n,k}(\kappa) \int_M \int_M \int_M Q_{n,k}(x, y) Q_{n,k}(y, z) d\mu(y) d\sigma(x) d\sigma(z) \\ &= a_{0,0}(\kappa) + \sum_{j=1}^{\infty} \sum_{k=1}^{\nu_n} a_{n,k}(\kappa) \int_M \left\{ \int_M Q_{n,k}(x, y) d\sigma(x) \right\}^2 d\mu(y). \end{aligned}$$

If ν is a probability measure on M that minimises $\mathcal{E}_\kappa(\sigma)$, i.e.,

$$\mathcal{E}_\kappa(\nu) = \min_{\sigma} \mathcal{E}_\kappa(\sigma),$$

where the minimum is taken over all the probability measures on M , then ν must satisfy

$$\int_M Q_{n,k}(x, y) d\nu(x) = 0, \quad k = 1, \dots, \nu_n, \quad n \geq 1.$$

Hence, since μ also annihilates all polynomials of degree ≥ 0 , $\nu - \mu$ annihilates all polynomials. Because the polynomials are dense in the continuous functions, we see that $\nu - \mu$ is the zero measure and the result is proved. \square

Heuristically, one expects that a point distribution Z of minimal energy gives a discrete approximation to the measure μ , in the sense that the integral with respect to the measure is approximated by a discrete sum over the points of Z . For the sphere, this was shown by Damelin and Grabner in [2] for Riesz kernels. The essence of our main result below is that we are able to formulate a general analogous result which works on M and for a subclass of admissible kernels κ . To describe this result, we need some more notations.

Let σ_α be a sequence of kernels converging to the δ distribution (the distribution for which all Fourier coefficients are unity) as $\alpha \rightarrow 0$. Let κ be admissible and for $\alpha < \alpha_0$ for some fixed α_0 , we wish the convolution $\kappa_\alpha = \kappa * \sigma_\alpha$ to have the following properties:

- (a) κ_α is positive definite
- (b) $\kappa_\alpha(x, y) \leq \kappa(x, y)$ for all $x, y \in M$.

If the above construction is possible, we say that κ is *strongly admissible*. Besides Riesz kernels on d dimensional spheres see [2, 3], we have as a further natural example on the 2-torus embedded in \mathbb{R}^4 , strongly admissible kernels defined as products of univariate kernels:

$$\kappa(x, y) = \rho(x_1, y_1)\rho(x_2, y_2), \quad x_1, y_1, x_2, y_2 \in S^1,$$

where

$$\rho(s, t) = |1 - st|^{-1/2}, \quad s, t \in S^1$$

and S^1 is the one dimensional circle (realized as a subset of \mathbb{R}^2). See [3] for further details.

We now give an interesting result which demonstrates the way in which results on the sphere can be transplanted onto more general manifolds. The reader is directed to [3] for the proof and further results.

Theorem 3. *Let κ be strongly admissible on M and $Z \subset M$ be a point subset of cardinality $N \geq 1$. Fix $x \in M$. If q is a polynomial of degree at most $n \geq 0$ on M then, for $\alpha < \alpha_0$,*

$$\begin{aligned} & |R(q, Z, \mu)| \\ & \leq \max_{j \leq n, l \leq \nu_n} \frac{1}{(a_{j,l}(\kappa_\alpha))^{1/2}} \|q\|_2 \left(E_\kappa(Z) + \frac{1}{N} \kappa_\alpha(x, x) - a_{0,0}(\kappa_\alpha) \right)^{1/2}. \end{aligned}$$

References

1. W. Freeden, T. Gervens, and M. Schreiner: *Constructive Approximation on the Sphere with Applications to Geomathematics*. Clarendon Press, Oxford, 1998.
2. S.B. Damelin and P. Grabner: Numerical integration, energy and asymptotic equidistribution on the sphere. *Journal of Complexity* **19**, 2003, 231–246.
3. S.B. Damelin, J. Levesley, and X. Sun: Quadrature estimates for compact homogeneous manifolds. Manuscript.
4. L. Kuipers and H. Niederreiter: *Uniform Distribution of Sequences*. Wiley-Interscience, New York, 1974.
5. J. Levesley and D.L. Ragozin: The density of translates of zonal kernels on compact homogeneous spaces. *J. Approx. Theory* **103**, 2000, 252–268.
6. G.D. Mostow: Equivariant embeddings in Euclidean space. *Ann. Math.* **65**, 1957, 432–446.

Minimal Discrete Energy Problems and Numerical Integration on Compact Sets in Euclidean Spaces

Steven B. Damelin¹ and Viktor Maymeskul²

¹ Institute for Mathematics and its Applications, University of Minnesota, Minneapolis, MN 55455, U.S.A., damelin@ima.umn.edu

² Department of Mathematical Sciences, Georgia Southern University, Georgia 30460, U.S.A., vmaymesk@georgiasouthern.edu

Summary. In this paper, we announce and survey recent results on (a) point energies, scar defects, separation and mesh norm for optimal $N \geq 1$ arrangements of points on a class of d -dimensional compact sets embedded in \mathbb{R}^n , $n \geq 1$, which interact through a Riesz potential, and (b) discrepancy estimates of numerical integration on the d -dimensional unit sphere S^d , $d \geq 2$.

1 Introduction

1.1 Discrete Riesz Energy Problems

The problem of uniformly distributing points on spheres (more generally, on compact sets in \mathbb{R}^n) is an interesting and difficult problem. It is folklore, that such problems were discussed already by Carl Friedrich Gauss in his famous *Disquisitiones arithmeticae*, although it is most likely that similar problems appeared in mathematical writings even before that time.

For $d \geq 1$, let S^d denote the d -dimensional unit sphere in \mathbb{R}^{d+1} , given by

$$x_1^2 + \cdots + x_{d+1}^2 = 1. \quad (1)$$

For $d = 1$, the problem is reduced to uniformly distributing N points on a circle, and equidistant points provide an obvious answer. For $d \geq 2$, the problem becomes much more difficult; in fact, there are numerous criteria for uniformity, resulting in different optimal configurations on the sphere. Many constructions of “well-distributed” point sets have been given in the literature. These include constructions of generalized spiral points, low-discrepancy point sets in the unit cube, which can be transformed via standard parameterizations, constructions given by integer solutions of the equation $x_1^2 + \cdots + x_{d+1}^2 = N$ projected onto the sphere, rotations of certain subgroups applied to points on the sphere, finite field constructions of point sets based on finite field solutions of (1), and associated combinatorial designs. See [2, 6, 8, 9, 7, 10, 11, 12] and the references cited therein.

In this paper, we are interested in studying certain arrangements of N points on a class of d -dimensional compact sets A embedded in \mathbb{R}^n . We assume that these points interact through a power law (Riesz) potential $V = r^{-s}$, where $s > 0$ and r is the Euclidean distance in \mathbb{R}^n .

For a compact set $A \subset \mathbb{R}^n$, $s > 0$, and a set $\omega_N = \{x_1, \dots, x_N\}$ of distinct points on A , the discrete Riesz s -energy associated with ω_N is given by

$$E_s(A, \omega_N) := \sum_{1 \leq i < j \leq N} |x_i - x_j|^{-s}. \tag{2}$$

Let $\omega_N^* := \{x_1^*, \dots, x_N^*\} \subset A$ be a configuration, for which $E_s(A, \omega_N)$ attains its minimal value; that is,

$$\mathcal{E}_s(A, N) := \min_{\omega_N \subset A} E_s(A, \omega_N) = E_s(A, \omega_N^*). \tag{3}$$

We shall call such minimizing configurations *s-extremal configurations*. It is well-known that, in general, s -extremal configurations are not always unique. For example, in the case of S^d , they are invariant under rotations. A natural physical interpretation of minimal energy problem on the sphere is the electron problem, which asks for distributions of electrons in stable equilibrium.

Natural questions that arise in studying the discrete Riesz energy are:

- (1) What is the asymptotic behavior of $\mathcal{E}_s(A, N)$, as $N \rightarrow \infty$?
- (2) How are s -extremal configurations distributed on A for large N ?

It is well-known that answers to these questions essentially depend on the relation between s and the Hausdorff dimension $d_H(A)$ of A . We demonstrate this fact with the following two classical examples. Throughout the paper, we denote by C, C_1, \dots positive constants, and by c, c_1, \dots sufficiently small positive constants (different each time, in general), that may depend on d, s, A but independent of N . We refer the reader to [8, 9] and the references cited therein for more details.

Example 1. The interval $[-1, 1]$, $d_H([-1, 1]) = 1$: It is known that $s = 1$ is the critical value in the sense that s -extremal configurations are distributed on $[-1, 1]$ differently for $s < 1$ and $s \geq 1$. Indeed, for $0 < s < 1$, the limiting distribution of s -extremal configurations has an arcsine-type density and, for $s \geq 1$, the limiting distribution is the uniform distribution on $[-1, 1]$. Concerning the minimal energies, they again behave differently for $s < 1$, $s = 1$, and $s > 1$. With $e_s := [\sqrt{\pi}\Gamma(1 + s/2)] / [\cos(\pi s/2)\Gamma((1 + s)/2)]$,

$$\mathcal{E}_s([-1, 1], N) \sim \begin{cases} (1/2)N^2 e_s, & s < 1, \\ (1/2)N^2 \ln N, & s = 1, \\ (1/2)^s \zeta(s) e(s) N^{1+s}, & s > 1, \end{cases}$$

where $\zeta(s)$ stands for the Riemann zeta function.

Example 2. The unit sphere S^d , $d_H(S^d) = d$: Here again, there are three cases to consider: $s < d$, $s = d$, and $s > d$. In all cases, see [6], the limiting distribution of s -extremal configurations is given by the normalized area measure σ_d on S^d , which is natural due to rotation invariance, but the asymptotic behavior of $\mathcal{E}_s(S^d, N)$ is quite different. With $\tau_{s,d}(N)$ denoting N^2 if $s < d$, $N^2 \ln N$ if $s = d$, and $N^{1+s/d}$ if $s > d$, the limit $\lim_{N \rightarrow \infty} \mathcal{E}_s(S^d, N) / \tau_{s,d}(N)$ exists and is known in the first two cases (see [6, 10]).

The dependence of the distribution of s -extremal configurations over A and the asymptotics for minimal discrete s -energy on s can be explained using potential theory. Indeed, for a probability Borel measure ν on A , its s -energy integral is defined to be

$$I_s(A, \nu) := \int_{A \times A} |x - y|^{-s} d\nu(x)d\nu(y), \tag{4}$$

which can be finite or infinite. For a set $\omega_N = \{x_1, \dots, x_N\} \subset A$, let

$$\nu^{\omega_N} := \frac{1}{N} \sum_{j=1}^N \delta_{x_j} \tag{5}$$

denote the normalized counting measure of ω_N (so that $\nu^{\omega_N}(A) = 1$). Then the discrete Riesz s -energy (2), associated with ω_N , can be written as

$$E_s(A, \omega_N) = \frac{N^2}{2} \int_{x \neq y} |x - y|^{-s} d\nu^{\omega_N}(x)d\nu^{\omega_N}(y). \tag{6}$$

where the integral represents a discrete analog of the s -energy integral (4).

If $s < d_H(A)$, then it is well-known that the energy integral (4) is minimized uniquely by the *equilibrium measure* ν_s^A . On the other hand, the normalized counting measure ν^{ω_N} of an s -extremal configuration minimizes the discrete energy integral in (6) over all sets ω_N on A . Thus, one can reasonably expect that, for N large, ν^{ω_N} is “close” to ν_s^A and, therefore, the minimal discrete s -energy $\mathcal{E}_s(A, N)$ is close to $(1/2)N^2 I_s(A, \nu_s^A)$.

If $s \geq d_H(A)$, then the energy integral (4) diverges for every measure ν . Thus, $\mathcal{E}_s(A, N)$ must grow faster than N^2 . Concerning the distribution of s -extremal points over A , the interactions are strong enough to force points to stay away from each other as far as possible since the closest neighbors are now dominating. So, s -extremal points distribute themselves over A in an equally spaced manner.

In Section 2, we describe some recent results of the authors obtained in [8, 9] concerning separation, mesh norm, and point energies of s -extremal Riesz configurations on a wide class of compact sets in \mathbb{R}^n , and refer the reader to some latest results of other authors in this area. In particular, we give new separation estimates for the Riesz points on the unit sphere S^d for the case $0 < s < d - 1$ and confirm *scar defects* conjecture ([3, 8, 9]) based on numerical experiments.

1.2 Numerical Integration and g -Functionals

Numerical integration and discrepancy estimates are important problems in applied mathematics and many applications, when one needs to approximate $\int_{\mathcal{B}} f d\zeta$, where $\mathcal{B} \subset \mathbb{R}^n$, $n \geq 3$, is a bounded domain or manifold, $d\zeta : \mathbb{R}^n \rightarrow \mathbb{R}$ is a Borel measure with compact support in \mathcal{B} , and f belongs to a suitable class of real valued functions on \mathcal{B} , by a finite sum using values of f at a discrete set of nodes ω_N . Such problems arise naturally in many areas of growing interest such as mathematical finance, physical geodesy, meteorology, and diverse mathematical areas such as approximation theory, spherical t -designs, discrepancy, combinatorics, Monte-Carlo and Quasi-Monte-Carlo methods, finite fields, information based complexity theory, and statistical learning theory.

In this paper, we consider the case when $\mathcal{B} = S^d$ and the measure $d\zeta$ is the normalized area measure σ_d .

For a set of nodes $\omega_N = \{x_{1,N}, \dots, x_{N,N}\} \subset S^d$, a natural measure for the quality of its distribution on the sphere is the spherical cap discrepancy

$$D(\omega_N) = \sup_{C \subseteq S^d} \left| \sum_{k=1}^N [\nu^{\omega_N} - \sigma_d](C) \right|,$$

where the supremum ranges over all spherical caps $C \subseteq S^d$ and ν^{ω_N} is the normalized counting measure (5) of ω_N . The discrepancy simply measures the maximal deviation between ν^{ω_N} and the normalized area measure σ_d over all spherical caps or, in other words, the worst error in numerical integration of indicator functions of spherical caps using the set of nodes ω_N .

For a continuous function $f : S^d \rightarrow \mathbb{R}$, we denote by

$$R(f, \omega_N) := \int_{S^d} f(x) d\sigma_d(x) - \frac{1}{N} \sum_{k=1}^N f(x_k) = \int_{S^d} f(x) d[\sigma_d - \nu^{\omega_N}]$$

the error in numerical integration on the sphere S^d using nodes in ω_N .

Clearly, to have $R(f, \omega_N) \rightarrow 0$, as $N \rightarrow \infty$, for any continuous function f on S^d , the points in ω_N should be distributed over S^d nicely in the sense that $D(\omega_N) \rightarrow 0$, as $N \rightarrow \infty$.

In Section 3, we briefly discuss spherical cap discrepancy and error estimates for numerical integration on S^d , and refer the interested reader to [6, 7] and the references cited therein for a comprehensive account of this vast and interesting subject. The methods used in [6, 7] are motivated by the discussion on s -energy and s -extremal Riesz points presented in Section 2. A crucial observation was the possibility of use of g -functionals, generalizing classical Riesz and logarithmic functionals, to estimate the second order terms in the expansions of g -energies, which yield errors in numerical integration valid for a large class of smooth functions on the sphere.

2 Point Energies, Separation, and Mesh Norm for Optimal Riesz Points on d -Rectifiable Sets

In this section, we focus on the results obtained by the authors in [8, 9], which are dealing with properties of s -extremal Riesz configurations on compact sets in \mathbb{R}^n , and refer an interested reader to the references and [6, 8, 7, 10] for results of other authors.

2.1 The Case $s > d$

We define a class \mathcal{A}^d of d -dimensional compact sets $A \subset \mathbb{R}^n$ for which, in the case $s \geq d$, the asymptotic behavior of $\mathcal{E}_s(A, N)$, separation and mesh norm estimates, and the limiting distribution of ω_N^* (in terms of weak-star convergence of normalized counting measures) over A have been recently obtained.

Definition 1. We say that a set A belongs to the class \mathcal{A}^d if, for some $n \geq d$, $A \subset \mathbb{R}^n$ and

- (1) $H^d(A) > 0$ and
- (2) A is a finite union of bi-Lipschitz images of compact sets in \mathbb{R}^d , that is

$$A = \bigcup_{i=1}^m \phi_i(K_i),$$

where each $K_i \subset \mathbb{R}^d$ is compact and $\phi_i : K_i \rightarrow \mathbb{R}^n$ is bi-Lipschitz on K_i , $i = 1, \dots, m$.

Here and throughout the paper, $H^d(\cdot)$ denotes the d -dimensional Hausdorff measure in \mathbb{R}^n .

For a collection $\omega_N = \{x_1, \dots, x_N\}$ of distinct points on a set $A \subset \mathbb{R}^n$, let

$$\delta(A, \omega_N) := \min_{i \neq j} |x_i - x_j|, \quad \rho(A, \omega_N) := \max_{x \in A} \min_{1 \leq j \leq N} |x - x_j|.$$

The quantity $\delta(A, \omega_N)$ is called the *separation radius* and gives the minimal distance between points in ω_N , while the *mesh norm* $\rho(A, \omega_N)$ means the maximal radius of a “cap” $E(x, r)$ (see (7)) on A , which does not contain points from ω_N . We also define the point energies of the points in ω_N by

$$E_{j,s}(A, \omega_N) := \sum_{i \neq j} |x_j - x_i|^{-s}, \quad j = 1, \dots, N.$$

The following two results were established in [8].

Theorem 1. Let $A \in \mathcal{A}^d$ and $s > d$. Then, for all $1 \leq j \leq N$,

$$E_{j,s}(A, \omega_N^*) \leq CN^{s/d}.$$

Corollary 1. For $A \in \mathcal{A}^d$, $s > d$, and any s -extremal configuration ω_N^* on A ,

$$\delta(A, \omega_N^*) \geq cN^{-1/d}.$$

We note that this is the best possible lower estimate on the separation radius. Under some additional restrictions on a set $A \in \mathcal{A}^d$, this estimate was obtained earlier in [10]. Concerning the mesh norm $\rho(A, \omega_n^*)$ of s -extremal configurations, the following result was proved in [9].

Theorem 2. Let $A \in \mathcal{A}^d$, $s > d$, and let ω_N^* be an s -extremal configuration on A . Then

$$\rho(A, \omega_N^*) \leq CN^{-1/d}.$$

Regarding point energies for s -extremal Riesz configurations, we define a subset $\tilde{\mathcal{A}}^d$ of \mathcal{A}^d (see [9]), for which we have obtained a lower estimate matching the upper one in Theorem 1.

Let, for $x \in A$ and $r > 0$,

$$E(x, r) := \{y \in A : |y - x| < r\}. \tag{7}$$

Definition 2. We say that a set $A \in \tilde{\mathcal{A}}^d$ if

- (1) $A \in \mathcal{A}^d$ and
- (2) there is a constant $c > 0$ such that, for any $x \in A$ and $r > 0$ small enough,

$$\text{diam}(E(x, r)) \geq cr. \tag{8}$$

Along with trivial examples, such as a set consisting of a finite number connected components (not singletons), the diameter condition holds for many sets with infinitely many connected components. Say, Cantor sets (known to be totally disconnected) with positive Hausdorff measure are in the class $\tilde{\mathcal{A}}^d$.

Theorem 3. Let $A \in \tilde{\mathcal{A}}^d$ and $s > d$. Then

$$c \leq N^{1/d} \delta(A, \omega_N^*) \leq C \tag{9}$$

and, therefore, for any $1 \leq j \leq N$,

$$E_{j,s}(A, \omega_N^*) \geq cN^{s/d}. \tag{10}$$

Combining Theorems 1 and 3 yields

Corollary 2. For $s > d$ and any s -extremal configuration ω_N^* on $A \in \tilde{\mathcal{A}}^d$,

$$c \leq \frac{\max_{1 \leq j \leq N} E_{j,s}(A, \omega_N^*)}{\min_{1 \leq j \leq N} E_{j,s}(A, \omega_N^*)} \leq C. \tag{11}$$

Thus, for $A \in \tilde{\mathcal{A}}^d$ and $s > d$, all point energies in an s -extremal configuration are asymptotically of the same order, as $N \rightarrow \infty$.

We note that estimates given in Theorems 2, 3, and Corollary 2 were obtained in [8], but with the diameter condition (8) replaced by the more restrictive measure condition $H^d(E(x, r)) \geq cr^d$.

Most likely, (11) is the best possible assertion in the sense that the point energies are not, in general, asymptotically equal, as $N \rightarrow \infty$. (Compare with the case of the unit sphere S^d and $0 < s < d - 1$ in Theorem 4(c) below.)

Simple examples show that the estimates (9), (10), and (11) are not valid, in general, for a set $A \in \mathcal{A}^d$ without an additional condition on its geometry. Indeed, as a counterexample, for $x \in \mathbb{R}^{d+1}$ with $|x| > 1$, let $A = S^d \cup \{x\}$.

2.2 The Case $0 < s < d - 1$ for S^d

In doing quadrature, it is important to know some specific properties of low discrepancy configurations, such as the separation radius, mesh ratio, and point energies. In [8], the authors established lower estimates on the separation radius for s -extremal Riesz configurations on S^d for $0 < s < d - 1$ and proved the asymptotic equivalence of the point energies, as $N \rightarrow \infty$.

Theorem 4. Let ω_N^* be an s -extremal configuration on S^d . Then

- (a) for $d \geq 2$ and $s < d - 1$, $\delta(S^d, \omega_N^*) \geq cN^{-1/(s+1)}$;
- (b) for $d \geq 3$ and $s \leq d - 2$, $\delta(S^d, \omega_N^*) \geq cN^{-1/(s+2)}$, which is sharp in s for $s = d - 2$;
- (c) for any $0 < s < d - 1$,

$$\lim_{N \rightarrow \infty} \frac{\max_{1 \leq j \leq N} E_{j,s}(S^d, \omega_N^*)}{\min_{1 \leq j \leq N} E_{j,s}(S^d, \omega_N^*)} = 1.$$

We remark that numerical computations for a sphere (see [3]) show that, for any $s > 0$, the point energies are nearly equal for almost all points that are of so-called “hexagonal” type. However, some (“pentagonal”) points have elevated energies and some (“heptagonal”) points have low energies. The transition from points that are “hexagonal” to those that are “pentagonal” or “heptagonal” induce scar defects, which are conjectured to vanish, as $N \rightarrow \infty$. Theorem 4(c) provides strong evidence for this conjecture for $0 < s < d - 1$. We refer the reader to a recent paper [11], where sharp separation results for s -extremal configurations are obtained in the case $d - 1 < s < d$. The separation radius for the case $s = d - 1$ was studied by Dahlberg in [4] and the cases $d - 1 < s < d$ by Kuijlaars et al. in [11].

3 Discrepancy and Errors of Numerical Integration on Spheres

The following discrepancy and numerical integration results were established in [6]. See also [7].

Definition 3. Let, for $\delta_0 > 0$, $g(t) : [-1 - \delta_0, 1] \rightarrow \mathbb{R}$ be a continuous function. We say that $g(t)$ is “admissible” if it satisfies the following conditions:

- (a) $g(t)$ is strictly increasing with $\lim_{t \rightarrow 1-} g(t) = \infty$.
- (b) If $g(t - \delta)$ is given by its ultraspherical expansion $\sum_{n=0}^{\infty} a_n(\delta) P_n^{(d)}(t)$, valid for $t \in [-1, 1]$, then we assume that, for all $n \geq 1$ and $0 < \delta \leq \delta_0$, $a_n(\delta) > 0$.
- (c) The integral

$$\int_{-1}^1 g(t)(1 - t^2)^{(d/2)-1} dt$$

converges.

Here $P_n^{(d)}$ are the ultraspherical polynomials corresponding to the d -dimensional sphere normalized by $P_n^{(d)}(1) = 1$.

One immediately checks that the following choices of admissible functions $g(t)$ yield the classical energy functionals: $g_L^0(t) := -2^{-1} \log[2(1 - t)]$ for the logarithmic energy and $g_R^s(t) := 2^{-s/2}(1 - t)^{-s/2}$, $s > 0$, for the Riesz s -energy.

For a set $\omega_N = \{x_1, \dots, x_N\} \subset S^d$, similarly to (2) and (3), we define

$$E_g(S^d, \omega_N) := \sum_{1 \leq i < j \leq N} g(\langle x_i, x_j \rangle),$$

where $\langle \cdot \rangle$ denotes inner product in \mathbb{R}^{d+1} , and

$$\mathcal{E}_g(S^d, N) := \min_{\omega_N \subset S^d} E_g(S^d, \omega_N).$$

A point set ω_N^* , for which the minimal energy $\mathcal{E}_g(S^d, N)$ is attained, is called a *minimal g-energy* point set. It was shown in [6] that, for any admissible function $g(t)$, the energy integral

$$I_g(S^d, \nu) := \int_{S^d \times S^d} g(\langle x, y \rangle) d\nu(x) d\nu(y)$$

is minimized by the normalized area measure σ_d amongst all Borel probability measures ν on S^d . Using arguments similar to those in examples 1 and 2, one expects that the normalized counting measure $\nu^{\omega_N^*}$ of ω_N^* gives a discrete approximation to the normalized area measure σ_d in the sense that the integral of any continuous function f on S^d against σ_d is approximated by the (N^{-1}) -weighted discrete sum of values of f at the points in ω_N^* .

Theorem 5. *Let $g(t)$ be admissible, $d \geq 2$, ω_N be a collection of N points on S^d , f be a polynomial of degree at most $n \geq 1$ on \mathbb{R}^{d+1} , and $0 < \delta \leq \delta_0$. Then*

$$(a) |R(f, \omega_N)| \leq \|f\|_2 \left(\frac{2N^{-2}E_g(S^d, \omega_N) - a_0(\delta) + N^{-1}g(1 - \delta)}{\min_{1 \leq k \leq n} [a_k(\delta)/Z(d, k)]} \right)^{1/2}$$

with $Z(d, k)$ counting the linearly independent spherical harmonics of degree k on S^d . Moreover, if $q = q(d)$ is the smallest integer satisfying $2q \geq d + 3$, then there exists a positive constant C , independent of N and ω_N , such that uniformly on $m \geq 1$ and $0 < \delta < \delta_0$ there holds

$$D_N(\omega_N) \leq C \left\{ \frac{1}{m} + \left(\frac{2N^{-2}E_g(S^d, \omega_N) - a_0(\delta) + N^{-1}g(1 - \delta)}{\min_{1 \leq k \leq n} [a_k(\delta)/Z(d, k)]} \right)^{1/2} \right\}.$$

(b) *Let f be a continuous function on S^d satisfying*

$$|f(x) - f(y)| \leq C_f \arccos(\langle x, y \rangle), \quad x, y \in S^d. \tag{12}$$

Then, for any $n \geq 1$,

$$|R(f, \omega_N)| \leq 12C_f \frac{d}{n} + \left(\frac{2N^{-2}E_g(S^d, \omega_N) - a_0(\delta) + N^{-1}g(1 - \delta)}{\min_{1 \leq k \leq n} [a_k(\delta)/Z(d, k)]} \right)^{1/2}.$$

Remark 1. Theorem 5 shows that second order terms in the expansion of minimal energies determine rates in errors of numerical integration over spheres. Indeed, one hopes that the energy term $2N^{-2}E_g(S^d, \omega_N)$ and the leading term $a_0(\delta)$ cancel each other sufficiently to allow for an exact error. An application of this idea was exploited first in [6] in the case $s = d$. (See Theorem 6 below.) See also [1].

We now quantify the error in Theorem 5 for d -extremal configurations on S^d (which are sets of minimal g_R^d -energy).

Theorem 6. *Let f be a continuous function on S^d satisfying (12), and let ω_N^* be a d -extremal configuration. Then*

$$|R(f, \omega_N^*)| = \mathcal{O}\left(\frac{C_f + \|f\|_\infty \sqrt{\log \log N}}{\sqrt{\log N}}\right)$$

with the implied constant depending only on d . Moreover,

$$D(\omega_N^*) = \mathcal{O}\left(\sqrt{\log \log N / \log N}\right).$$

We remark that it is widely believed that the order above may indeed be improvable to a negative power of N . Thus far, however, it is not clear how to prove whether this belief is indeed correct.

Acknowledgement

The first author is supported, in part, by EP/C000285 and NSF-DMS-0439734.

References

1. J. Brauchart: Invariance principles for energy functionals on spheres. *Monatsh. Math.* **141**(2), 2004, 101–117.
2. B. Bajnok, S.B. Damelin, J. Li, and G. Mullen: A constructive finite field method for scattering points on the surface of a d -dimensional sphere. *Computing* **68**, 2002, 97–109.
3. M. Bowick, A. Cacciuto, D.R. Nelson, and A. Travesset: Crystalline order on a sphere and the generalized Thomson problem. *Phys. Rev. Lett.* **89**, 2002, 185–502.
4. B.E.J. Dahlberg: On the distribution of Fekete points. *Duke Math.* **45**, 1978, 537–542.
5. S.B. Damelin: A discrepancy theorem for harmonic functions on the d dimensional sphere with applications to point cloud scatterings. Submitted.
6. S.B. Damelin and P. Grabner: Energy functionals, numerical integration and asymptotic equidistribution on the sphere. *J. Complexity* **19**, 2003, 231–246; Corrigendum, *J. Complexity*, to appear.
7. S.B. Damelin, J. Levesley, and X. Sun: Energy estimates and the Weyl criterion on compact homogeneous manifolds. This volume.
8. S.B. Damelin and V. Maymeskul: On point energies, separation radius and mesh norm for s -extremal configurations on compact sets in \mathbb{R}^n . *Journal of Complexity* **21**(6), 845–863.
9. S.B. Damelin and V. Maymeskul: On point energies, separation radius and mesh norm for s -extremal configurations on compact sets in \mathbb{R}^n (II). Submitted.
10. D. Hardin and E.B. Saff: Discretizing manifolds via minimal energy points. *Notices of Amer. Math. Soc.* **51**(10), 2004, 1186–1194.
11. A.B.J. Kuijlaars, E.B. Saff, and X. Sun: On separation of minimal Riesz energy points on spheres in Euclidean spaces. Submitted.
12. A. Lubotzky, R. Phillips, and P. Sarnak: Hecke operators and distributing points on the sphere I-II. *Comm. Pure App. Math.* **39-40**, 1986/1987, 148–186, 401–420.

Numerical Quadrature of Highly Oscillatory Integrals Using Derivatives

Sheehan Olver

Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge CB3 0WA, UK, S.Olver@damtp.cam.ac.uk

Summary. Numerical approximation of highly oscillatory functions is an area of research that has received considerable attention in recent years. Using asymptotic expansions as a point of departure, we derive Filon-type and Levin-type methods. These methods have the wonderful property that they improve with accuracy as the frequency of oscillations increases. A generalization of Levin-type methods to integrals over higher dimensional domains will also be presented.

1 Introduction

A highly oscillatory integral is defined as

$$I[f] = \int_{\Omega} f e^{i\omega g} dV,$$

where f and g are smooth functions, $\omega \gg 1$ and Ω is some domain in \mathbb{R}^d . The parameter ω is a positive real number that represents the frequency of oscillations: large ω implies that the number of oscillations of $e^{i\omega g}$ in Ω is large. Furthermore, we will assume that g has no critical points; i.e., $\nabla g \neq 0$ in the closure of Ω . The goal of this paper is to numerically approximate such integrals, with attention paid to asymptotics, as $\omega \rightarrow \infty$.

For large values of ω , traditional quadrature techniques fail to approximate $I[f]$ efficiently. Each sample point for Gauss-Legendre quadrature is effectively a random value on the range of oscillation, unless the number of sample points is sufficiently greater than the number of oscillations. For the multivariate case, the number of sample points needed to effectively use repeated univariate quadrature grows exponentially with each dimension. In the univariate case with no stationary points, the integral $I[f]$ is $\mathcal{O}(\omega^{-1})$ for increasing ω [7]. This compares with an error of order $\mathcal{O}(1)$ when using Gauss-Legendre quadrature [1]. In other words, it is more accurate to approximate $I[f]$ by zero than to use Gauss-Legendre quadrature when ω is large! In this paper, we will demonstrate several methods for approximating $I[f]$ such that the accuracy improves as the frequency ω increases.

2 Univariate Asymptotic Expansion and Filon-type Methods

This section consists of an overview of the relevant material from [1]. We focus on the case where $g' \neq 0$ in $[a, b]$, in other words there are no stationary points. The idea behind recent research into highly oscillatory integrals is to derive an asymptotic expansion for $I[f]$, which we then use to find the order of error of other, more efficient, methods. The key observation is that

$$\begin{aligned} I[f] &= \int_a^b f e^{i\omega g} dx = \frac{1}{i\omega} \int_a^b \frac{f}{g'} \frac{d}{dx} [e^{i\omega g}] dx \\ &= \frac{1}{i\omega} \left[\frac{f}{g'} e^{i\omega g} \right]_a^b - \frac{1}{i\omega} \int_a^b \frac{d}{dx} \left[\frac{f}{g'} \right] e^{i\omega g} dx = Q[f] - \frac{1}{i\omega} I \left[\left(\frac{f}{g'} \right)' \right], \end{aligned}$$

where $Q[f] = \frac{1}{i\omega} \left[\frac{f}{g'} e^{i\omega g} \right]_a^b$. Note that the integral in the error term is $\mathcal{O}(\omega^{-1})$ [7], hence $Q[f]$ approximates $I[f]$ with an error of order $\mathcal{O}(\omega^{-2})$. Moreover, the error term is another highly oscillatory integral, hence we can use $Q[f]$ to approximate it as well. Clearly, by continuing this process, we derive the following asymptotic expansion:

$$I[f] \sim \sum_{k=1}^{\infty} \frac{1}{(i\omega)^k} \left(\sigma_k[f](b) e^{i\omega g(b)} - \sigma_k[f](a) e^{i\omega g(a)} \right),$$

where

$$\sigma_1[f] = \frac{f}{g'}, \quad \sigma_{k+1}[f] = \frac{\sigma_k[f]'}{g'}, \quad k \geq 1.$$

Note that, if f and its first $s - 1$ derivatives are zero at the endpoints, then the first s terms of this expansion are zero and $I[f] \sim \mathcal{O}(\omega^{-s-1})$.

We could, of course, use the partial sums of the asymptotic expansion to approximate $I[f]$. This approximation would improve with accuracy, the larger the frequency of oscillations ω . Unfortunately, the expansion will not typically converge for fixed ω , and there is a limit to how accurate the approximation can be. Hence we derive a Filon-type method. The idea is to approximate f by v using Hermite interpolation, i.e., v is a polynomial such that

$$v(x_k) = f(x_k), v'(x_k) = f'(x_k), \dots, v^{(m_k-1)}(x_k) = f^{(m_k-1)}(x_k),$$

for some set of nodes $\{x_0, \dots, x_\nu\}$ and multiplicities $\{m_0, \dots, m_\nu\}$, and $k = 0, 1, \dots, \nu$. If the moments of $e^{i\omega g}$ are available, then we can calculate $I[v]$ explicitly. Thus define $Q^F[f] = I[v]$. This method has an error

$$I[f] - Q^F[f] = I[f] - I[v] = I[f - v] = \mathcal{O}(\omega^{-s-1}),$$

where $s = \min\{m_0, m_\nu\}$. This follows since f and the first $s - 1$ derivatives are zero at the endpoints, thus the first s terms of the asymptotic expansion are zero. Because the accuracy of $Q^F[f]$ depends on the accuracy of v interpolating f , adding additional sample points and multiplicities will typically decrease the error.

3 Univariate Levin-type Method

Another method for approximating highly oscillatory integrals was developed by Levin in [3]. This method uses collocation instead of interpolation, removing the requirement that moments are computable. If there exists a function F such that $\frac{d}{dx}[Fe^{i\omega g}] = fe^{i\omega g}$, then

$$I[f] = \int_a^b fe^{i\omega g} dx = \int_a^b \frac{d}{dx}[Fe^{i\omega g}]dx = [Fe^{i\omega g}]_a^b.$$

We can rewrite the condition as $\mathcal{L}[F] = f$ for the operator $\mathcal{L}[F] = F' + i\omega g'F$. Hence we approximate F by some function v using collocation, i.e., if $v = \sum c_k \psi_k$ is a linear combination of basis functions $\{\psi_k\}$, then we solve for $\{c_k\}$ using the system $\mathcal{L}[v](x_j) = f(x_j)$, at some set of points $\{x_0, \dots, x_\nu\}$. We can then define the approximation to be

$$Q^L[f] = \int_a^b \mathcal{L}[v]e^{i\omega g} dx = \int_a^b \frac{d}{dx}[ve^{i\omega g}]dx = [ve^{i\omega g}]_a^b.$$

In [4], the current author generalized this method to include multiplicities, i.e., to each sample point x_j associate a multiplicity m_j . This results in the system

$$\mathcal{L}[v](x_j) = f(x_j), \mathcal{L}[v]'(x_j) = f'(x_j), \dots, \mathcal{L}[v]^{(m_j-1)}(x_j) = f^{(m_j-1)}(x_j), \quad (1)$$

for $j = 0, 1, \dots, \nu$. If every multiplicity m_j is one, then this is equivalent to the original Levin method. As in a Filon-type method, if the multiplicities at the end-point are greater than or equal to s , then $I[f] - Q^L[f] = \mathcal{O}(\omega^{-s-1})$, subject to the regularity condition. This condition states that the basis $\{g'\psi_k\}$ can interpolate at the given nodes and multiplicities.

To prove that $Q^L[f]$ has an asymptotic order of $\mathcal{O}(\omega^{-s-1})$, we look at the error term $I[f] - Q^L[f] = I[f - \mathcal{L}[v]]$. If we can show that $\mathcal{L}[v]$ and its derivatives are bounded for increasing ω , the order of error will follow from the asymptotic expansion. Let A be the matrix associated with the system (1), in other words $A\mathbf{c} = \mathbf{f}$, where $\mathbf{c} = [c_0, \dots, c_n]^T$, and \mathbf{f} is the vector associated with the right-hand side of (1). We can write $A = P + i\omega G$, where P and G are independent of ω , and G is the matrix associated with interpolating at the given nodes and multiplicities by the basis $\{g'\psi_k\}$. Thence $\det A = (i\omega)^{n+1} \det G + \mathcal{O}(\omega^n)$. The regularity condition ensures that $\det G \neq 0$, thus $\det A \neq 0$ and $(\det A)^{-1} = \mathcal{O}(\omega^{-n-1})$. Cramer's rule states that $c_k = \frac{\det D_k}{\det A}$, where D_k is the matrix A with the $(k+1)$ th column replaced by \mathbf{f} . Since D_k has one row independent of ω , $\det D_k = \mathcal{O}(\omega^{-n})$, and it follows that $c_k = \mathcal{O}(\omega^{-1})$. Thus $\mathcal{L}[v] = \mathcal{O}(1)$, for $\omega \rightarrow \infty$.

Unlike a Filon-type method, we do not need to compute moments in order to compute $Q^L[f]$. Furthermore, if g has no stationary points and the basis $\{\psi_k\}$ is a Chebyshev set [6]—such as the standard polynomial basis $\psi_k(x) = x^k$ —then the regularity condition is always satisfied. This follows since, if $\{\psi_k\}$ is a Chebyshev set, then $\{g'\psi_k\}$ is also a Chebyshev set.

The following example will demonstrate the effectiveness of this method. Consider the integral $\int_0^1 \cosh x e^{i\omega(x^2+x)} dx$, in other words, $f(x) = \cosh x$ and $g(x) = x^2 + x$. We have no stationary points and moments are computable, hence all the methods discussed so far are applicable. We compare the asymptotic method with

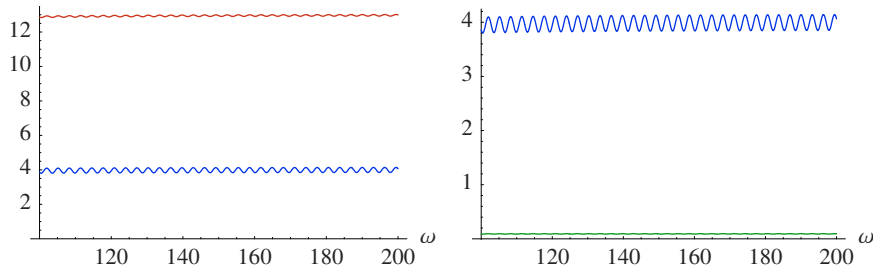


Fig. 1. The error scaled by ω^3 of the asymptotic expansion (left figure, top), $Q^L[f]$ (left figure, bottom)/(right figure, top) and $Q^F[f]$ (right figure, bottom) both with only endpoints and multiplicities two, for $I[f] = \int_0^1 \cosh x e^{i\omega(x^2+x)} dx$.

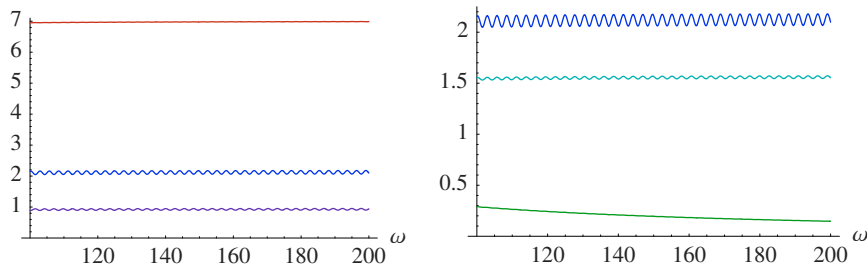


Fig. 2. The error scaled by ω^3 of the asymptotic expansion (left figure, top), $Q^L[f]$ collocating at the endpoints with multiplicities two (left figure, middle)/(right figure, top), $Q^L[f]$ collocating at the endpoints with multiplicities two and midpoint with multiplicity one (left figure, bottom), $Q^L[f]$ with asymptotic basis collocating at endpoints with multiplicities one (right figure, middle) and $Q^L[f]$ with asymptotic basis collocating at endpoints and midpoint with multiplicity one (right figure, bottom), for $I[f] = \int_0^1 \log(x+1) e^{i\omega e^x \sin x} dx$.

a Filon-type method and a Levin-type method, each with nodes $\{0, 1\}$ and multiplicities both two. For this choice of f and g , the Levin-type method is a significant improvement over the asymptotic expansion, whilst the Filon-type method is even more accurate. Not pictured is what happens when additional nodes and multiplicities are added. Adding additional nodes at $\frac{1}{4}$, $\frac{1}{2}$ and $\frac{3}{4}$ with multiplicities all one causes the error of the Levin-type method to drop to roughly equivalent to the current Filon-type method, whilst the error of the Filon-type method decreases even more, to approximately $10^{-5}\omega^{-3}$.

As an example of an integral for which a Filon-type method will not work, consider the case where $f(x) = \log(x+1)$ with oscillator $g(x) = e^x \sin x$. This oscillator is sufficiently complicated so that the moments are unknown. On the other hand, a Levin-type method works wonderfully, as seen in Figure 2. This figure compares the

errors of the asymptotic expansion with a levin-type method collocating at only the endpoints and a levin-type method collocating at the endpoints and the midpoint, where all multiplicities are one.

Unlike a Filon-type method, there is no reason we need to use polynomials for our collocation basis. By choosing our basis wisely we can significantly decrease the error, and, surprisingly, increase the asymptotic order. We define the asymptotic basis, named after its similarity to the terms in the asymptotic expansion, as:

$$\psi_0 = 1, \quad \psi_1 = \frac{f}{g'}, \quad \psi_{k+1} = \frac{\psi'_k}{g'}, \quad k = 1, 2, \dots$$

It turns out that this choice of basis results in an order of error of $\mathcal{O}(\omega^{-n-s-1})$, where $n + 1$ is equal to the number of equations in the collocation system (1), assuming that the regularity condition is satisfied. This has the wonderful property that adding collocation points within the interval of integration increases the order. See [4] for a proof of the order of error. The right-hand side of Figure 2 demonstrates the effectiveness of this choice of basis. Many more examples can be found in [4].

4 Multivariate Levin-type Method

In this section, based on work from [5], we will discuss how to generalize Levin-type methods for integrating

$$I_g[f, \Omega] = \int_{\Omega} f e^{i\omega g} dV,$$

where $\Omega \subset \mathbb{R}^d$ is a multivariate piecewise smooth domain and g has no critical points in the closure of Ω , i.e., $\nabla g \neq 0$. We emphasize the dependence of I on g and Ω in this section, as we will need to deal with multiple oscillators in order to derive a Levin-type method. We will similarly denote a univariate Levin-type method as $Q_g^L[f, \Omega]$, for $\Omega = (a, b)$. For simplicity we will demonstrate how to derive a multivariate Levin-type method on a two-dimensional quarter unit circle H as seen in Figure 3, though the technique discussed can readily be generalized to other domains—including higher dimensional domains.

The asymptotic expansion and Filon-type methods were generalized to higher dimensional simplices and polytopes in [2]. Suppose that Ω is a polytope such that the oscillator g is not orthogonal to the boundary of Ω at any point on the boundary, which we call the non-resonance condition. From [2] we know that there exists an asymptotic expansion of the form

$$I_g[f, \Omega] \sim \sum_{k=0}^{\infty} \frac{1}{(-i\omega)^{k+d}} \Theta_k[f], \tag{2}$$

where $\Theta_k[f]$ depends on f and its partial derivatives of order less than or equal to k , evaluated at the vertices of Ω . Hence, if we interpolate f by a polynomial v at the vertices of Ω with multiplicities at least $s - 1$, then $I[f - v] = \mathcal{O}(\omega^{-s-d})$.

We will now use this asymptotic expansion to construct a multivariate Levin-type method. In the univariate case, we determined the collocation operator \mathcal{L} using the fundamental theorem of calculus. We mimic this by using the Stokes' theorem. Define

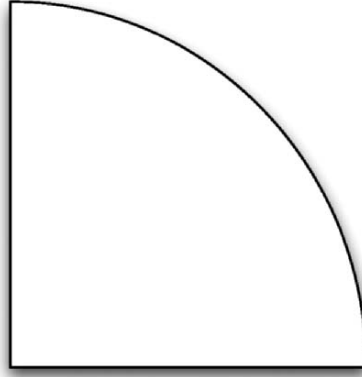


Fig. 3. Diagram of a unit quarter circle H .

the differential form $\rho = v(x, y)e^{i\omega g(x, y)}(dx + dy)$, where $v(x, y) = \sum c_k \psi_k(x, y)$ for some basis $\{\psi_k\}$. Then

$$\begin{aligned} d\rho &= (v_x + i\omega g_x v)e^{i\omega g} dx \wedge dy + (v_y + i\omega g_y v)e^{i\omega g} dy \wedge dx \\ &= (v_x + i\omega g_x v - v_y - i\omega g_y v)e^{i\omega g} dx \wedge dy. \end{aligned}$$

Define the collocation operator $\mathcal{L}[v] = v_x + i\omega g_x v - v_y - i\omega g_y v$. For some sequence of nodes $\{\mathbf{x}_0, \dots, \mathbf{x}_\nu\} \subset \mathbb{R}^2$ and multiplicities $\{m_0, \dots, m_\nu\}$, we can determine the coefficients c_k by solving the system

$$\mathcal{D}^{\mathbf{m}} \mathcal{L}[v](\mathbf{x}_k) = \mathcal{D}^{\mathbf{m}} f(\mathbf{x}_k), \quad 0 \leq |\mathbf{m}| \leq m_k - 1, \quad k = 0, 1, \dots, \nu, \quad (3)$$

where $\mathbf{m} \in \mathbb{N}^2$, $|\mathbf{m}|$ is the sum of the rows of the vector \mathbf{m} and $\mathcal{D}^{\mathbf{m}}$ is the partial derivative operator. We then obtain, using $T_1(t) = [\cos t, \sin t]^\top$, $T_2(t) = [0, 1 - t]^\top$, and $T_3(t) = [t, 0]^\top$ as the positively oriented boundary,

$$\begin{aligned} I_g[f, \Omega] &\approx I_g[\mathcal{L}[v], \Omega] = \iint_H d\rho = \oint_{\partial H} \rho = \oint_{\partial H} v e^{i\omega g} (dx + dy) \\ &= \int_0^{\frac{\pi}{2}} v(T_1(t)) e^{i\omega g(T_1(t))} [1, 1] T_1'(t) dt \\ &\quad + \int_0^1 v(T_2(t)) e^{i\omega g(T_2(t))} [1, 1] T_2'(t) dt \\ &\quad + \int_0^1 v(T_3(t)) e^{i\omega g(T_3(t))} [1, 1] T_3'(t) dt \\ &= \int_0^{\frac{\pi}{2}} v(\cos t, \sin t) e^{i\omega g(\cos t, \sin t)} (\cos t - \sin t) dt \\ &\quad - \int_0^1 v(0, 1 - t) e^{i\omega g(0, 1-t)} dt + \int_0^1 v(t, 0) e^{i\omega g(t, 0)} dt. \end{aligned} \quad (4)$$

This is the sum of three univariate highly oscillatory integrals, with oscillators $e^{i\omega g(\cos t, \sin t)}$, $e^{i\omega g(0, 1-t)}$, and $e^{i\omega g(t, 0)}$. If we assume that these three oscillators have no stationary points, then we can approximate each of these integrals with a univariate Levin-type method, as described above. Hence we define:

$$Q_g^L[f, H] = Q_{g_1}^L[f_1, \left(0, \frac{\pi}{2}\right)] + Q_{g_2}^L[f_2, (0, 1)] + Q_{g_3}^L[f_3, (0, 1)],$$

for

$$\begin{aligned} f_1(t) &= v(\cos t, \sin t)(\cos t - \sin t), & g_1(t) &= g(\cos t, \sin t), \\ f_2(t) &= -v(0, 1 - t), & g_2(t) &= g(0, 1 - t), \\ f_3(t) &= v(t, 0), & g_3(t) &= g(t, 0). \end{aligned}$$

For the purposes of proving the order, we assume that the multiplicity at each endpoint of these univariate Levin-type methods is equal to the multiplicity at the point mapped to by the respective T_k .

Note that requiring that the univariate oscillators be free of stationary points is equivalent to requiring that ∇g is not orthogonal to the boundary of H , i.e., the non-resonance condition. Indeed,

$$\nabla g(T_k(t))^\top T_k'(t) = (g \circ T_k)'(t) = g'_k(t),$$

hence $g'_k(\xi) = 0$ if and only if ∇g is orthogonal to the boundary of H at the point $T_k(\xi)$. We also have a multivariate version of the regularity condition, which simply states that each univariate Levin-type method satisfies the regularity condition, and that the two-dimensional basis $\{(g_x - g_y)\psi_k\}$ can interpolate f at the given nodes and multiplicities. It turns out, subject to the non-resonance condition and the regularity condition, that $I_g[f, H] - Q_g^L[f, H] = \mathcal{O}(\omega^{-s-2})$, for s equal to the minimum of the multiplicities at the vertices of H .

From [5], we know that the asymptotic expansion (2) can be generalized to the non-polytope domain H , depending on the vertices of H . Hence we first show that $I_g[f, H] - I_g[\mathcal{L}[v], H] = \mathcal{O}(\omega^{-s-2})$. The proof of this is almost identical to univariate case. We show that $\mathcal{L}[v]$ is bounded for increasing ω . As before the system (3) can be written as $A\mathbf{c} = \mathbf{f}$, where again $A = P + i\omega G$ for matrices P and G independent of ω , and G is the matrix associated with interpolation at the given nodes and multiplicities by the basis $\{(g_x - g_y)\psi_k\}$. The new regularity condition ensures that $\det G \neq 0$, hence, again due to Cramer's rule, each c_k is of order $\mathcal{O}(\omega^{-1})$. Thus $\mathcal{L}[v] = \mathcal{O}(1)$ for increasing ω , and the asymptotic expansion shows that $I_g[f, H] - I_g[\mathcal{L}[v], H] = I_g[f - \mathcal{L}[v], H] = \mathcal{O}(\omega^{-s-2})$.

We now show that $I_g[\mathcal{L}[v], H] - Q_g^L[f, H] = \mathcal{O}(\omega^{-s-2})$. Note that (4) is equal to $I_g[\mathcal{L}[v], H]$. But we know that each integrand f_k is of order $\mathcal{O}(\omega^{-1})$. It follows that when we approximate these integrals using Q^L the error is of order $\mathcal{O}(\omega^{-s-2})$. A proof for general domains, as well as a generalization of the asymptotic basis, can be found in [5].

We now demonstrate the effectiveness of this method. Consider the case where $f(x, y) = \cos(x - 2y)$, with oscillator $g(x, y) = x^2 + x - y$. The univariate integrals will have oscillators $g_1(t) = \cos^2 t + \cos t - \sin t$, $g_2(t) = t - 1$, and $g_3(t) = t^2 + t$. Since these oscillators are free from stationary points, the non-resonance condition is satisfied. If we collocate at the vertices with multiplicities all one, then we obtain the left-hand side of Figure 4. Increasing the multiplicities to two and adding the interpolation point $[\frac{1}{3}, \frac{1}{3}]$ with multiplicity one gives us the right-hand side. This results in the order increasing by one. More examples can be found in [5].

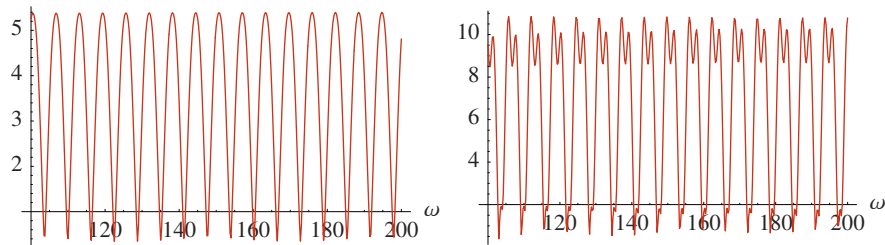


Fig. 4. The error scaled by ω^3 of $Q_g^L[f, H]$ collocating only at the vertices with multiplicities all one (left), and the error scaled by ω^4 collocating at the vertices with multiplicities two and the point $[\frac{1}{3}, \frac{1}{3}]$ with multiplicity one (right), for $I_g[f, H] = \int_H \cos(x - 2y) e^{i\omega(x^2+x-y)} dV$.

References

1. A. Iserles and S.P. Nørsett: Efficient quadrature of highly oscillatory integrals using derivatives. *Proceedings Royal Soc. A.* **461**, 2005, 1383–1399.
2. A. Iserles and S.P. Nørsett: *Quadrature Methods for Multivariate Highly Oscillatory Integrals Using Derivatives*. Technical report NA2005/02, DAMTP, University of Cambridge. *Math. Comp.*, to appear.
3. D. Levin: Analysis of a collocation method for integrating rapidly oscillatory functions. *J. Comput. Appl. Math.* **78**, 1997, 131–138.
4. S. Olver: *Moment-Free Numerical Integration of Highly Oscillatory Functions*. Technical report NA2005/04, DAMTP, University of Cambridge. *IMA Journal of Numerical Analysis*, to appear.
5. S. Olver: *On the Quadrature of Multivariate Highly Oscillatory Integrals over Non-Polytope Domains*. Technical report NA2005/07, DAMTP, University of Cambridge, 2005.
6. M.J.D. Powell: *Approximation Theory and Methods*. Cambridge University Press, Cambridge, 1981.
7. E. Stein: *Harmonic Analysis: Real-Variable Methods, Orthogonality, and Oscillatory Integrals*. Princeton University Press, Princeton, NJ, 1993.

Index

- asymptotic
 - approximation, 335
 - polynomials, 242
 - series, 344
- ball source
 - biharmonic, 214
 - cubic, 214
 - explicit, 212
 - Gaussian, 214
 - linear, 214
 - triharmonic, 214
- basin of attraction, 119
- Bayes
 - linear methodology, 188
- Bayesian
 - inversion, 147
 - statistics, 158, 186
- bivariate spline, 219
- Blasius equation, 274
- Bochner
 - measure, 352
 - theorem, 351
- Buckley-Leverett equation, 96
- calibration curve, 181
- cell
 - alignment, 126
 - motility, 124
- cluster analysis, 70
- clustering, 31, 76
 - algorithm, 32
 - kernel-based, 37
- compact homogeneous manifold, 358
- computational intelligence, 31
- convolution, 351
- coordinate measuring machine, 168
- Dawson's integral, 273
- delay differential equation
 - scalar periodic, 297
 - stochastic, 287, 308
- dimension-elevation, 27
- Dirichlet series, 339
- discrete quasi-interpolation, 229
- distance defect ratio, 23
- domain singularity, 23
- dynamical system, 113
- elasticity, 135
- energy
 - estimate, 358
 - on manifold, 365
- estimation procedure
 - sequential, 289
- Euler-Maclaurin formula, 333
- experimental design, 190
- explicit
 - ball source, 212
 - line source, 208
- exponential integral, 344
- factorial series, 338
- feature extraction, 72
- Filon-type method, 380
- filtering, 52
- five-spot problem, 95
- flexi-knot spline, 250

- friction contact, 141
- Gauss-Markov regression, 167
- Gaussian hypergeometric series, 341
- Gelfand
 - pair, 351
 - transform, 354
- generalized
 - distance regression, 172
 - footpoint problem, 177
- generalized motion group, 349, 353
- geometry-driven binary partition, 25
- highly oscillatory integral, 378
- hydrodynamic equation, 105
- hyperbolic conservation law, 85
- image
 - approximation, 19
 - denoising, 51
 - simplification, 51
 - smoothing, 51
- integral
 - equation
 - Volterra, 319
 - highly oscillatory, 378
 - interpolation, 203
 - weighted, 279
- integro-differential equation, 124
- interpolation
 - integral, 203
- inverse problem, 147, 162
- kernel-based learning, 37, 66
- knot density function, 250
- kriging, 161, 349
- laser tracker measurement, 169
- Levin-type method, 381
- line source
 - cubic, 211
 - explicit, 208
 - Gaussian, 209
 - linear, 209
 - multiquadric, 212
 - thinplate spline, 212
- Lyapunov
 - exponent, 308
 - function, 118
- maximum probability interpolation, 161
- metrology, 167
- minimal discrete energy problem, 368
- moving least-squares, 108
- network training, 64
- neural network, 34
 - ranking, 8, 11
- neutral data fitting, 259
- numerical
 - analytic continuation, 337
 - quadrature, 378
- Padé approximants, 338
- particle
 - flow simulation, 97
 - method, 86
 - finite volume, 87
 - semi-Lagrangian, 86
- perceptron, 7
- Plancherel
 - measure, 354
 - theorem, 354
- polyharmonic spline, 89
- polynomial
 - approximation, 20
 - spline, 279
- positive definite
 - function, 349
 - kernel, 359
- Powell-Sabin
 - finite element, 220
 - quasi-interpolation, 219
 - tensioned quadratic B-spline, 225
- quasi-interpolation
 - discrete, 229
 - Powell-Sabin, 219
- radial basis function, 61, 108, 113, 161, 269, 349
 - polyharmonic spline, 83
- random field, 150, 158
- ranking, 3
- reflection invariant function, 355
- regression
 - Gauss-Markov, 167
- reservoir
 - forecasting, 186

- simulation, 95, 187
- Riemann zeta function, 331, 339
- Riesz kernel, 365
- scattered data, 350
 - interpolation, 148, 350
- sequential estimator, 289
- shape control, 219
- smoothed particle hydrodynamics, 103
- special function, 331
- spectral Galerkin method, 135
- spherical
 - Bessel function, 352
 - function, 351
- spline
 - approximation, 249
 - collocation, 319
 - flexi-knot, 250
 - polynomial, 279
 - projection, 319
- stochastic sampling, 161
- strictly positive definite function, 349, 353
- support vector machine, 6
- surface fitting, 182
- tension property, 229
- Thomas-Fermi equation, 272
- Tikhonov regularisation, 53, 163
- tracer transportation, 93
- track data approximation, 215
- traveling salesman problem, 39
- truncation error, 335
- uncertainty
 - evaluation, 186
 - matrix, 168
- voice
 - conversion, 66
 - morphing, 61
- Volterra integral equation, 319
- wavelet analysis, 64
- weighted integral, 279
- Weil's formula, 354
- WENO reconstruction, 88
- Weyl criterion, 363
- zonal basis function, 350