

INTERDISCIPLINARY MATHEMATICAL SCIENCES

Series Editor: Jinqiao Duan (*Illinois Inst. of Tech., USA*)

Editorial Board: Ludwig Arnold, Roberto Camassa, Peter Constantin,
Charles Doering, Paul Fischer, Andrei V. Fursikov, Fred R. McMorris,
Daniel Schertzer, Bjorn Schmalfuss, Xiangdong Ye, and
Jerzy Zabczyk

Published

Vol. 1: Global Attractors of Nonautonomous Dissipative Dynamical Systems

David N. Cheban

Vol. 2: Stochastic Differential Equations: Theory and Applications

A Volume in Honor of Professor Boris L. Rozovskii
eds. Peter H. Baxendale & Sergey V. Lototsky

Vol. 3: Amplitude Equations for Stochastic Partial Differential Equations

Dirk Blömker

Vol. 4: Mathematical Theory of Adaptive Control

Vladimir G. Sragovich

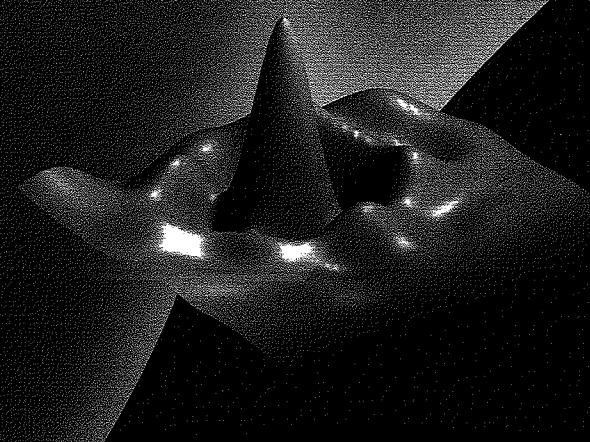
Vol. 5: The Hilbert–Huang Transform and Its Applications

Norden E. Huang & Samuel S. P. Shen

Vol. 6: Meshfree Approximation Methods with MATLAB

Gregory E. Fasshauer

Interdisciplinary Mathematical Sciences – Vol. 6



Meshfree Approximation Methods with MATLAB

Gregory E. Fasshauer

Illinois Institute of Technology, USA

 **World Scientific**

NEW JERSEY • LONDON • SINGAPORE • BEIJING • SHANGHAI • HONG KONG • TAIPEI • CHENNAI

Published by

World Scientific Publishing Co. Pte. Ltd.
5 Toh Tuck Link, Singapore 596224
USA office: 27 Warren Street, Suite 401-402, Hackensack, NJ 07601
UK office: 57 Shelton Street, Covent Garden, London WC2H 9HE

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

MESHFREE APPROXIMATION METHODS WITH MATLAB
(With CD-ROM)

Interdisciplinary Mathematical Sciences — Vol. 6

Copyright © 2007 by World Scientific Publishing Co. Pte. Ltd.

All rights reserved. This book, or parts thereof, may not be reproduced in any form or by any means, electronic or mechanical, including photocopying, recording or any information storage and retrieval system now known or to be invented, without written permission from the Publisher.

For photocopying of material in this volume, please pay a copying fee through the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, USA. In this case permission to photocopy is not required from the publisher.

ISBN-13 978-981-270-633-1
ISBN-10 981-270-633-X
ISBN-13 978-981-270-634-8 (pbk)
ISBN-10 981-270-634-8 (pbk)

This book is dedicated to
Inge, Conny, Marc and Patrick.

Preface

Traditional numerical methods, such as finite element, finite difference, or finite volume methods, were motivated mostly by early one- and two-dimensional simulations of engineering problems via partial differential equations (PDEs). The discretization involved in all of these methods requires some sort of underlying computational mesh, *e.g.*, a triangulation of the region of interest. Creation of these meshes (and possible re-meshing) becomes a rather difficult task in three dimensions, and virtually impossible for higher-dimensional problems. This is where *meshfree* methods enter the picture. Meshfree methods are often — but by no means have to be — radially symmetric in nature. This is achieved by composing some univariate basic function with a (Euclidean) norm, and therefore turning a problem involving many space dimensions into one that is virtually one-dimensional. Such *radial basis functions* are at the heart of this book. Some people have argued that there are three “big technologies” for the numerical solution of PDEs, namely finite difference, finite element, and spectral methods. While these technologies came into their own right in successive decades, namely finite difference methods in the 1950s, finite element methods in the 1960s, and spectral methods in the 1970s, meshfree methods started to appear in the mathematics literature in the 1980s, and they are now on their way to becoming “big technology” number four. In fact, we will demonstrate in later parts of this book how different types of meshfree methods can be viewed as generalizations of the traditional “big three”.

Multivariate meshfree approximation methods are being studied by many researchers. They exist in many flavors and are known under many names, *e.g.*, diffuse element method, element-free Galerkin method, generalized finite element method, *hp*-clouds, meshless local Petrov-Galerkin method, moving least squares method, partition of unity finite element method, radial basis function method, reproducing kernel particle method, smooth particle hydrodynamics method.

In this book we are concerned mostly with the moving least squares (MLS) and radial basis function (RBF) methods. We will consider all different kinds of aspects of these meshfree approximation methods: How to construct them? Are these constructions mathematically justifiable? How accurate are they? Are there ways to implement them efficiently with standard mathematical software packages such

as MATLAB? How do they compare with traditional methods? How do the various flavors of meshfree methods differ from one another, and how are they similar to one another? Is there a general framework that captures all of these methods? What sort of applications are they especially well suited for?

While we do present much of the underlying theory for RBF and MLS approximation methods, the emphasis in this book is not on proofs. For readers who are interested in all the mathematical details and intricacies of the theory we recommend the two excellent recent monographs [Buhmann (2003); Wendland (2005a)]. Instead, our objective is to make the theory accessible to a wide audience that includes graduate students and practitioners in all sorts of science and engineering fields. We want to put the mathematical theory in the context of applications and provide MATLAB implementations which give the reader an easy entry into meshfree approximation methods. The skilled reader should then easily be able to modify the programs provided here for his/her specific purposes.

In a certain sense the present book was inspired by the beautiful little book [Trefethen (2000)]. While the present book is much more expansive (filling more than five hundred pages with forty-seven MATLAB¹ programs, one Maple² program, one hundred figures, more than fifty tables, and more than five hundred references), it is our aim to provide the reader with relatively simple MATLAB code that illustrates just about every aspect discussed in the book.

All MATLAB programs printed in the text (as well as a few modifications discussed) are also included on the enclosed CD. The folder MATLAB contains M-files and data files of type MAT that have been written and tested with MATLAB 7. For those readers who do not have access to MATLAB 7, the folder MATLAB6 contains versions of these files that are compatible with the older MATLAB release. The main difference between the two versions is the use of anonymous functions in the MATLAB 7 code as compared to inline functions in the MATLAB 6 version. Two packages from the MATLAB Central File Exchange [MCFE] are used throughout the book: the function haltonseq written by Daniel Dougherty and used to generate sequences of Halton points; the *kd*-tree library (given as a set of MATLAB MEX-files) written by Guy Shechter and used to generate the *kd*-tree data structure underlying our sparse matrices based on compactly supported basis functions. Both of these packages are discussed in Appendix A and need to be downloaded separately. The folder Maple on the CD contains the one Maple file mentioned above.

The manuscript for this book and some of its earlier incarnations have been used in graduate level courses and seminars at Northwestern University, Vanderbilt University, and the Illinois Institute of Technology. Special thanks are due to Jon

¹MATLAB® is a trademark of The MathWorks, Inc. and is used with permission. The MathWorks does not warrant the accuracy of the text or exercises in this book. This book's use or discussion of MATLAB software or related products does not constitute endorsement or sponsorship by The MathWorks of a particular pedagogical approach or particular use of the MATLAB software.

²Maple™ is a registered trademark of Waterloo Maple Inc.

Cherrie, John Erickson, Paritosh Mokhasi, Larry Schumaker, and Jack Zhang for reading various portions of the manuscript and/or MATLAB code and providing helpful feedback. Finally, thanks are due to all the people at World Scientific Publishing Co. who helped make this project a success: Rajesh Babu, Ying Oi Chiew, Linda Kwan, Rok Ting Tan, and Yubing Zhai.

Greg Fasshauer
Chicago, IL, January 2007

Contents

<i>Preface</i>	vii
1. Introduction	1
1.1 Motivation: Scattered Data Interpolation in \mathbb{R}^s	2
1.1.1 The Scattered Data Interpolation Problem	2
1.1.2 Example: Interpolation with Distance Matrices	4
1.2 Some Historical Remarks	13
2. Radial Basis Function Interpolation in MATLAB	17
2.1 Radial (Basis) Functions	17
2.2 Radial Basis Function Interpolation	19
3. Positive Definite Functions	27
3.1 Positive Definite Matrices and Functions	27
3.2 Integral Characterizations for (Strictly) Positive Definite Functions	31
3.2.1 Bochner's Theorem	31
3.2.2 Extensions to Strictly Positive Definite Functions	32
3.3 Positive Definite Radial Functions	33
4. Examples of Strictly Positive Definite Radial Functions	37
4.1 Example 1: Gaussians	37
4.2 Example 2: Laguerre-Gaussians	38
4.3 Example 3: Poisson Radial Functions	39
4.4 Example 4: Matérn Functions	41
4.5 Example 5: Generalized Inverse Multiquadratics	41
4.6 Example 6: Truncated Power Functions	42
4.7 Example 7: Potentials and Whittaker Radial Functions	43
4.8 Example 8: Integration Against Strictly Positive Definite Kernels	45

4.9	Summary	45
5.	Completely Monotone and Multiply Monotone Functions	47
5.1	Completely Monotone Functions	47
5.2	Multiply Monotone Functions	49
6.	Scattered Data Interpolation with Polynomial Precision	53
6.1	Interpolation with Multivariate Polynomials	53
6.2	Example: Reproduction of Linear Functions Using Gaussian RBFs	55
6.3	Scattered Data Interpolation with More General Polynomial Precision	57
6.4	Conditionally Positive Definite Matrices and Reproduction of Constant Functions	59
7.	Conditionally Positive Definite Functions	63
7.1	Conditionally Positive Definite Functions Defined	63
7.2	Conditionally Positive Definite Functions and Generalized Fourier Transforms	65
8.	Examples of Conditionally Positive Definite Functions	67
8.1	Example 1: Generalized Multiquadratics	67
8.2	Example 2: Radial Powers	69
8.3	Example 3: Thin Plate Splines	70
9.	Conditionally Positive Definite Radial Functions	73
9.1	Conditionally Positive Definite Radial Functions and Completely Monotone Functions	73
9.2	Conditionally Positive Definite Radial Functions and Multiply Monotone Functions	75
9.3	Some Special Properties of Conditionally Positive Definite Functions of Order One	76
10.	Miscellaneous Theory: Other Norms and Scattered Data Fitting on Manifolds	79
10.1	Conditionally Positive Definite Functions and p -Norms	79
10.2	Scattered Data Fitting on Manifolds	83
10.3	Remarks	83
11.	Compactly Supported Radial Basis Functions	85
11.1	Operators for Radial Functions and Dimension Walks	85
11.2	Wendland's Compactly Supported Functions	87

11.3	Wu's Compactly Supported Functions	88
11.4	Oscillatory Compactly Supported Functions	90
11.5	Other Compactly Supported Radial Basis Functions	92
12.	Interpolation with Compactly Supported RBFs in MATLAB	95
12.1	Assembly of the Sparse Interpolation Matrix	95
12.2	Numerical Experiments with CSRBFS	99
13.	Reproducing Kernel Hilbert Spaces and Native Spaces for Strictly Positive Definite Functions	103
13.1	Reproducing Kernel Hilbert Spaces	103
13.2	Native Spaces for Strictly Positive Definite Functions	105
13.3	Examples of Native Spaces for Popular Radial Basic Functions	108
14.	The Power Function and Native Space Error Estimates	111
14.1	Fill Distance and Approximation Orders	111
14.2	Lagrange Form of the Interpolant and Cardinal Basis Functions	112
14.3	The Power Function	115
14.4	Generic Error Estimates for Functions in $\mathcal{N}_\Phi(\Omega)$	117
14.5	Error Estimates in Terms of the Fill Distance	119
15.	Refined and Improved Error Bounds	125
15.1	Native Space Error Bounds for Specific Basis Functions	125
15.1.1	Infinitely Smooth Basis Functions	125
15.1.2	Basis Functions with Finite Smoothness	126
15.2	Improvements for Native Space Error Bounds	127
15.3	Error Bounds for Functions Outside the Native Space	128
15.4	Error Bounds for Stationary Approximation	130
15.5	Convergence with Respect to the Shape Parameter	132
15.6	Polynomial Interpolation as the Limit of RBF Interpolation	133
16.	Stability and Trade-Off Principles	135
16.1	Stability and Conditioning of Radial Basis Function Interpolants	135
16.2	Trade-Off Principle I: Accuracy vs. Stability	138
16.3	Trade-Off Principle II: Accuracy and Stability vs. Problem Size	140
16.4	Trade-Off Principle III: Accuracy vs. Efficiency	140
17.	Numerical Evidence for Approximation Order Results	141
17.1	Interpolation for $\varepsilon \rightarrow 0$	141
17.1.1	Choosing a Good Shape Parameter via Trial and Error	142

17.1.2	The Power Function as Indicator for a Good Shape Parameter	142	23.1	Shepard's Method	205
17.1.3	Choosing a Good Shape Parameter via Cross Validation	146	23.2	MLS Approximation with Nontrivial Polynomial Reproduction	207
17.1.4	The Contour-Padé Algorithm	151	24.	MLS Approximation with MATLAB	211
17.1.5	Summary	152	24.1	Approximation with Shepard's Method	211
17.2	Non-stationary Interpolation	153	24.2	MLS Approximation with Linear Reproduction	216
17.3	Stationary Interpolation	155	24.3	Plots of Basis-Dual Basis Pairs	222
18.	The Optimality of RBF Interpolation	159	25.	Error Bounds for Moving Least Squares Approximation	225
18.1	The Connection to Optimal Recovery	159	25.1	Approximation Order of Moving Least Squares	225
18.2	Orthogonality in Reproducing Kernel Hilbert Spaces	160	26.	Approximate Moving Least Squares Approximation	229
18.3	Optimality Theorem I	162	26.1	High-order Shepard Methods via Moment Conditions	229
18.4	Optimality Theorem II	163	26.2	Approximate Approximation	230
18.5	Optimality Theorem III	164	26.3	Construction of Generating Functions for Approximate MLS Approximation	232
19.	Least Squares RBF Approximation with MATLAB	165	27.	Numerical Experiments for Approximate MLS Approximation	237
19.1	Optimal Recovery Revisited	165	27.1	Univariate Experiments	237
19.2	Regularized Least Squares Approximation	166	27.2	Bivariate Experiments	241
19.3	Least Squares Approximation When RBF Centers Differ from Data Sites	168	28.	Fast Fourier Transforms	243
19.4	Least Squares Smoothing of Noisy Data	170	28.1	NFFT	243
20.	Theory for Least Squares Approximation	177	28.2	Approximate MLS Approximation via Non-uniform Fast Fourier Transforms	245
20.1	Well-Posedness of RBF Least Squares Approximation	177	29.	Partition of Unity Methods	249
20.2	Error Bounds for Least Squares Approximation	179	29.1	Theory	249
21.	Adaptive Least Squares Approximation	181	29.2	Partition of Unity Approximation with MATLAB	251
21.1	Adaptive Least Squares using Knot Insertion	181	30.	Approximation of Point Cloud Data in 3D	255
21.2	Adaptive Least Squares using Knot Removal	184	30.1	A General Approach via Implicit Surfaces	255
21.3	Some Numerical Examples	188	30.2	An Illustration in 2D	257
22.	Moving Least Squares Approximation	191	30.3	A Simplistic Implementation in 3D via Partition of Unity Approximation in MATLAB	260
22.1	Discrete Weighted Least Squares Approximation	191	31.	Fixed Level Residual Iteration	265
22.2	Standard Interpretation of MLS Approximation	192	31.1	Iterative Refinement	265
22.3	The Backus-Gilbert Approach to MLS Approximation	194	31.2	Fixed Level Iteration	267
22.4	Equivalence of the Two Formulations of MLS Approximation	198	31.3	Modifications of the Basic Fixed Level Iteration Algorithm	269
22.5	Duality and Bi-Orthogonal Bases	199	31.4	Iterated Approximate MLS Approximation in MATLAB	270
22.6	Standard MLS Approximation as a Constrained Quadratic Optimization Problem	202	31.5	Iterated Shepard Approximation	274
22.7	Remarks	202			
23.	Examples of MLS Generating Functions	205			

32.	Multilevel Iteration	277
32.1	Stationary Multilevel Interpolation	277
32.2	A MATLAB Implementation of Stationary Multilevel Interpolation	279
32.3	Stationary Multilevel Approximation	283
32.4	Multilevel Interpolation with Globally Supported RBFs	287
33.	Adaptive Iteration	291
33.1	A Greedy Adaptive Algorithm	291
33.2	The Faul-Powell Algorithm	298
34.	Improving the Condition Number of the Interpolation Matrix	303
34.1	Preconditioning: Two Simple Examples	304
34.2	Early Preconditioners	305
34.3	Preconditioned GMRES via Approximate Cardinal Functions . .	309
34.4	Change of Basis	311
34.5	Effect of the “Better” Basis on the Condition Number of the Interpolation Matrix	314
34.6	Effect of the “Better” Basis on the Accuracy of the Interpolant .	316
35.	Other Efficient Numerical Methods	321
35.1	The Fast Multipole Method	321
35.2	Fast Tree Codes	327
35.3	Domain Decomposition	331
36.	Generalized Hermite Interpolation	333
36.1	The Generalized Hermite Interpolation Problem	333
36.2	Motivation for the Symmetric Formulation	335
37.	RBF Hermite Interpolation in MATLAB	339
38.	Solving Elliptic Partial Differential Equations via RBF Collocation	345
38.1	Kansa’s Approach	345
38.2	An Hermite-based Approach	348
38.3	Error Bounds for Symmetric Collocation	349
38.4	Other Issues	350
39.	Non-Symmetric RBF Collocation in MATLAB	353
39.1	Kansa’s Non-Symmetric Collocation Method	353
40.	Symmetric RBF Collocation in MATLAB	365

40.1	Symmetric Collocation Method	365
40.2	Summarizing Remarks on the Symmetric and Non-Symmetric Collocation Methods	372
41.	Collocation with CSRBFS in MATLAB	375
41.1	Collocation with Compactly Supported RBFs	375
41.2	Multilevel RBF Collocation	380
42.	Using Radial Basis Functions in Pseudospectral Mode	387
42.1	Differentiation Matrices	388
42.2	PDEs with Boundary Conditions via Pseudospectral Methods .	390
42.3	A Non-Symmetric RBF-based Pseudospectral Method	391
42.4	A Symmetric RBF-based Pseudospectral Method	394
42.5	A Unified Discussion	396
42.6	Summary	398
43.	RBF-PS Methods in MATLAB	401
43.1	Computing the RBF-Differentiation Matrix in MATLAB	401
43.1.1	Solution of a 1-D Transport Equation	403
43.2	Use of the Contour-Padé Algorithm with the PS Approach . . .	405
43.2.1	Solution of the 1D Transport Equation Revisited	405
43.3	Computation of Higher-Order Derivatives	407
43.3.1	Solution of the Allen-Cahn Equation	409
43.4	Solution of a 2D Helmholtz Equation	411
43.5	Solution of a 2D Laplace Equation with Piecewise Boundary Conditions	415
43.6	Summary	416
44.	RBF Galerkin Methods	419
44.1	An Elliptic PDE with Neumann Boundary Conditions	419
44.2	A Convergence Estimate	420
44.3	A Multilevel RBF Galerkin Algorithm	421
45.	RBF Galerkin Methods in MATLAB	423
Appendix A Useful Facts from Discrete Mathematics		427
A.1	Halton Points	427
A.2	<i>kd</i> -Trees	428
Appendix B Useful Facts from Analysis		431
B.1	Some Important Concepts from Measure Theory	431
B.2	A Brief Summary of Integral Transforms	432

B.3	The Schwartz Space and the Generalized Fourier Transform	433
Appendix C	Additional Computer Programs	435
C.1	MATLAB Programs	435
C.2	Maple Programs	440
Appendix D	Catalog of RBFs with Derivatives	443
D.1	Generic Derivatives	443
D.2	Formulas for Specific Basic Functions	444
D.2.1	Globally Supported, Strictly Positive Definite Functions .	444
D.2.2	Globally Supported, Strictly Conditionally Positive Definite Functions of Order 1	445
D.2.3	Globally Supported, Strictly Conditionally Positive Definite Functions of Order 2	446
D.2.4	Globally Supported, Strictly Conditionally Positive Definite Functions of Order 3	446
D.2.5	Globally Supported, Strictly Conditionally Positive Definite Functions of Order 4	447
D.2.6	Globally Supported, Strictly Positive Definite and Oscillatory Functions	447
D.2.7	Compactly Supported, Strictly Positive Definite Functions	448
<i>Bibliography</i>		451
<i>Index</i>		491

Chapter 1

Introduction

Meshfree methods have gained much attention in recent years, not only in the mathematics but also in the engineering community. Thus, much of the work concerned with meshfree approximation methods is interdisciplinary — at the interface between mathematics and numerous application areas (see the partial list below). Moreover, computation with high-dimensional data is an important issue in many areas of science and engineering. Many traditional numerical methods can either not handle such problems at all, or are limited to very special (regular) situations. Meshfree methods are often better suited to cope with changes in the geometry of the domain of interest (*e.g.*, free surfaces and large deformations) than classical discretization techniques such as finite differences, finite elements or finite volumes. Another obvious advantage of meshfree discretizations is — of course — their independence from a mesh. Mesh generation is still the most time consuming part of any mesh-based numerical simulation. Since meshfree discretization techniques are based only on a set of independent points, these costs of mesh generation are eliminated. Meshfree approximation methods can be seen to provide a new generation of numerical tools. Other traditional numerical methods such as the finite element, finite difference or finite volume methods are usually limited to problems involving two or three parameters (space dimensions). However, in many applications the number of parameters can easily range in the hundreds or even thousands. Multivariate approximation methods present one way to address these issues.

Applications of meshfree methods can be found

- in many different areas of science and engineering via *scattered data modeling* (*e.g.*, fitting of potential energy surfaces in chemistry; coupling of engineering models with sets of incompatible parameters; mapping problems in geodesy, geophysics, meteorology);
- in many different areas of science and engineering via *solution of partial differential equations* (*e.g.*, solution of gas dynamics equations, Boltzmann and Fokker-Planck equations in six-dimensional phase space; problems involving moving discontinuities such as cracks and shocks, multi-scale resolution, large material distortions; elasticity studies in plate and shell bending

- problems; applications in nanotechnology);
- in *non-uniform sampling* (e.g., medical imaging, tomographic reconstruction);
 - in *mathematical finance* (e.g., option pricing);
 - in *computer graphics* (e.g., representation of surfaces from point information such as laser range scan data, image warping);
 - in *learning theory, neural networks* and *data mining* (e.g., kernel approximation, support vector machines);
 - in *optimization*.

Since many of these applications either come down to a function approximation problem, or include function approximation as a fundamental component, we will begin our discussion with — and in fact base a large part of the contents of this book on — the multivariate scattered data interpolation problem.

1.1 Motivation: Scattered Data Interpolation in \mathbb{R}^s

We will now describe the general process of scattered data fitting, which is one of the fundamental problems in approximation theory and data modeling in general. Our desire to have a well-posed problem formulation will naturally lead to an introductory example based on the use of so-called *distance matrices*. In the next chapters we will generalize this approach by introducing the concept of a radial basis function.

1.1.1 The Scattered Data Interpolation Problem

In many scientific disciplines one faces the following problem: We are given a set of data (measurements, and locations at which these measurements were obtained), and we want to find a rule which allows us to deduce information about the process we are studying also at locations different from those at which we obtained our measurements. Thus, we are trying to find a function \mathcal{P}_f which is a “good” fit to the given data. There are many ways to decide what we mean by “good”, and the only criterion we will consider now is that we want the function \mathcal{P}_f to exactly match the given measurements at the corresponding locations. This approach is called *interpolation*, and if the locations at which the measurements are taken do not lie on a uniform or regular grid, then the process is called *scattered data interpolation*.

To give a precise definition we assume that the measurement locations (or *data sites*) are labeled \mathbf{x}_j , $j = 1, \dots, N$, and the corresponding measurements (or *data values*) are called y_j . We will use \mathcal{X} to denote the set of data sites and assume that $\mathcal{X} \subset \Omega$ for some region Ω in \mathbb{R}^s . Throughout this book we will restrict our discussion to scalar-valued data, i.e., $y_j \in \mathbb{R}$. However, much of the following can be generalized easily to problems with vector-valued data. In many of our later

discussions we will assume that the data are obtained by sampling some (unknown) function f at the data sites, i.e., $y_j = f(\mathbf{x}_j)$, $j = 1, \dots, N$. Our notation \mathcal{P}_f for the interpolating function emphasizes the connection between the interpolant and the data function f . We are now ready for a precise formulation of the scattered data interpolation problem.

Problem 1.1 (Scattered Data Interpolation). Given data (\mathbf{x}_j, y_j) , $j = 1, \dots, N$, with $\mathbf{x}_j \in \mathbb{R}^s$, $y_j \in \mathbb{R}$, find a (continuous) function \mathcal{P}_f such that $\mathcal{P}_f(\mathbf{x}_j) = y_j$, $j = 1, \dots, N$.

The fact that we allow \mathbf{x}_j to lie in an arbitrary s -dimensional space \mathbb{R}^s means that the formulation of Problem 1.1 allows us to cover many different types of applications. If $s = 1$ the data could, e.g., be a series of measurements taken over a certain time period, thus the “data sites” \mathbf{x}_j would correspond to certain time instances. For $s = 2$ we can think of the data being obtained over a planar region, and so \mathbf{x}_j corresponds to the two coordinates in the plane. For instance, we might want to produce a map that shows the rainfall in the state we live in based on the data collected at weather stations located throughout the state. For $s = 3$ we might think of a similar situation in space. One possibility is that we could be interested in the temperature distribution inside some solid body. Higher-dimensional examples might not be that intuitive, but a multitude of them exist, e.g., in finance, optimization, economics or statistics, but also in artificial intelligence or learning theory.

A convenient and common approach to solving the scattered data problem is to make the assumption that the function \mathcal{P}_f is a linear combination of certain *basis functions* B_k , i.e.,

$$\mathcal{P}_f(\mathbf{x}) = \sum_{k=1}^N c_k B_k(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^s. \quad (1.1)$$

Solving the interpolation problem under this assumption leads to a system of linear equations of the form

$$\mathbf{A}\mathbf{c} = \mathbf{y},$$

where the entries of the *interpolation matrix* \mathbf{A} are given by $A_{jk} = B_k(\mathbf{x}_j)$, $j, k = 1, \dots, N$, $\mathbf{c} = [c_1, \dots, c_N]^T$, and $\mathbf{y} = [y_1, \dots, y_N]^T$.

Problem 1.1 will be *well-posed*, i.e., a solution to the problem will exist and be unique, if and only if the matrix \mathbf{A} is non-singular.

In the univariate setting it is well known that one can interpolate to arbitrary data at N distinct data sites using a polynomial of degree $N-1$. For the multivariate setting, however, there is the following negative result (see [Mairhuber (1956); Curtis (1959)]).

Theorem 1.1 (Mairhuber-Curtis). If $\Omega \subset \mathbb{R}^s$, $s \geq 2$, contains an interior point, then there exist no Haar spaces of continuous functions except for one-dimensional ones.

In order to understand this theorem we need

Definition 1.1. Let the finite-dimensional linear function space $\mathcal{B} \subseteq C(\Omega)$ have a basis $\{B_1, \dots, B_N\}$. Then \mathcal{B} is a *Haar space* on Ω if

$$\det A \neq 0$$

for any set of distinct x_1, \dots, x_N in Ω . Here A is the matrix with entries $A_{jk} = B_k(x_j)$.

Note that existence of a Haar space guarantees invertibility of the interpolation matrix A , *i.e.*, existence and uniqueness of an interpolant of the form (1.1) to data specified at x_1, \dots, x_N from the space \mathcal{B} . As mentioned above, univariate polynomials of degree $N - 1$ form an N -dimensional Haar space for data given at x_1, \dots, x_N .

The Mairhuber-Curtis theorem tells us that if we want to have a well-posed multivariate scattered data interpolation problem we can no longer fix in advance the set of basis functions we plan to use for interpolation of arbitrary scattered data. For example, it is not possible to perform unique interpolation with (multivariate) polynomials of degree N to data given at arbitrary locations in \mathbb{R}^2 . Instead, the basis should depend on the data locations. We will give a simple example of such an interpolation scheme in the next subsection.

Proof. [of Theorem 1.1] Let $s \geq 2$ and assume that \mathcal{B} is a Haar space with basis $\{B_1, \dots, B_N\}$ with $N \geq 2$. We need to show that this leads to a contradiction.

We let x_1, \dots, x_N be a set of distinct points in $\Omega \subset \mathbb{R}^s$ and A the matrix with entries $A_{jk} = B_k(x_j)$, $j, k = 1, \dots, N$. Then, by the definition of a Haar space, we have

$$\det A \neq 0. \quad (1.2)$$

Now, consider a closed path P in Ω connecting only x_1 and x_2 . This is possible since — by assumption — Ω contains an interior point. We can exchange the positions of x_1 and x_2 by moving them continuously along the path P (without interfering with any of the other x_j). This means, however, that rows 1 and 2 of the determinant (1.2) have been exchanged, and so the determinant has changed sign.

Since the determinant is a continuous function of x_1 and x_2 we must have had $\det = 0$ at some point along P . This contradicts (1.2). \square

1.1.2 Example: Interpolation with Distance Matrices

In order to obtain data dependent approximation spaces as suggested by the Mairhuber-Curtis theorem we now consider a simple example. As a “testfunction” we employ the function

$$f_s(x) = 4^s \prod_{d=1}^s x_d(1 - x_d), \quad x = (x_1, \dots, x_s) \in [0, 1]^s.$$

This function is zero on the boundary of the unit cube in \mathbb{R}^s and has a maximum value of one at the center of the cube. A simple MATLAB script defining f_s is given as Program C.1 in Appendix C.

We will use a set of uniformly scattered data sites in the unit cube at which we sample our testfunction f_s . This will be accomplished here (and in many other examples later on) by resorting to the so-called *Halton points*. These are uniformly distributed random points in $(0, 1)^s$. A set of 289 Halton points in the unit square in \mathbb{R}^2 is shown in Figure 1.1. More details on Halton points are presented in Appendix A. In our computational experiments we generate Halton points using the program `haltonseq.m` written by Daniel Dougherty. This function can be downloaded from the MATLAB Central File Exchange (see [MCFE]).

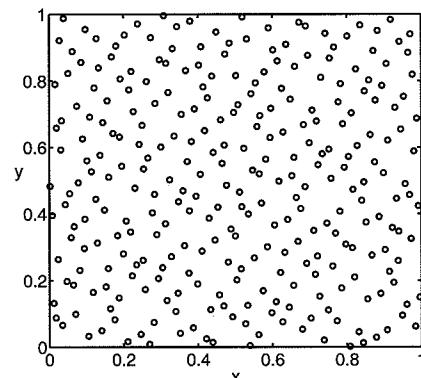


Fig. 1.1 289 Halton points in the unit square in \mathbb{R}^2 .

As explained in the previous subsection we are interested in constructing a (continuous) function \mathcal{P}_f that interpolates the samples obtained from f_s at the set of Halton points, *i.e.*, such that

$$\mathcal{P}_f(x_j) = f_s(x_j), \quad x_j \text{ a Halton point.}$$

As pointed out above, if $s = 1$, then this problem is often solved using univariate polynomials or splines. For a small number of data sites polynomials may work satisfactorily. However, if the number of points increases, *i.e.*, the polynomial degree grows, then it is well known that one should use splines (or piecewise polynomials) to avoid oscillations. The simplest solution is to use a continuous piecewise linear spline, *i.e.*, to “connect the dots”. It is also well known that one possible basis for the space of piecewise linear splines interpolating data at a given set of points in $[0, 1]$ consists of the shifts of the absolute value function to the data sites. In other words, we can construct the piecewise linear spline interpolant by assuming \mathcal{P}_f is of the form

$$\mathcal{P}_f(x) = \sum_{k=1}^N c_k |x - x_k|, \quad x \in [0, 1],$$

and then determine the coefficients c_k by satisfying the interpolation conditions

$$\mathcal{P}_f(x_j) = f_1(x_j), \quad j = 1, \dots, N.$$

Clearly, the basis functions $B_k = |\cdot - x_k|$ are dependent on the data sites as suggested by the Mairhuber-Curtis theorem. The points x_k to which the basic function $B(x) = |x|$ is shifted are usually referred to as *centers*. While there may be circumstances that suggest choosing these centers different from the data sites one generally picks the centers to coincide with the data sites. This simplifies the analysis of the method, and is sufficient for many applications. Since the functions B_k are (radially) symmetric about their centers x_k this constitutes the first example of *radial basis functions*. We will formally introduce the notion of a radial function in the next chapter.

Of course, one can imagine many other ways to construct an N -dimensional data-dependent basis for the purpose of scattered data interpolation. However, the use of shifts of one single basic function makes the radial basis function approach particularly elegant.

Note that we distinguish between *basis* functions B_k and the *basic* function B . We use this terminology to emphasize that there is one basic function B which generates the basis via shifts to the various centers.

Coming back to the scattered data problem, we find the coefficients c_k by solving the linear system

$$\begin{bmatrix} |x_1 - x_1| & |x_1 - x_2| & \dots & |x_1 - x_N| \\ |x_2 - x_1| & |x_2 - x_2| & \dots & |x_2 - x_N| \\ \vdots & \vdots & \ddots & \vdots \\ |x_N - x_1| & |x_N - x_2| & \dots & |x_N - x_N| \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} f_1(x_1) \\ f_1(x_2) \\ \vdots \\ f_1(x_N) \end{bmatrix}. \quad (1.3)$$

As mentioned earlier, for higher space dimensions s such a data dependent basis is required. Thus, even though the construction of piecewise linear splines in higher space dimensions is a different one (they are closely associated with an underlying computational mesh), the idea just presented suggests a very simple generalization of univariate piecewise linear splines that works for any space dimension.

The matrix in (1.3) above is an example of a *distance matrix*. Such matrices have been studied in geometry and analysis in the context of isometric embeddings of metric spaces for a long time (see, e.g., [Baxter (1991); Blumenthal (1938); Bochner (1941); Micchelli (1986); Schoenberg (1938a); Wells and Williams (1975)] and also Chapter 10). It is known that the distance matrix based on the Euclidean distance between a set of distinct points in \mathbb{R}^s is always non-singular (see Section 9.3 for more details). Therefore, we can solve the scattered data interpolation problem we posed on $[0, 1]^s$ by assuming

$$\mathcal{P}_f(\mathbf{x}) = \sum_{k=1}^N c_k \|\mathbf{x} - \mathbf{x}_k\|_2, \quad \mathbf{x} \in [0, 1]^s, \quad (1.4)$$

and then determine the coefficients c_k by solving the linear system

$$\begin{bmatrix} \|\mathbf{x}_1 - \mathbf{x}_1\|_2 & \|\mathbf{x}_1 - \mathbf{x}_2\|_2 & \dots & \|\mathbf{x}_1 - \mathbf{x}_N\|_2 \\ \|\mathbf{x}_2 - \mathbf{x}_1\|_2 & \|\mathbf{x}_2 - \mathbf{x}_2\|_2 & \dots & \|\mathbf{x}_2 - \mathbf{x}_N\|_2 \\ \vdots & \vdots & \ddots & \vdots \\ \|\mathbf{x}_N - \mathbf{x}_1\|_2 & \|\mathbf{x}_N - \mathbf{x}_2\|_2 & \dots & \|\mathbf{x}_N - \mathbf{x}_N\|_2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} f_s(\mathbf{x}_1) \\ f_s(\mathbf{x}_2) \\ \vdots \\ f_s(\mathbf{x}_N) \end{bmatrix}.$$

This is precisely the interpolation method we will choose to illustrate with our first MATLAB script *DistanceMatrixFit.m* (see Program 1.2 below) and the supporting figures and tables. A typical basis function for the Euclidean distance matrix fit, $B_k(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_k\|_2$, is shown in Figure 1.2 for the case $\mathbf{x}_k = \mathbf{0}$ and $s = 2$.

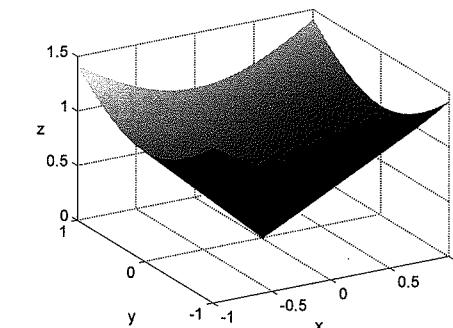


Fig. 1.2 A typical basis function for the Euclidean distance matrix centered at the origin in \mathbb{R}^2 .

Before we discuss the actual interpolation program we first list a subroutine used in many of our later examples. It is called *DistanceMatrix.m* and we use it to compute the matrix of pairwise Euclidean distances of two (possibly different) sets of points in \mathbb{R}^s . In the code these two sets are denoted by *dsites* and *ctrns*. In most of our examples both of these sets will coincide with the set \mathcal{X} of data sites.

Program 1.1. DistanceMatrix.m

```
% DM = DistanceMatrix(dsites,ctrs)
% Forms the distance matrix of two sets of points in R^s,
% i.e., DM(i,j) = || datasite_i - center_j ||_2.
% Input
%   dsites: Mxs matrix representing a set of M data sites in R^s
%           (i.e., each row contains one s-dimensional point)
%   ctrns: Nxs matrix representing a set of N centers in R^s
%           (one center per row)
% Output
%   DM:      MxN matrix whose i,j position contains the Euclidean
%           distance between the i-th data site and j-th center
```

```

1 function DM = DistanceMatrix(dsites,ctrs)
2 [M,s] = size(dsites); [N,s] = size(ctrs);
3 DM = zeros(M,N);
% Accumulate sum of squares of coordinate differences
% The ndgrid command produces two MxN matrices:
% dr, consisting of N identical columns (each containing
% the d-th coordinate of the M data sites)
% cc, consisting of M identical rows (each containing
% the d-th coordinate of the N centers)
4 for d=1:s
5     [dr,cc] = ndgrid(dsites(:,d),ctrs(:,d));
6     DM = DM + (dr-cc).^2;
7 end
8 DM = sqrt(DM);

```

Note that this subroutine can easily be modified to produce a p -norm distance matrix by making the obvious changes to lines 6 and 8 of the code in Program 1.1. We will come back to this idea in Chapter 10.

Our first main script is Program 1.2. This script can be used to compute the distance matrix interpolant to data sampled from the test function f_s provided by Program C.1. We use Halton points and are able to select the space dimension s and number of points N by editing lines 1 and 2 of the code. The subroutine `MakeSDGrid.m` which we use to compute the equally spaced points in the s -dimensional unit cube on line 6 of `DistanceMatrixFit.m` is provided in Appendix C. These equally spaced points are used as evaluation points and to compute errors. Note that since the distance matrix interpolant is of the form (1.4) its simultaneous evaluation at the entire set of evaluation points amounts to a matrix-vector product of the evaluation matrix \mathbf{EM} and the coefficients \mathbf{c} . Here the evaluation matrix has the same structure as the interpolation matrix and can also be computed using the subroutine `DistanceMatrix.m` (only using evaluation points in place of the data sites, see line 9 of `DistanceMatrixFit.m`). The coefficient vector \mathbf{c} is supplied directly as solution of the linear system $\mathbf{Ac} = \mathbf{f}$ (see (1.3) and the MATLAB expression $\mathbf{IM}\backslash\mathbf{rhs}$ on line 10 of the program). The evaluation points are subsequently used for the error computation in lines 11–13 and are also used for plotting purposes in the last part of the program (lines 16–35). Note that for this example we know the function f_s that generated the data, and therefore are able to compute the error in our reconstruction. The subroutines that produce the 2D and 3D plots on lines 24–32 are provided in Appendix C. Note that the use of `reshape` on lines 22–23 and 27–29 corresponds to the use of `meshgrid` for plotting purposes.

Program 1.2. `DistanceMatrixFit.m`

```
% DistanceMatrixFit
% Script that uses Euclidean distance matrices to perform
```

```

% scattered data interpolation for arbitrary space dimensions
% Calls on: DistanceMatrix, MakeSDGrid, testfunction
% Uses: haltonseq (written by Daniel Dougherty from MATLAB
% Central File Exchange)
1 s = 3;
2 k = 2; N = (2^k+1)^s;
3 neval = 10; M = neval^s;
% Use Halton points as data sites and centers
4 dsites = haltonseq(N,s);
5 ctrs = dsites;
% Create neval^s equally spaced evaluation locations in the
% s-dimensional unit cube
6 epoints = MakeSDGrid(s,neval);
% Create right-hand side vector,
% i.e., evaluate the test function at the data sites
7 rhs = testfunction(s,dsites);
% Compute distance matrix for the data sites and centers
8 IM = DistanceMatrix(dsites,ctrs);
% Compute distance matrix for evaluation points and centers
9 EM = DistanceMatrix(epoints,ctrs);
% Evaluate the interpolant on evaluation points
% (evaluation matrix * solution of interpolation system)
10 Pf = EM * (IM\rhs);
% Compute exact solution,
% i.e., evaluate test function on evaluation points
11 exact = testfunction(s,epoints);
% Compute maximum and RMS errors on evaluation grid
12 maxerr = norm(Pf-exact,inf);
13 rms_err = norm(Pf-exact)/sqrt(M);
14 fprintf('RMS error: %e\n', rms_err)
15 fprintf('Maximum error: %e\n', maxerr)
16 switch s
17 case 1
18 plot(epoints, Pf)
19 figure; plot(epoints, abs(Pf-exact))
20 case 2
21 fview = [-30,30];
22 xe = reshape(epoints(:,2),neval,neval);
23 ye = reshape(epoints(:,1),neval,neval);
24 PlotSurf(xe,ye,Pf,neval,exact,maxerr,fview);
25 PlotError2D(xe,ye,Pf,exact,maxerr,neval,fview);
26 case 3

```

```

27 xe = reshape(epoints(:,2),neval,neval,neval);
28 ye = reshape(epoints(:,1),neval,neval,neval);
29 ze = reshape(epoints(:,3),neval,neval,neval);
30 xslice = .25:.25:1; yslice = 1; zslice = [0,0.5];
31 PlotSlices(xe,ye,ze,Pf,neval,xslice,yslice,zslice);
32a PlotErrorSlices(xe,ye,ze,Pf,exact,neval,...  

32b     xslice,yslice,zslice);
33 otherwise
34     disp('Cannot display plots for s>3')
35 end

```

In Tables 1.1 and 1.2 as well as Figures 1.3 and 1.4 we present some examples computed with Program 1.2. The number M of evaluation points (determined by `neval` on line 3 of the code) we used for the cases $s = 1, 2, \dots, 6$, was 1000, 1600, 1000, 256, 1024, and 4096, respectively (*i.e.*, `neval` = 1000, 40, 10, 4, 4, and 4, respectively). Note that, as the space dimension s increases, more and more of the evaluation points lie on the boundary of the domain, while the data sites (which are given as Halton points) are located in the interior of the domain. The value k listed in Tables 1.1 and 1.2 is the same as the `k` in line 2 of Program 1.2. The formula for the root-mean-square error (RMS-error) is given by

$$\text{RMS-error} = \sqrt{\frac{1}{M} \sum_{j=1}^M [\mathcal{P}_f(\xi_j) - f(\xi_j)]^2} = \frac{1}{\sqrt{M}} \|\mathcal{P}_f - f\|_2, \quad (1.5)$$

where the ξ_j , $j = 1, \dots, M$ are the *evaluation points*. Formula (1.5) is used on line 13 of Program 1.2.

The basic MATLAB code for the solution of any kind of RBF interpolation problem will be very similar to Program 1.2. Note in particular that the data used — even for the distance matrix interpolation considered here — can also be “real” data. In that case one simply needs to replace lines 4 and 7 of the program by appropriate code that generates the data sites and data values for the right-hand side.

The plots on the left of Figures 1.3 and 1.4 display the graphs of the distance matrix fits for space dimensions $s = 1, 2$, and 3, respectively, while those on the right depict the corresponding errors. For the 1D plots (in Figure 1.3) we used 5 Halton points to interpolate the testfunction f_1 . The piecewise linear nature of the interpolant is clearly visible at this resolution. If we use more points then the fit becomes more accurate — see Table 1.1 — but then it is no longer possible to distinguish the piecewise linear nature of the interpolant. The 2D plot (top left of Figure 1.4) interpolates the testfunction f_2 at 289 Halton points. The graph of \mathcal{P}_f is false-colored according to the absolute error (indicated by the color bar at the right of the plot). The bottom plot in Figure 1.4 shows a slice plot of the distance matrix interpolant to f_3 based on 729 Halton points. For this plot the colors represent function values (again indicated by the color bar on the right).

Table 1.1 Distance matrix fit to N Halton points in $[0, 1]^s$, $s = 1, 2, 3$.

k	N	1D		2D		3D	
		RMS-error	N	RMS-error	N	RMS-error	
1	3	5.896957e-001	9	1.937341e-001	27	9.721476e-002	
2	5	3.638027e-001	25	6.336315e-002	125	6.277141e-002	
3	9	1.158328e-001	81	2.349093e-002	729	2.759452e-002	
4	17	3.981270e-002	289	1.045010e-002			
5	33	1.406188e-002	1089	4.326940e-003			
6	65	5.068541e-003	4225	1.797430e-003			
7	129	1.877013e-003					
8	257	7.264159e-004					
9	513	3.016376e-004					
10	1025	1.381896e-004					
11	2049	6.907386e-005					
12	4097	3.453179e-005					

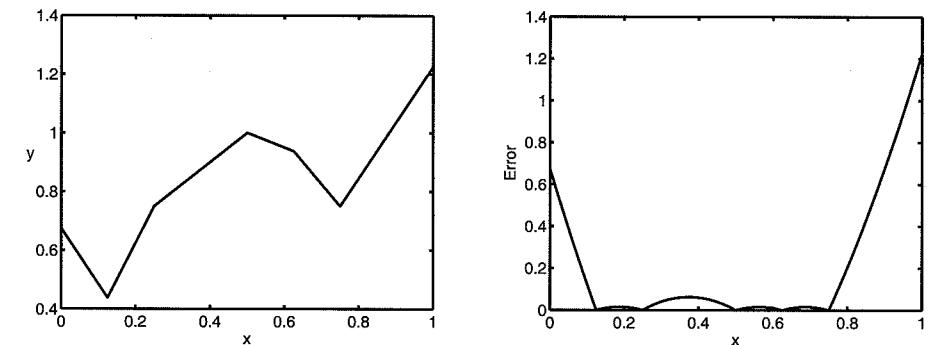


Fig. 1.3 Fit (left) and absolute error (right) for 5 point distance matrix interpolation in 1D.

In the right half of Figures 1.3 and 1.4 we show absolute errors for the distance matrix interpolants displayed in the left column. We use analogous color schemes, *i.e.*, the 2D plot (top part of Figure 1.4) is false-colored according to the absolute error, and so is the 3D plot (bottom) since now the “function value” corresponds to the absolute error. We can see clearly that most of the error is concentrated near the boundary of the domain. In fact, the absolute error is about one order of magnitude larger near the boundary than it is in the interior of the domain. This is no surprise since the data sites are located in the interior. However, even for uniformly spaced data sites (including points on the boundary) the main error in radial basis function interpolation is usually located near the boundary.

From this first simple example we can observe a number of other features. Most of them are characteristic for the radial basis function interpolants we will be studying later on. First, the basis functions employed, $B_k = \|\cdot - x_k\|_2$, are radially sym-

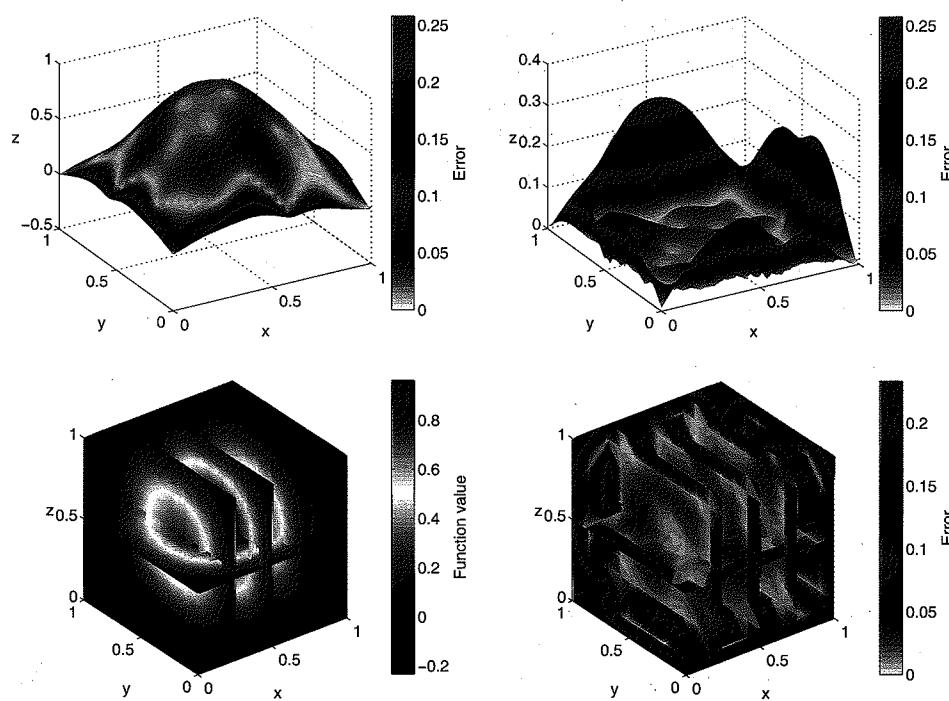


Fig. 1.4 Fits (left) and errors (right) for distance matrix interpolation with 289 points in 2D (top), and 729 points in 3D (bottom).

Table 1.2 Distance matrix fit to N Halton points in $[0, 1]^s$, $s = 4, 5, 6$.

k	4D		5D		6D	
	N	RMS-error	N	RMS-error	N	RMS-error
1	81	1.339581e-001	243	9.558350e-002	729	5.097600e-002
2	625	6.817424e-002	3125	3.118905e-002		

metric. Second, as the MATLAB scripts show, the method is extremely simple to implement for any space dimension s . For example, no underlying computational mesh is required to compute the interpolant. The process of mesh generation is a major factor when working in higher space dimensions with polynomial-based methods such as splines or finite elements. All that is required for our method is the pairwise distance between the data sites. Therefore, we have what is known as a *meshfree* (or *meshless*) method.

Third, the accuracy of the method improves if we add more data sites. In fact, it seems that the RMS-error in Tables 1.1 and 1.2 is reduced by a factor of about two from one row to the next. Since we use $(2^k + 1)^s$ uniformly distributed random

data points in row k this indicates a convergence rate of roughly $\mathcal{O}(h)$, where h can be viewed as something like the average distance or meshsize of the set \mathcal{X} of data sites (we will be more precise later on).

Another thing to note is that the simple distance function interpolant used here (as well as many other radial basis function interpolants used later) requires the solution of a system of linear equations with a dense $N \times N$ matrix. This makes it very costly to apply the method in its simple form to large data sets. Moreover, as we will see later, these matrices also tend to be rather ill-conditioned. These are the reasons why we can only present results for relatively small data sets in higher space dimensions using this simple approach.

In the remainder of this book it is our goal to present alternatives to this basic interpolation method that address the problems mentioned above such as limitation to small data sets, ill-conditioning, limited accuracy and limited smoothness of the interpolant.

1.2 Some Historical Remarks

Originally, the motivation for the basic meshfree approximation methods (radial basis function and moving least squares methods) came from applications in geodesy, geophysics, mapping, or meteorology. Later, applications were found in many other areas such as in the numerical solution of PDEs, computer graphics, artificial intelligence, statistical learning theory, neural networks, signal and image processing, sampling theory, statistics (kriging), finance, and optimization. It should be pointed out that meshfree local regression methods have been used independently in statistics for well over 100 years (see, e.g., [Cleveland and Loader (1996)] and the references therein). In fact, the basic moving least squares method (known also as local regression in the statistics literature) can be traced back at least to the work of [Gram (1883); Woolhouse (1870); De Forest (1873); De Forest (1874)].

In the literature on approximation theory and related applications areas some historical landmark contributions have come from

- Donald Shepard, who as an undergraduate student at Harvard University, suggested the use of what are now called *Shepard functions* in the late 1960s (see Chapter 22). The publication [Shepard (1968)] discusses the basic inverse distance weighted Shepard method and some modifications thereof. The method was at the time incorporated into a computer program, SYMAP, for map making.
- Rolland Hardy, who was a geodesist at Iowa State University. He introduced the so-called *multiquadratics* (MQs) in the early 1970s (see, e.g., [Hardy (1971)] or Chapter 8). Hardy's work was primarily concerned with applications in geodesy and mapping.

- Robert L. Harder and Robert N. Desmarais, who were aerospace engineers at MacNeal-Schwendler Corporation (MSC Software), and NASA's Langley Research Center. They introduced the so-called *thin plate splines* (TPSs) in 1972 (see, e.g., [Harder and Desmarais (1972)] or Chapter 8). Their work was concerned mostly with aircraft design.
- Jean Duchon, a mathematician at the Université Joseph Fourier in Grenoble, France. Duchon suggested a variational approach minimizing the integral of $\nabla^2 f$ in \mathbb{R}^2 which also leads to the thin plate splines. This work was done in the mid 1970s and is considered to be the foundation of the variational approach to radial basis functions (see [Duchon (1976); Duchon (1977); Duchon (1978); Duchon (1980)]) or Chapter 13).
- Jean Meinguet, a mathematician at Université Catholique de Louvain in Louvain, Belgium. Meinguet introduced what he called *surface splines* in the late 1970s. Surface splines and thin plate splines fall under what we will refer to as *polyharmonic splines* (see, e.g., [Meinguet (1979a); Meinguet (1979b); Meinguet (1979c); Meinguet (1984)] or Chapter 8).
- Peter Lancaster and Kes Šalkauskas, mathematicians at the University of Calgary, Canada. They published [Lancaster and Šalkauskas (1981)] introducing the *moving least squares method* (a generalization of Shepard functions).
- Richard Franke, a mathematician at the Naval Postgraduate School in Monterey, California. In [Franke (1982a)] he compared various scattered data interpolation methods, and concluded MQs and TPSs were the best. Franke also conjectured that the interpolation matrix for MQs is invertible.
- Wolodymyr (Wally) Madych, a mathematician at the University of Connecticut, and Stuart Alan Nelson, a mathematician from Iowa State University. In 1983 they completed their manuscript [Madych and Nelson (1983)] in which they proved Franke's conjecture (and much more) based on a variational approach. However, this manuscript was never published. Other fundamental papers by these two authors are, e.g., [Madych and Nelson (1988); Madych and Nelson (1990a); Madych and Nelson (1992)].
- Charles Micchelli, a mathematician at the IBM Watson Research Center. Micchelli published the paper [Micchelli (1986)]. He also proved Franke's conjecture. His proofs are rooted in the work of [Bochner (1932); Bochner (1933)] and [Schoenberg (1937); Schoenberg (1938a); Schoenberg (1938b)] on positive definite and completely monotone functions. This is also the approach we will follow throughout much of this book.
- Grace Wahba, a statistician at the University of Wisconsin. She studied the use of thin plate splines for statistical purposes in the context of smoothing *noisy data* and data on spheres, and introduced the ANOVA and cross validation approaches to the radial basis function setting (see, e.g., [Wahba

(1979); Wahba (1981); Wahba and Wendelberger (1980)]). One of the first monographs on the subject is [Wahba (1990b)].

- Robert Schaback, a mathematician at the University of Göttingen, Germany. *Compactly supported radial basis functions* (CSRBFs) were introduced in [Schaback (1995a)]; and a very popular family of CSRBFs was presented by Holger Wendland (also a mathematician in Göttingen) in his Ph.D. thesis (see also [Wendland (1995)] and Chapter 11). Both of these authors have contributed extensively to the field of radial basis functions. We mention particularly the recent monograph [Wendland (2005a)].

Chapter 2

Radial Basis Function Interpolation in MATLAB

Before we discuss any of the theoretical foundation of radial basis functions we want to get a feel for what they are all about. We saw in the introductory chapter that it is easy to use Euclidean distance matrices to compute a solution to the scattered data interpolation problem. However, we also pointed out a number of limitations to that approach such as the limited accuracy and limited smoothness. It turns out that we can maintain the underlying structure presented by the distance matrix approach and address these limitations by composing the distance function with certain “good” univariate functions.

2.1 Radial (Basis) Functions

As a first example we pick a function well-represented in many branches of mathematics, namely the *Gaussian*

$$\varphi(r) = e^{-(\varepsilon r)^2}, \quad r \in \mathbb{R}.$$

Our *shape parameter* ε is related to the variance σ^2 of the normal distribution function by $\varepsilon^2 = 1/(2\sigma^2)$. If we compose the Gaussian with the Euclidean distance function $\|\cdot\|_2$ we obtain for any fixed *center* $\mathbf{x}_k \in \mathbb{R}^s$ a *multivariate* function

$$\Phi_k(\mathbf{x}) = e^{-\varepsilon^2 \|\mathbf{x} - \mathbf{x}_k\|_2^2}, \quad \mathbf{x} \in \mathbb{R}^s.$$

Obviously, the connection between Φ_k and φ is given by

$$\Phi_k(\mathbf{x}) = \varphi(\|\mathbf{x} - \mathbf{x}_k\|_2).$$

It is this connection that gives rise to the name *radial basis function* (RBF). The following is a formal definition of a radial function.

Definition 2.1. A function $\Phi : \mathbb{R}^s \rightarrow \mathbb{R}$ is called *radial* provided there exists a *univariate* function $\varphi : [0, \infty) \rightarrow \mathbb{R}$ such that

$$\Phi(\mathbf{x}) = \varphi(r), \quad \text{where } r = \|\mathbf{x}\|,$$

and $\|\cdot\|$ is some norm on \mathbb{R}^s — usually the Euclidean norm.

Definition 2.1 says that for a radial function Φ

$$\|\mathbf{x}_1\| = \|\mathbf{x}_2\| \implies \Phi(\mathbf{x}_1) = \Phi(\mathbf{x}_2), \quad \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^s.$$

In other words, the value of Φ at any point at a certain fixed distance from the origin (or any other fixed center point) is constant. Thus, Φ is radially (or spherically) symmetric about its center. Definition 2.1 shows that the Euclidean distance function we used in the introduction is just a special case of a radial (basis) function. Namely, with $\varphi(r) = r$.

Figure 2.1 shows the graphs of two Gaussian radial basis functions, one with shape parameter $\varepsilon = 1$ (left) and one with $\varepsilon = 3$ (right) (both centered at the origin in \mathbb{R}^2). A smaller value of ε (*i.e.*, larger variance) causes the function to become “flatter”, while increasing ε leads to a more peaked RBF, and therefore localizes its influence. We will see soon that the choice of ε has a profound influence on both the accuracy and numerical stability of the solution to our interpolation problem.

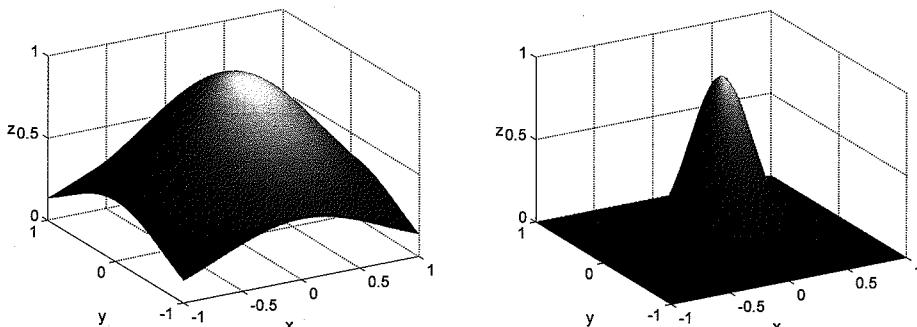


Fig. 2.1 Gaussian with $\varepsilon = 1$ (left) and $\varepsilon = 3$ (right) centered at the origin in \mathbb{R}^2 .

Definition 2.1 and the discussion leading up to it show again why it makes sense to call φ the *basic* function, and $\Phi_k(\|\cdot\|_2)$ (centered at \mathbf{x}_k) a radial *basis* function. One single basic function generates all of the basis functions that are used in the expansion (1.1).

Radial function interpolants have the nice property of being invariant under all Euclidean transformations (*i.e.*, translations, rotations, and reflections). By this we mean that it does not matter whether we first compute the RBF interpolant and then apply a Euclidean transformation, or if we first transform the data and then compute the interpolant. This is an immediate consequence of the fact that Euclidean transformations are characterized by orthogonal transformation matrices and are therefore 2-norm-invariant. Invariance under translation, rotation and reflection is often desirable in applications.

Moreover, the application of radial functions to the solution of the scattered data interpolation problem (as well as many other multivariate approximation problems) benefits from the fact that the interpolation problem becomes insensitive to the

dimension s of the space in which the data sites lie. Instead of having to deal with a multivariate function Φ (whose complexity will increase with increasing space dimension s) we can work with the same univariate function φ for all choices of s .

2.2 Radial Basis Function Interpolation

Instead of using simple distance matrices as we did earlier, we now use a radial basis function expansion to solve the scattered data interpolation problem in \mathbb{R}^s by assuming

$$\mathcal{P}_f(\mathbf{x}) = \sum_{k=1}^N c_k \varphi(\|\mathbf{x} - \mathbf{x}_k\|_2), \quad \mathbf{x} \in \mathbb{R}^s. \quad (2.1)$$

The coefficients c_k are found by enforcing the interpolation conditions, and thus solving the linear system

$$\begin{bmatrix} \varphi(\|\mathbf{x}_1 - \mathbf{x}_1\|_2) & \varphi(\|\mathbf{x}_1 - \mathbf{x}_2\|_2) & \dots & \varphi(\|\mathbf{x}_1 - \mathbf{x}_N\|_2) \\ \varphi(\|\mathbf{x}_2 - \mathbf{x}_1\|_2) & \varphi(\|\mathbf{x}_2 - \mathbf{x}_2\|_2) & \dots & \varphi(\|\mathbf{x}_2 - \mathbf{x}_N\|_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi(\|\mathbf{x}_N - \mathbf{x}_1\|_2) & \varphi(\|\mathbf{x}_N - \mathbf{x}_2\|_2) & \dots & \varphi(\|\mathbf{x}_N - \mathbf{x}_N\|_2) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} f(\mathbf{x}_1) \\ f(\mathbf{x}_2) \\ \vdots \\ f(\mathbf{x}_N) \end{bmatrix}.$$

As the solution of the scattered data interpolation problem hinges entirely on the solution of this system of linear equations we will devote the next chapter to the question of when (*i.e.*, for what type of basic functions φ) the system matrix is non-singular.

For the numerical example presented below we restrict ourselves to the two-dimensional case $s = 2$. As basic function φ we will use both Gaussians and the linear function $\varphi(r) = r$ which gives rise to the Euclidean distance matrix approach used earlier.

The code of the MATLAB script `RBFInterpolation2D.m` (see Program 2.1) we use to perform RBF interpolation in 2D is very similar to the earlier script `DistanceMatrixFit.m`. It also makes use of the subroutine `DistanceMatrix.m`. While it is easy to write a version of the interpolation script that works for any space dimension s (just as we did in `DistanceMatrixFit.m`) we will stick with a basic 2D version here.

In line 1 we define the Gaussian RBF as a MATLAB anonymous function that accepts a matrix argument (namely the output from `DistanceMatrix`) along with its shape parameter. Note that this feature is only available since MATLAB Release 7. For older MATLAB versions we suggest using an `inline` function instead (see the programs in the folder `Matlab6` of the enclosed CD). If execution speed is important, then one should explicitly provide the function (either hardcoded directly where needed, or as an M-file). This latter approach will always be more efficient than the `inline` or even anonymous function approach. However, then the interpolation program is no longer as generic.

We can replace the definition of the Gaussian on line 1 by the definition of the linear function $\varphi(r) = r$ or any other admissible RBF we will encounter later. In lines 2–6 we define a test function that we will sample similarly to the function f_s used in the introductory example. Here (and in many later examples) we use *Franke's function*

$$\begin{aligned} f(x, y) = & \frac{3}{4} e^{-1/4((9x-2)^2 + (9y-2)^2)} + \frac{3}{4} e^{-(1/49)(9x+1)^2 - (1/10)(9y+1)^2} \\ & + \frac{1}{2} e^{-1/4((9x-7)^2 + (9y-3)^2)} - \frac{1}{5} e^{-(9x-4)^2 - (9y-7)^2} \end{aligned} \quad (2.2)$$

which is a standard test function for 2D scattered data fitting. Note that we used (x, y) to denote the two components of $x \in \mathbb{R}^2$. The graph of Franke's function over the unit square is shown in Figure 2.2.

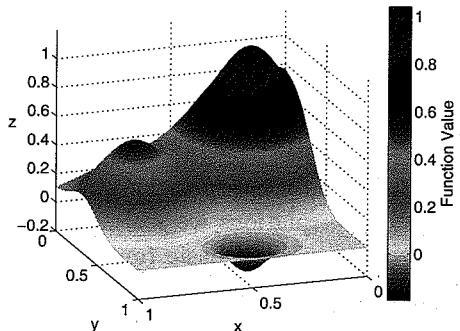


Fig. 2.2 Franke's test function.

For many of our examples we use data locations that have been saved in files named `Data2D_%d%s` where the number of points (`%d`) is taken from the progression $\{(2^k + 1)^2\} = \{9, 25, 81, 289, 1089, 4225, \dots\}$. The characters `u` or `h` (in place of `%s`) are used to denote either uniformly spaced points, or Halton points in the unit square. The set of data points is defined and loaded in lines 7 and 8. As in the earlier example we consider here only the case where the centers for the RBFs coincide with the data locations (line 9).

A grid of evaluation points used to evaluate our interpolant for the purposes of rendering and error computation is defined in lines 10 and 11. The test data (right-hand side of the interpolation equations) are computed on line 12 where the test function is sampled at the data sites.

The main part of the code is given by lines 13–17. Note that this part is very similar to the corresponding segment (lines 7–9) in `DistanceMatrixFit.m`. The only difference is that we now apply the basic function φ to the entire distance matrices in order to obtain the interpolation and evaluation matrices.

Program 2.1. RBFInterpolation2D.m

```
% RBFInterpolation2D
% Script that performs basic 2D RBF interpolation
% Calls on: DistanceMatrix
%
% Define the Gaussian RBF and shape parameter
1 rbf = @(e,r) exp(-(e*r).^2); ep = 21.1;
% Define Franke's function as testfunction
2 f1 = @(x,y) 0.75*exp(-((9*x-2).^2+(9*y-2).^2)/4);
3 f2 = @(x,y) 0.75*exp(-((9*x+1).^2/49+(9*y+1).^2/10));
4 f3 = @(x,y) 0.5*exp(-((9*x-7).^2+(9*y-3).^2)/4);
5 f4 = @(x,y) 0.2*exp(-((9*x-4).^2+(9*y-7).^2));
6 testfunction = @(x,y) f1(x,y)+f2(x,y)+f3(x,y)-f4(x,y);
7 N = 1089; gridtype = 'h';
% Load data points
8 name = sprintf('Data2D_%d%s',N,gridtype); load(name)
9 ctrs = dsites;
10 neval = 40; grid = linspace(0,1,neval);
11 [xe,ye] = meshgrid(grid); epoints = [xe(:) ye(:)];
% Evaluate the test function at the data points
12 rhs = testfunction(dsites(:,1),dsites(:,2));
% Compute distance matrix between the data sites and centers
13 DM_data = DistanceMatrix(dsites,ctrs);
% Compute interpolation matrix
14 IM = rbf(ep,DM_data);
% Compute distance matrix between evaluation points and centers
15 DM_eval = DistanceMatrix(epoints,ctrs);
% Compute evaluation matrix
16 EM = rbf(ep,DM_eval);
% Compute RBF interpolant
% (evaluation matrix * solution of interpolation system)
17 Pf = EM * (IM\rhs);
% Compute exact solution, i.e.,
% evaluate test function on evaluation points
18 exact = testfunction(epoints(:,1),epoints(:,2));
% Compute errors on evaluation grid
19 maxerr = norm(Pf-exact,inf);
20 rms_err = norm(Pf-exact)/neval;
21 fprintf('RMS error: %e\n', rms_err)
22 fprintf('Maximum error: %e\n', maxerr)
23 fview = [160,20]; % for Franke's function
24 PlotSurf(xe,ye,Pf,neval,exact,maxerr,fview);
25 PlotError2D(xe,ye,Pf,exact,maxerr,neval,fview);
```

In Table 2.1 we report the results of a series of experiments in which we compute Gaussian RBF and distance matrix interpolants to increasingly larger sets of data. We use one fixed value of ε for all of the experiments with the Gaussians. This type of approximation is known as *non-stationary* approximation. Its counterpart is known as *stationary* approximation.

Even though we do not perform stationary interpolation in this experiment we take a minute to explain the essential difference between the two approaches. In the stationary setting we would scale the shape parameter ε according to the *fill distance* (or meshsize) h so that we end up using “peaked” basis functions for densely spaced data and “flat” basis functions for coarsely spaced data. We will use the fill distance as a measure of the data distribution. The fill distance is usually defined as

$$h = h_{\mathcal{X}, \Omega} = \sup_{x \in \Omega} \min_{x_j \in \mathcal{X}} \|x - x_j\|_2, \quad (2.3)$$

and it indicates how well the data in the set \mathcal{X} fill out the domain Ω . A geometric interpretation of the fill distance is given by the radius of the largest possible empty ball that can be placed among the data locations inside Ω (see Figure 2.3). Sometimes the synonym *covering radius* is used. In our MATLAB code we can estimate the fill distance via

$$h_X = \max(\min(DM_eval')) \quad (2.4)$$

where DM_eval is the matrix consisting of pairwise distances between the evaluation points (placed on a fine uniform grid in Ω) and the data sites \mathcal{X} (c.f. line 15 of Program 2.1). Note that we transpose the non-symmetric evaluation matrix. This corresponds to finding — for each evaluation point — the distance to the corresponding closest data site, and then setting $h_{\mathcal{X}, \Omega}$ as the worst of those distances. Figure 2.3 illustrates the fill distance for a set of 25 Halton points. Note that in this case the largest “hole” in the data is near the boundary.

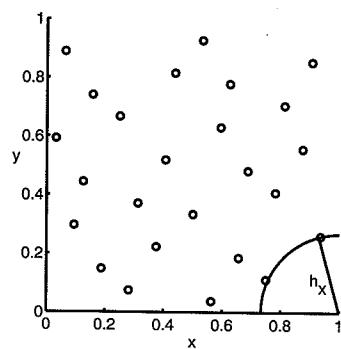


Fig. 2.3 The fill distance for $N = 25$ Halton points ($h_{\mathcal{X}, \Omega} \approx 0.2667$).

We will take a closer look at the differences between stationary and non-stationary interpolation in later chapters of this book.

In the following examples we will clearly see the effects the shape parameter has on the condition number of the interpolation matrix (and therefore the numerical stability) of our computations. In order to be able to use our script `RBFInterpolation2D.m` in conjunction with Gaussians to produce a meaningful sequence of non-stationary experiments, i.e., with a fixed value of the shape parameter ε , we are required to take the fairly large value $\varepsilon = 21.1$. Otherwise computation with the relatively densely spaced point set of $N = 4225$ Halton points results in MATLAB warnings of ill-conditioning. This means that — for the non-stationary approach — the basis functions are too localized on the smaller point sets, and the approximation is very poor (see Figure 2.4).

The test results for a non-stationary interpolation experiment using Gaussians and Euclidean distance matrices for Franke’s function are shown in Table 2.1. As just pointed out, we note that a fit with Gaussians and a small shape parameter such as $\varepsilon = 1$ would quickly lead to a numerical breakdown. For as few as $N = 25$ data points and $\varepsilon = 1$ MATLAB issues a “matrix close to singular” warning with an estimated reciprocal condition number of $RCOND=3.986027e-020$.

Table 2.1 Non-stationary RBF interpolation to Franke’s function using Gaussians ($\varepsilon = 21.1$) and Euclidean distance matrices.

k	N	Gaussian		distance matrix	
		RMS-error	max-error	RMS-error	max-error
1	9	3.647169e-001	1.039682e+000	1.323106e-001	4.578028e-001
2	25	3.203404e-001	9.670980e-001	6.400558e-002	2.767871e-001
3	81	2.152222e-001	8.455161e-001	1.343780e-002	6.733130e-002
4	289	7.431729e-002	7.219253e-001	3.707360e-003	3.057540e-002
5	1089	1.398297e-002	3.857234e-001	1.143589e-003	1.451950e-002
6	4225	4.890709e-004	1.940675e-002	4.002749e-004	8.022336e-003

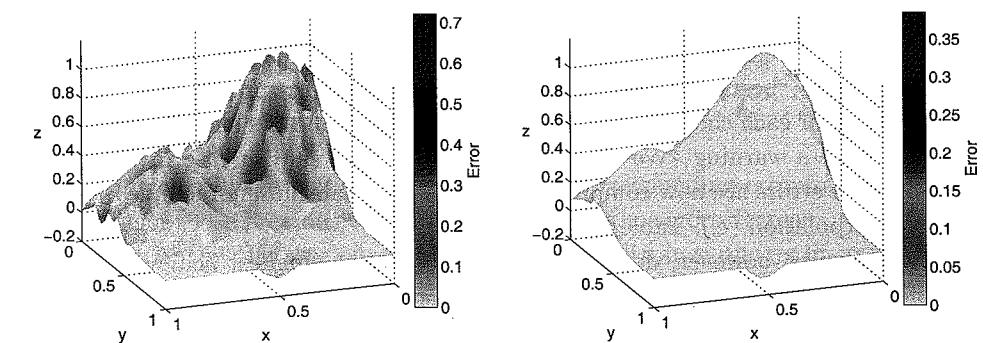


Fig. 2.4 Gaussian RBF interpolant with $\varepsilon = 21.1$ at $N = 289$ (left) and at $N = 1089$ Halton points (right).

If we look at the entries in Table 2.1 then we see that — contrary to what we announced earlier — the results based on the distance matrix fit are more accurate than those obtained with Gaussians. This will change, however, if we try to optimize our choice of the shape parameter ε (see the results of the next experiment in Table 2.2).

In a second experiment we consider the same test function and Gaussian basis functions. Now, however, we want to study the effects of the shape parameter. Therefore, in Figure 2.5 we display both the maximum and RMS errors as a function of the shape parameter ε for four fixed data sets (81, 289, 1089 and 4225 Halton points). These curves reveal some of the problems associated with radial basis function interpolation — especially when working with globally supported basis functions, *i.e.*, dense matrices. We see that the errors decrease with decreasing ε (of course, they also decrease with decreasing fill distance — but that is not what we are concerned with now). However, the error curves are not monotonic. We can identify an optimal value of ε for which both errors are minimal (the minima of the two error curves occur at almost the same place). Moreover, there is a value of ε at which the computational results become unpredictable, and the error curves become erratic. This point is associated with severe ill-conditioning of the system matrix. Since MATLAB issues a warning when attempting to solve an ill-conditioned linear system, we refer to the smallest value of ε for which we do not see a MATLAB warning as the “safe” value of ε (for a given set \mathcal{X} of data sites and basic function φ). The interesting fact about the four plots displayed in Figure 2.5 is that for the smaller data sets ($N = 81$ and $N = 289$) the minimum errors are obtained for a “safe” ε , while for the larger sets ($N = 1089$ and $N = 4225$) the minimum errors are obtained in the “unsafe” range. Therefore, we are computing in a certain “gray zone”. We are obtaining highly accurate solutions from severely ill-conditioned linear systems. We will come back later to this interesting feature of radial basis function interpolation (called *uncertainty* or *trade-off principle*). It is conceivable (and in fact possible [Fornberg and Wright (2004)]) to obtain even more accurate results by using a more stable way to evaluate the radial basis function interpolant (see the discussion in Chapters 16 and 17).

In Table 2.2 we list the “best possible” results for stationary Gaussian interpolation. We view “best” in two different ways. For the results presented in columns 3–5 we select for each choice of N the smallest possible value of ε such that MATLAB does not issue a warning. We refer to this case as the “safe” ε case in Table 2.2. Most of these errors are now comparable (or smaller) than those for simple distance matrix interpolation (*c.f.* Table 2.1).

These results, however, do not always represent the smallest achievable error. Therefore, we present (in columns 6–8) results for those “optimal” values of ε which yield the smallest RMS-error. These results are obtained in the “gray zone” mentioned above. For example, if we use $N = 1089$ Halton points and shape parameter $\varepsilon = 6.2$ then MATLAB issues a warning with $\text{RCOND} = 2.683527e-020$. However,

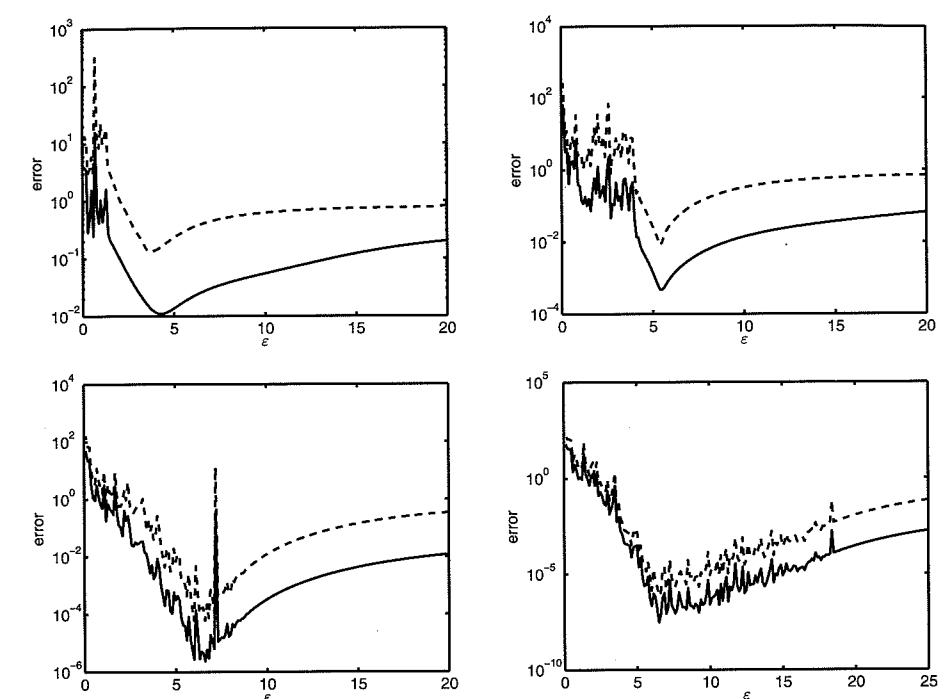


Fig. 2.5 Maximum (dashed/top curve) and RMS (solid/bottom curve) errors vs. ε for 81 (top left), 289 (top right), 1089 (bottom left), and 4225 Halton points (bottom right).

Table 2.2 “Optimal” RBF interpolation to Franke’s function using Gaussians.

k	N	ε	smallest “safe” ε		smallest RMS-error	
			RMS-error	max-error	ε	RMS-error
1	9	0.02	3.658421e-001	1.580259e+000	2.23	1.118026e-001
2	25	0.32	3.629342e-001	2.845554e+000	3.64	4.032550e-002
3	81	1.64	1.743059e-001	2.398284e+000	4.28	1.090601e-002
4	289	4.73	2.785388e-003	5.472502e-002	5.46	4.610079e-004
5	1089	10.5	4.945428e-004	1.812246e-002	6.2	2.498848e-006
6	4225	21.1	4.890709e-004	1.940675e-002	6.3	4.269292e-008

as we can see in Table 2.2 and Figure 2.5, the corresponding errors are now much smaller than those previously obtained. Moreover, the errors decrease at a rate that is faster than the $\mathcal{O}(h)$ we observed earlier for the distance matrix fit example.

Of course, if the data we are trying to fit are not sampled from a known test function then we will not be able to choose an “optimal” shape parameter by monitoring the RMS error. The associated issues of ill-conditioning, preconditioning, optimal shape parameter selection, and alternate stable evaluation methods via a Contour-Padé algorithm are studied later in Chapters 16 and 17.

Chapter 3

Positive Definite Functions

We noted in the previous chapters that the solution of the scattered data interpolation problem with RBFs boils down to the solution of a system of linear equations

$$A\mathbf{c} = \mathbf{y},$$

where the system matrix A has entries $\varphi(\|\mathbf{x}_j - \mathbf{x}_k\|_2)$, $j, k = 1, \dots, N$. We know from linear algebra that this system will have a unique solution whenever the matrix A is non-singular. While no one has yet succeeded in characterizing the class of all basic functions φ that generate a non-singular system matrix for any set $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ of distinct data sites, the situation is much better if we consider *positive definite matrices*.

In this chapter we present the main theoretical results underlying this approach along with some of their proofs. A series of examples are presented in the next chapter. A comprehensive treatment of the mathematical theory needed for scattered data interpolation with strictly positive definite functions (see Def. 3.2 below) is presented in the recent monograph [Wendland (2005a)].

3.1 Positive Definite Matrices and Functions

Definition 3.1. A real symmetric matrix A is called *positive semi-definite* if its associated quadratic form is non-negative, *i.e.*,

$$\sum_{j=1}^N \sum_{k=1}^N c_j c_k A_{jk} \geq 0 \quad (3.1)$$

for $\mathbf{c} = [c_1, \dots, c_N]^T \in \mathbb{R}^N$.

If the quadratic form (3.1) is zero only for $\mathbf{c} \equiv \mathbf{0}$, then A is called *positive definite*.

An important property of positive definite matrices is that all their eigenvalues are positive, and therefore a positive definite matrix is non-singular (but certainly not vice versa).

If we therefore had basis functions B_k in the expansion (1.1) that generate a positive definite interpolation matrix, we would always have a well-posed interpolation problem. To this end we introduce the concept of a *positive definite function* from classical analysis.

Positive definite functions were first considered in classical analysis early in the 20th century. [Mathias (1923)] seems to have been the first to define and study positive definite functions. An overview of the development of positive definite functions up to the mid 1970s can be found in [Stewart (1976)]. However, as we see from the definition below, positive definite functions were — unfortunately — defined in analogy to positive semi-definite matrices. Therefore, in order to meet our goal of having a well-posed interpolation problem, it is necessary to sharpen the classical notion of a positive definite function to that of a *strictly* positive definite one. This concept does not seem to have been studied until [Micchelli (1986)] made the connection between scattered data interpolation and positive definite functions. This leads to an unfortunate difference in terminology used in the context of matrices and functions. Instead of rewriting history we will adhere to this terminology here. We would like to point out that when reading recent articles (especially in the radial basis function literature) dealing with (strictly) positive definite functions one has to be aware of the fact that some authors have tried to “correct” history, and now refer to strictly positive definite functions as positive definite functions.

Definition 3.2. A complex-valued continuous function $\Phi : \mathbb{R}^s \rightarrow \mathbb{C}$ is called *positive definite on \mathbb{R}^s* if

$$\sum_{j=1}^N \sum_{k=1}^N c_j \overline{c_k} \Phi(\mathbf{x}_j - \mathbf{x}_k) \geq 0 \quad (3.2)$$

for any N pairwise different points $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^s$, and $\mathbf{c} = [c_1, \dots, c_N]^T \in \mathbb{C}^N$.

The function Φ is called *strictly positive definite on \mathbb{R}^s* if the quadratic form (3.2) is zero only for $\mathbf{c} \equiv 0$.

We note that even though we are interested in problems with real data and real coefficients, an extension of the notion of positive definiteness to cover complex coefficients \mathbf{c} and complex-valued functions Φ as done in Definition 3.2 will be helpful when deriving some properties of (strictly) positive definite functions later on. Moreover, the celebrated *Bochner's theorem* (see Theorem 3.3) characterizes exactly the positive definite functions of Definition 3.2. In all practical circumstances, however, we will be concerned with real-valued functions only, and a characterization of such functions appears below as Theorem 3.2. It should also be noted that Definition 3.2 implies that only functions whose quadratic form is real are candidates for (strictly) positive definite functions.

Example 3.1. Here, and throughout this book, we will denote the standard inner product of \mathbf{x} and \mathbf{y} in \mathbb{R}^s by $\mathbf{x} \cdot \mathbf{y}$. With this notation the function $\Phi(\mathbf{x}) = e^{i\mathbf{x} \cdot \mathbf{y}}$,

for $\mathbf{y} \in \mathbb{R}^s$ fixed, is positive definite on \mathbb{R}^s since the quadratic form in Definition 3.2 becomes

$$\begin{aligned} \sum_{j=1}^N \sum_{k=1}^N c_j \overline{c_k} \Phi(\mathbf{x}_j - \mathbf{x}_k) &= \sum_{j=1}^N \sum_{k=1}^N c_j \overline{c_k} e^{i(\mathbf{x}_j - \mathbf{x}_k) \cdot \mathbf{y}} \\ &= \sum_{j=1}^N c_j e^{i\mathbf{x}_j \cdot \mathbf{y}} \sum_{k=1}^N \overline{c_k} e^{-i\mathbf{x}_k \cdot \mathbf{y}} \\ &= \left| \sum_{j=1}^N c_j e^{i\mathbf{x}_j \cdot \mathbf{y}} \right|^2 \geq 0. \end{aligned}$$

Definition 3.2 and the discussion preceding it suggest that we should use strictly positive definite functions as basis functions in (1.1), i.e., $B_k(\mathbf{x}) = \Phi(\mathbf{x} - \mathbf{x}_k)$, or

$$\mathcal{P}_f(\mathbf{x}) = \sum_{k=1}^N c_k \Phi(\mathbf{x} - \mathbf{x}_k), \quad \mathbf{x} \in \mathbb{R}^s. \quad (3.3)$$

Note that at this point we do not require Φ to be a radial function. In fact, the function \mathcal{P}_f of (3.3) will yield an interpolant that is *translation invariant*, i.e., the interpolant to translated data is the same as the translated interpolant to the original data. In order to obtain invariance also under rotations and reflections we will later specialize to strictly positive definite functions that are also radial on \mathbb{R}^s .

We will now discuss some of the most important properties and characterizations of (strictly) positive definite functions. For the sake of completeness we present a list of some basic properties of (strictly) positive definite functions and some examples.

Theorem 3.1. *Some basic properties of positive definite functions are*

- (1) *Non-negative finite linear combinations of positive definite functions are positive definite. If Φ_1, \dots, Φ_n are positive definite on \mathbb{R}^s and $c_j \geq 0$, $j = 1, \dots, n$, then*

$$\Phi(\mathbf{x}) = \sum_{j=1}^n c_j \Phi_j(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^s,$$

is also positive definite. Moreover, if at least one of the Φ_j is strictly positive definite and the corresponding $c_j > 0$, then Φ is strictly positive definite.

- (2) $\Phi(\mathbf{0}) \geq 0$.
- (3) $\Phi(-\mathbf{x}) = \overline{\Phi(\mathbf{x})}$.
- (4) *Any positive definite function is bounded. In fact,*

$$|\Phi(\mathbf{x})| \leq \Phi(\mathbf{0}).$$

- (5) *If Φ is positive definite with $\Phi(\mathbf{0}) = 0$ then $\Phi \equiv 0$.*
- (6) *The product of (strictly) positive definite functions is (strictly) positive definite.*

Proof. Properties (1) and (2) follow immediately from Definition 3.2.

To show (3) we let $N = 2$, $\mathbf{x}_1 = \mathbf{0}$, $\mathbf{x}_2 = \mathbf{x}$, and choose $c_1 = 1$ and $c_2 = c$. Then the quadratic form in Definition 3.2 becomes

$$\sum_{j=1}^2 \sum_{k=1}^2 c_j \overline{c_k} \Phi(\mathbf{x}_j - \mathbf{x}_k) = (1 + |c|^2) \Phi(\mathbf{0}) + c \Phi(\mathbf{x}) + \overline{c} \Phi(-\mathbf{x}) \geq 0$$

for every $c \in \mathbb{C}$. Taking $c = 1$ and $c = i$ (where $i = \sqrt{-1}$), respectively, we can see that both $\Phi(\mathbf{x}) + \Phi(-\mathbf{x})$ and $i(\Phi(\mathbf{x}) - \Phi(-\mathbf{x}))$ must be real. This, however, is only possible if $\Phi(-\mathbf{x}) = \overline{\Phi(\mathbf{x})}$.

For the proof of (4) we let $N = 2$, $\mathbf{x}_1 = \mathbf{0}$, $\mathbf{x}_2 = \mathbf{x}$, and choose $c_1 = |\Phi(\mathbf{x})|$ and $c_2 = -\overline{\Phi(\mathbf{x})}$. Then the quadratic form in Definition 3.2 is

$$\sum_{j=1}^2 \sum_{k=1}^2 c_j \overline{c_k} \Phi(\mathbf{x}_j - \mathbf{x}_k) = 2\Phi(\mathbf{0})|\Phi(\mathbf{x})|^2 - \Phi(-\mathbf{x})\Phi(\mathbf{x})|\Phi(\mathbf{x})| - \Phi^2(\mathbf{x})|\Phi(\mathbf{x})| \geq 0.$$

Since $\Phi(-\mathbf{x}) = \overline{\Phi(\mathbf{x})}$ by Property (3), this gives

$$2\Phi(\mathbf{0})|\Phi(\mathbf{x})|^2 - 2|\Phi(\mathbf{x})|^3 \geq 0.$$

If $|\Phi(\mathbf{x})| > 0$, we divide by $|\Phi(\mathbf{x})|^2$ and the statement follows immediately. In case $|\Phi(\mathbf{x})| \equiv 0$ the statement holds trivially.

Property (5) follows immediately from (4), and Property (6) is a consequence of a theorem by Schur in the field of linear algebra which states that the elementwise (or Hadamard) product of positive (semi-)definite matrices is positive (semi-)definite. For more details we refer the reader to [Cheney and Light (1999)] or [Wendland (2005a)]. \square

Example 3.2. The cosine function is positive definite on \mathbb{R} since, for $x \in \mathbb{R}$, we have $\cos x = \frac{1}{2}(e^{ix} + e^{-ix})$. Now Property (1) and Example 3.1 can be invoked.

Property (3) shows that any real-valued (strictly) positive definite function has to be even. However, it is also possible to characterize real-valued (strictly) positive definite functions using only *real* coefficients (see [Wendland (2005a)] for details), *i.e.*,

Theorem 3.2. A real-valued continuous function Φ is positive definite on \mathbb{R}^s if and only if it is even and

$$\sum_{j=1}^N \sum_{k=1}^N c_j c_k \Phi(\mathbf{x}_j - \mathbf{x}_k) \geq 0 \quad (3.4)$$

for any N pairwise different points $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^s$, and $\mathbf{c} = [c_1, \dots, c_N]^T \in \mathbb{R}^N$.

The function Φ is strictly positive definite on \mathbb{R}^s if the quadratic form (3.4) is zero only for $\mathbf{c} \equiv \mathbf{0}$.

3.2 Integral Characterizations for (Strictly) Positive Definite Functions

We will now summarize some facts about integral characterizations of positive definite functions. They were established in the 1930s by Bochner and Schoenberg. However, we will also mention the more recent extensions to strictly positive definite and strictly completely/multiply monotone functions that are essential to the application of the theory to the scattered data interpolation problem. A much more detailed discussion of this material is presented in the recent book [Wendland (2005a)]. Some frequently used integral transforms are listed in Appendix B. Integral characterizations of the closely related completely and multiply monotone functions are presented in Chapter 5.

3.2.1 Bochner's Theorem

One of the most celebrated results on positive definite functions is their characterization in terms of Fourier transforms established by Bochner in 1932 (for $s = 1$) and 1933 (for general s).

Theorem 3.3 (Bochner). A (complex-valued) function $\Phi \in C(\mathbb{R}^s)$ is positive definite on \mathbb{R}^s if and only if it is the Fourier transform of a finite non-negative Borel measure μ on \mathbb{R}^s , i.e.

$$\Phi(\mathbf{x}) = \hat{\mu}(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^s}} \int_{\mathbb{R}^s} e^{-i\mathbf{x} \cdot \mathbf{y}} d\mu(\mathbf{y}), \quad \mathbf{x} \in \mathbb{R}^s.$$

Proof. There are many proofs of this theorem. Bochner's original proof can be found in [Bochner (1933)]. Other proofs can be found, *e.g.*, in the books [Cuppens (1975)] or [Gel'fand and Vilenkin (1964)]. A proof using the Riesz representation theorem to interpret the Borel measure as a distribution, and then taking advantage of distributional Fourier transforms can be found in the book [Wendland (2005a)].

We will prove only the one (easy) direction. It is this part of the statement that is important for the application to scattered data interpolation. We assume Φ is the Fourier transform of a finite non-negative Borel measure and show Φ is positive definite. Thus,

$$\begin{aligned} \sum_{j=1}^N \sum_{k=1}^N c_j \overline{c_k} \Phi(\mathbf{x}_j - \mathbf{x}_k) &= \frac{1}{\sqrt{(2\pi)^s}} \sum_{j=1}^N \sum_{k=1}^N \left[c_j \overline{c_k} \int_{\mathbb{R}^s} e^{-i(\mathbf{x}_j - \mathbf{x}_k) \cdot \mathbf{y}} d\mu(\mathbf{y}) \right] \\ &= \frac{1}{\sqrt{(2\pi)^s}} \int_{\mathbb{R}^s} \left[\sum_{j=1}^N c_j e^{-i\mathbf{x}_j \cdot \mathbf{y}} \sum_{k=1}^N \overline{c_k} e^{i\mathbf{x}_k \cdot \mathbf{y}} \right] d\mu(\mathbf{y}) \\ &= \frac{1}{\sqrt{(2\pi)^s}} \int_{\mathbb{R}^s} \left| \sum_{j=1}^N c_j e^{-i\mathbf{x}_j \cdot \mathbf{y}} \right|^2 d\mu(\mathbf{y}) \geq 0. \end{aligned}$$

The last inequality holds because of the conditions imposed on the measure μ . \square

Remark 3.1. We can see from Theorem 3.3 that the function $\Phi(\mathbf{x}) = e^{i\mathbf{x} \cdot \mathbf{y}}$ of Example 3.1 can be considered as the *fundamental positive definite function* since all other positive definite functions are obtained as (infinite) linear combinations of this function. While Property (1) of Theorem 3.1 implies that linear combinations of Φ will again be positive definite, the remarkable content of Bochner's Theorem is the fact that indeed *all* positive definite functions are generated by Φ .

3.2.2 Extensions to Strictly Positive Definite Functions

In order to accomplish our goal of guaranteeing a well-posed interpolation problem we have to extend (if possible) Bochner's characterization to *strictly* positive definite functions.

We begin with a sufficient condition for a function to be strictly positive definite on \mathbb{R}^s .

For this result we require the notion of the *carrier* of a (non-negative) Borel measure defined on some topological space X (see also Appendix B). This set is given by

$$X \setminus \bigcup \{O : O \text{ is open and } \mu(O) = 0\}.$$

Theorem 3.4. Let μ be a non-negative finite Borel measure on \mathbb{R}^s whose carrier is a set of nonzero Lebesgue measure. Then the Fourier transform of μ is strictly positive definite on \mathbb{R}^s .

Proof. As in the proof of Bochner's theorem we have

$$\begin{aligned} \sum_{j=1}^N \sum_{k=1}^N c_j \bar{c}_k \hat{\mu}(\mathbf{x}_j - \mathbf{x}_k) &= \frac{1}{\sqrt{(2\pi)^s}} \sum_{j=1}^N \sum_{k=1}^N c_j \bar{c}_k \left[\int_{\mathbb{R}^s} e^{-i(\mathbf{x}_j - \mathbf{x}_k) \cdot \mathbf{y}} d\mu(\mathbf{y}) \right] \\ &= \frac{1}{\sqrt{(2\pi)^s}} \int_{\mathbb{R}^s} \left[\sum_{j=1}^N c_j e^{-i\mathbf{x}_j \cdot \mathbf{y}} \sum_{k=1}^N \bar{c}_k e^{i\mathbf{x}_k \cdot \mathbf{y}} \right] d\mu(\mathbf{y}) \\ &= \frac{1}{\sqrt{(2\pi)^s}} \int_{\mathbb{R}^s} \left| \sum_{j=1}^N c_j e^{-i\mathbf{x}_j \cdot \mathbf{y}} \right|^2 d\mu(\mathbf{y}) \geq 0. \end{aligned}$$

Now let

$$g(\mathbf{y}) = \sum_{j=1}^N c_j e^{-i\mathbf{x}_j \cdot \mathbf{y}},$$

and assume that the points \mathbf{x}_j are all distinct and $c \neq 0$. In this case the functions $\mathbf{y} \mapsto e^{-i\mathbf{x}_j \cdot \mathbf{y}}$ are linearly independent so that $g \neq 0$. Since g is an entire function its zero set, i.e., $\{\mathbf{y} \in \mathbb{R}^s : g(\mathbf{y}) = 0\}$ can have no accumulation point and therefore it has Lebesgue measure zero (see, e.g., [Cheney and Light (1999)]). Now, the only

remaining way to make the above inequality an equality is if the carrier of μ is contained in the zero set of g , i.e., has Lebesgue measure zero. This, however, is ruled out in the hypothesis of the theorem. \square

Work toward an analog of Bochner's theorem, i.e., a complete integral characterization of functions that are strictly positive definite on \mathbb{R}^s , is given in [Chang (1996)] for the case $s = 1$.

The following corollary gives us a way to *construct* strictly positive definite functions.

Corollary 3.1. Let f be a continuous non-negative function in $L_1(\mathbb{R}^s)$ which is not identically zero. Then the Fourier transform of f is strictly positive definite on \mathbb{R}^s .

Proof. This is a special case of the previous theorem in which the measure μ has Lebesgue density f . Thus, we use the measure μ defined for any Borel set B by

$$\mu(B) = \int_B f(\mathbf{x}) dx.$$

Then the carrier of μ is equal to the (closed) support of f . However, since f is non-negative and not identically equal to zero, its support has positive Lebesgue measure, and hence the Fourier transform of f is strictly positive definite by the preceding theorem. \square

Finally, a criterion to check whether a given function is strictly positive definite is given in [Wendland (2005a)].

Theorem 3.5. Let Φ be a continuous function in $L_1(\mathbb{R}^s)$. Φ is strictly positive definite if and only if Φ is bounded and its Fourier transform is non-negative and not identically equal to zero.

Theorem 3.5 is of fundamental importance and we will come back to this theorem several times later on. In fact, the proof of Theorem 3.5 in [Wendland (2005a)] shows that — if $\Phi \not\equiv 0$ (which implies that then also $\hat{\Phi} \not\equiv 0$) — we need to ensure only that $\hat{\Phi}$ be non-negative in order for Φ to be strictly positive definite.

3.3 Positive Definite Radial Functions

We now turn our attention to positive definite *radial* functions. Recall that Definition 3.2 characterizes (strictly) positive definite functions in terms of multivariate functions Φ . However, when we are dealing with radial functions, i.e., $\Phi(\mathbf{x}) = \varphi(\|\mathbf{x}\|)$, then it will be convenient to also refer to the univariate function φ as a *positive definite radial function*. While this does present a slight abuse of our terminology for positive definite functions this is what is commonly done in the literature.

An immediate consequence of this notational convention is

Lemma 3.1. If $\Phi = \varphi(\|\cdot\|)$ is (strictly) positive definite and radial on \mathbb{R}^s then Φ is also (strictly) positive definite and radial on \mathbb{R}^σ for any $\sigma \leq s$.

We now return to integral characterizations and begin with a theorem due to Schoenberg (see, e.g., [Schoenberg (1938a)], p.816, or [Wells and Williams (1975)], p.27).

Theorem 3.6. A continuous function $\varphi : [0, \infty) \rightarrow \mathbb{R}$ is positive definite and radial on \mathbb{R}^s if and only if it is the Bessel transform of a finite non-negative Borel measure μ on $[0, \infty)$, i.e.

$$\varphi(r) = \int_0^\infty \Omega_s(rt) d\mu(t).$$

Here

$$\Omega_s(r) = \begin{cases} \cos r & \text{for } s = 1, \\ \Gamma\left(\frac{s}{2}\right) \left(\frac{2}{r}\right)^{(s-2)/2} J_{(s-2)/2}(r) & \text{for } s \geq 2, \end{cases}$$

and $J_{(s-2)/2}$ is the classical Bessel function of the first kind of order $(s-2)/2$.

As above, now the function $\Phi(x) = \cos(x)$ from Example 3.2 can be viewed as the fundamental positive definite radial function on \mathbb{R} . We will see below (in Example 3 of Chapter 4) that the characterization of Theorem 3.6 immediately suggests a class of (even strictly) positive definite radial functions. As for the basic 1D example, the measure μ will simply be a point evaluation measure.

A Fourier transform characterization of strictly positive definite radial functions on \mathbb{R}^s can be found in [Wendland (2005a)]. It is essentially a combination of Theorem 3.5 and the formula in Theorem B.1 of Appendix B for the Fourier transform of a radial function:

Theorem 3.7. A continuous function $\varphi : [0, \infty) \rightarrow \mathbb{R}$ such that $r \mapsto r^{s-1} \varphi(r) \in L_1[0, \infty)$ is strictly positive definite and radial on \mathbb{R}^s if and only if the s -dimensional Fourier transform

$$\mathcal{F}_s \varphi(r) = \frac{1}{\sqrt{r^{s-2}}} \int_0^\infty \varphi(t) t^{\frac{s}{2}} J_{(s-2)/2}(rt) dt$$

is non-negative and not identically equal to zero.

Since Lemma 3.1 states that any function that is (strictly) positive definite and radial on \mathbb{R}^s is also (strictly) positive definite and radial on \mathbb{R}^σ for any $\sigma \leq s$, those functions which are (strictly) positive definite and radial on \mathbb{R}^s for all s are of particular interest. The class of functions that are positive definite on \mathbb{R}^s for all s was also characterized by Schoenberg ([Schoenberg (1938a)], pp. 817–821). An extension to the strictly positive definite case can be found in [Micchelli (1986)]:

Theorem 3.8 (Schoenberg). A continuous function $\varphi : [0, \infty) \rightarrow \mathbb{R}$ is strictly positive definite and radial on \mathbb{R}^s for all s if and only if it is of the form

$$\varphi(r) = \int_0^\infty e^{-r^2 t^2} d\mu(t),$$

where μ is a finite non-negative Borel measure on $[0, \infty)$ not concentrated at the origin.

As suggested for Theorem 3.6 above, letting μ be a point evaluation measure in Theorem 3.8 we obtain that the Gaussian is strictly positive definite and radial on \mathbb{R}^s for all s (c.f. Example 1 of Chapter 4).

The Schoenberg characterization of (strictly) positive definite radial functions on \mathbb{R}^s for all s (Theorem 3.8) implies that we have a finite non-negative Borel measure μ on $[0, \infty)$ such that

$$\varphi(r) = \int_0^\infty e^{-r^2 t^2} d\mu(t).$$

If we want to find a zero r_0 of φ then we have to solve

$$\varphi(r_0) = \int_0^\infty e^{-r_0^2 t^2} d\mu(t) = 0.$$

Since the exponential function is positive and the measure is non-negative, it follows that μ must be the zero measure. However, then φ is identically equal to zero. Therefore, a non-trivial function φ that is positive definite and radial on \mathbb{R}^s for all s can have no zeros. This implies in particular that

Theorem 3.9. There are no oscillatory univariate continuous functions that are strictly positive definite and radial on \mathbb{R}^s for all s . Moreover, there are no compactly supported univariate continuous functions that are strictly positive definite and radial on \mathbb{R}^s for all s .

An equivalent argument for the oscillatory case is given in Theorem 2.3 of [Fornberg et al. (2004)].

Chapter 4

Examples of Strictly Positive Definite Radial Functions

We now present a number of functions that are covered by the theory presented thus far. While it is possible to include a *shape parameter* ε for all of the functions presented in the examples below by rescaling \mathbf{x} to $\varepsilon\mathbf{x}$, we avoid its use in the formulation of all but the Gaussian example to keep the formulas as simple as possible. We do, however, use a shape parameter when plotting some of the basis functions.

Our use of the shape parameter does not always match its “traditional” use. For example, Hardy introduced his inverse multiquadratics (see Example 5 below) in the form $\Phi(\|\mathbf{x}\|) = 1/\sqrt{c^2 + \|\mathbf{x}\|^2}$ with shape parameter c . It is, of course, straightforward to transform this representation to the one suggested above, *i.e.*, $\Phi(\|\mathbf{x}\|) = 1/\sqrt{1 + \varepsilon^2 \|\mathbf{x}\|^2}$, by setting $c^2 = 1/\varepsilon^2$ and scaling the result by $1/|\varepsilon|$.

Our use of the shape parameter as a factor applied directly to \mathbf{x} has the advantage of providing a unified treatment in which a decrease of the shape parameter always has the effect of producing “flat” basis functions, while increasing ε leads to more peaked (or localized) basis functions.

4.1 Example 1: Gaussians

We can now show that the *Gaussian*

$$\Phi(\mathbf{x}) = e^{-\varepsilon^2 \|\mathbf{x}\|^2}, \quad \varepsilon > 0, \quad (4.1)$$

is strictly positive definite (and radial) on \mathbb{R}^s for any s . This is due to the fact that the Fourier transform of a Gaussian is essentially a Gaussian. In fact,

$$\hat{\Phi}(\boldsymbol{\omega}) = \frac{1}{(\sqrt{2\varepsilon})^s} e^{-\frac{\|\boldsymbol{\omega}\|^2}{4\varepsilon^2}},$$

and this is positive independent of the space dimension s . In particular, for $\varepsilon = \frac{1}{\sqrt{2}}$ we have $\hat{\Phi} = \Phi$. Plots of Gaussian RBFs were presented in Fig. 2.1. Clearly, the Gaussians are infinitely differentiable. Some of its derivatives (as well as those of many other RBFs) are collected in Appendix D.

Another argument to show that Gaussians are strictly positive definite and radial on \mathbb{R}^s for any s that avoids dealing with Fourier transforms will become available later. It will make use of completely monotone functions.

Recall that Property (1) of Theorem 3.1 shows that any finite non-negative linear combination of (strictly) positive definite functions is again (strictly) positive definite. Moreover, we just saw that Gaussians are strictly positive definite and radial on all \mathbb{R}^s . Now, the Schoenberg characterization of functions that are (strictly) positive definite and radial on any \mathbb{R}^s , Theorem 3.8, states that *all* such functions are given as infinite linear combinations of Gaussians. Therefore, the Gaussians can be viewed as the fundamental member of the family of functions that are strictly positive definite and radial on \mathbb{R}^s for all s .

Since Gaussians play a central role in statistics this is a good place to mention that positive definite functions are — up to a normalization $\Phi(0) = 1$ — identical with characteristic functions of distribution functions in statistics.

4.2 Example 2: Laguerre-Gaussians

In order to obtain a generalization of Gaussians we start with the *generalized Laguerre polynomials* $L_n^{s/2}$ of degree n and order $s/2$ defined by their Rodrigues formula (see, e.g., formula (6.2.1) in [Andrews *et al.* (1999)])

$$L_n^{s/2}(t) = \frac{e^{t-s/2}}{n!} \frac{d^n}{dt^n} (e^{-t} t^{n+s/2}), \quad n = 1, 2, 3, \dots$$

An explicit formula for the generalized Laguerre polynomials is

$$L_n^{s/2}(t) = \sum_{k=0}^n \frac{(-1)^k}{k!} \binom{n+s/2}{n-k} t^k.$$

We then define the *Laguerre-Gaussians*

$$\Phi(\mathbf{x}) = e^{-\|\mathbf{x}\|^2} L_n^{s/2}(\|\mathbf{x}\|^2), \quad (4.2)$$

and list their Fourier transforms as

$$\hat{\Phi}(\boldsymbol{\omega}) = \frac{e^{-\frac{\|\boldsymbol{\omega}\|^2}{4}}}{\sqrt{2^s}} \sum_{j=0}^n \frac{\|\boldsymbol{\omega}\|^{2j}}{j! 4^j} \geq 0. \quad (4.3)$$

Note that the definition of the Laguerre-Gaussians depends on the space dimension s . Therefore they are strictly positive definite and radial on \mathbb{R}^s (and by Lemma 3.1 also on \mathbb{R}^σ for any $\sigma \leq s$).

Laguerre-Gaussian functions for some special choices of s and n are listed in Table 4.1. Figure 4.1 shows a Laguerre-Gaussian for $s = 1, n = 2$, and for $s = 2, n = 2$ displayed with a shape parameter $\varepsilon = 3$ and scaled so that $\Phi(\mathbf{0}) = 1$. Moreover, the Laguerre-Gaussians are infinitely smooth for all choices of n and s .

Note that the Laguerre-Gaussians (while being strictly positive definite functions) are not positive. Since the Laguerre-Gaussians are oscillatory functions we

know from Theorem 3.9 that they cannot be strictly positive definite and radial on \mathbb{R}^s for all s . We will encounter these functions later in the context of approximate moving least squares approximation (*c.f.* Chapter 26).

Table 4.1 Laguerre-Gaussians for various choices of s and n .

s	$n = 1$	$n = 2$
1	$\left(\frac{3}{2} - x ^2\right) e^{- x ^2}$	$\left(\frac{15}{8} - \frac{5}{2} x ^2 + \frac{1}{2} x ^4\right) e^{- x ^2}$
2	$(2 - \ \mathbf{x}\ ^2) e^{-\ \mathbf{x}\ ^2}$	$(3 - 3\ \mathbf{x}\ ^2 + \frac{1}{2}\ \mathbf{x}\ ^4) e^{-\ \mathbf{x}\ ^2}$
3	$\left(\frac{5}{2} - \ \mathbf{x}\ ^2\right) e^{-\ \mathbf{x}\ ^2}$	$\left(\frac{35}{8} - \frac{7}{2}\ \mathbf{x}\ ^2 + \frac{1}{2}\ \mathbf{x}\ ^4\right) e^{-\ \mathbf{x}\ ^2}$

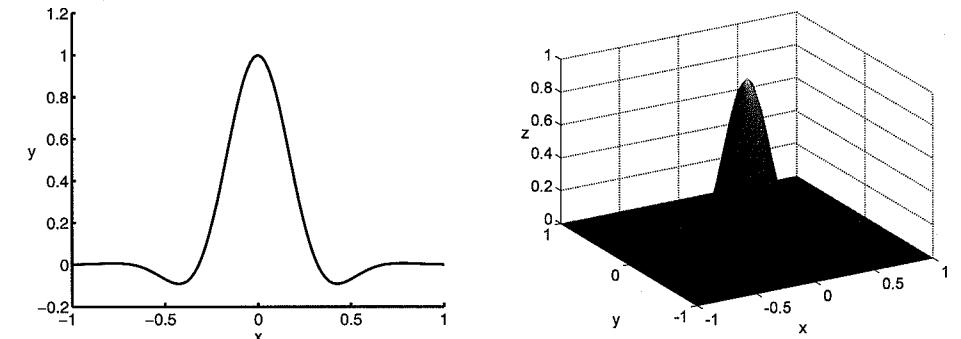


Fig. 4.1 Laguerre-Gaussians with $s = 1, n = 2$ (left) and $s = 2, n = 2$ (right) centered at the origin.

4.3 Example 3: Poisson Radial Functions

Another class of oscillatory functions that are strictly positive definite and radial on \mathbb{R}^s (and all \mathbb{R}^σ for $\sigma \leq s$) were recently studied by Fornberg and co-workers (see [Fornberg *et al.* (2004)] and also [Flyer (2006)]). These functions are of the form

$$\Phi(\mathbf{x}) = \frac{J_{s/2-1}(\|\mathbf{x}\|)}{\|\mathbf{x}\|^{s/2-1}}, \quad s \geq 2, \quad (4.4)$$

where J_ν is the Bessel function of the first kind of order ν . While these functions are not defined at the origin they can be extended to be infinitely differentiable in all of \mathbb{R}^s .

The functions (4.4) were already studied by Schoenberg (see the discussion surrounding Theorem 3.6) who suggested calling them *Poisson functions*. In fact, the functions in (4.4) are (up to the scale factor $2^{(s-2)/2} \Gamma(s/2)$) the functions Ω_s of

Theorem 3.6 and therefore can be viewed as the fundamental member of the family of functions that are strictly positive definite and radial on \mathbb{R}^s for fixed s .

Schoenberg showed that the functions Ω_s are given by

$$\Omega_s(\mathbf{x}) = \frac{1}{\omega_{s-1}} \int_{S^{s-1}} e^{i\mathbf{x} \cdot \sigma} d\sigma,$$

where ω_{s-1} denotes the area of the unit sphere S^{s-1} in \mathbb{R}^s , and $d\sigma$ denotes the usual measure on S^{s-1} .

The Poisson functions are another generalization of Gaussians (the fundamental strictly positive definite radial function on \mathbb{R}^s for all s) since the following limiting relation due to John von Neumann holds (see the discussion in [Schoenberg (1938a)]):

$$\lim_{s \rightarrow \infty} \Omega_s(r\sqrt{2s}) = e^{-r^2}.$$

Since the Poisson radial functions are defined in terms of Bessel functions they are also *band-limited*, i.e., their Fourier transform has compact support. In fact, the Fourier transform of Φ in \mathbb{R}^σ , $\sigma \leq s$, is given by (see [Flyer (2006)])

$$\hat{\Phi}(\omega) = \frac{1}{2^{\sigma-1}\Gamma(\frac{s-\sigma}{2})\pi^\sigma} (1 - \|\omega\|^2)^{(s-\sigma-2)/2}, \quad -1 < \omega_1, \dots, \omega_s < 1.$$

Some of these Poisson functions are listed in Table 4.2 and displayed in Figure 4.2 (where a shape parameter $\varepsilon = 10$ was used for the plots).

Table 4.2 Poisson functions for various choices of s .

$s = 2$	$s = 3$	$s = 4$	$s = 5$
$J_0(\ \mathbf{x}\)$	$\sqrt{\frac{2}{\pi}} \frac{\sin(\ \mathbf{x}\)}{\ \mathbf{x}\ }$	$\frac{J_1(\ \mathbf{x}\)}{\ \mathbf{x}\ }$	$\sqrt{\frac{2}{\pi}} \frac{\sin(\ \mathbf{x}\) - \ \mathbf{x}\ \cos(\ \mathbf{x}\)}{\ \mathbf{x}\ ^3}$

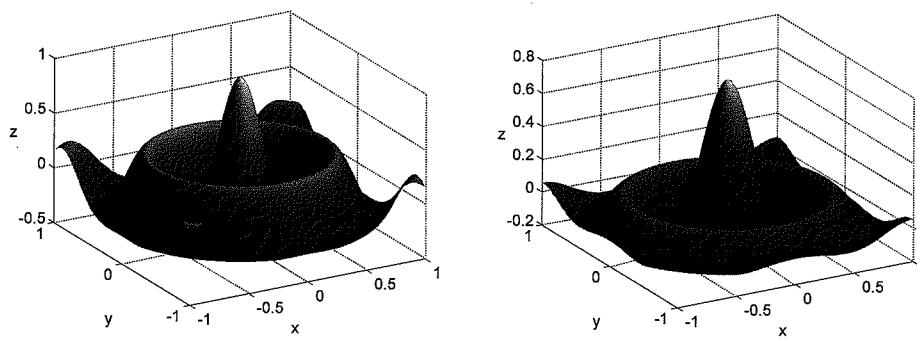


Fig. 4.2 Poisson functions with $s = 2$ (left) and $s = 3$ (right) centered at the origin in \mathbb{R}^2 .

4.4 Example 4: Matérn Functions

A fourth example of strictly positive definite functions is given by the class of *Matérn functions* which are quite common in the statistics literature (see, e.g., [Matérn (1986)] or [Stein (1999)])

$$\Phi(\mathbf{x}) = \frac{K_{\beta-\frac{s}{2}}(\|\mathbf{x}\|)\|\mathbf{x}\|^{\beta-\frac{s}{2}}}{2^{\beta-1}\Gamma(\beta)}, \quad \beta > \frac{s}{2}. \quad (4.5)$$

Here K_ν is the *modified Bessel function of the second kind* (sometimes also called modified Bessel function of the third kind, or MacDonald's function) of order ν . The Fourier transform of the Matérn functions is given by the *Bessel kernels*

$$\hat{\Phi}(\omega) = (1 + \|\omega\|^2)^{-\beta} > 0.$$

Therefore the Matérn functions are strictly positive definite on \mathbb{R}^s for all $s < 2\beta$. Schaback calls these functions *Sobolev splines* (see, e.g., [Schaback (1995a)] or his earlier discussion in [Schaback (1993)]) since they are naturally related to Sobolev spaces (see Chapter 13). These functions are also discussed in the relatively early paper [Dix and Osgden (1994)].

Some simple representatives of the family of Matérn functions are listed (up to a dimension-dependent scale factor) in Table 4.3. Note that the scaled functions listed in Table 4.3 do not depend on s . Since the modified Bessel functions are positive, so are the Matérn functions. Two examples are displayed in Figure 4.3. The function on the left is displayed using a shape parameter $\varepsilon = 3$. The plot on the right is scaled so that the value at the origin equals one and uses a shape parameter $\varepsilon = 10$. Note that the function on the left (corresponding to $\beta = \frac{s+1}{2}$) is not differentiable at the origin. The Matérn function for $\beta = \frac{s+3}{2}$ is C^2 smooth, and that for $\beta = \frac{s+5}{2}$ is in $C^4(\mathbb{R}^s)$.

Table 4.3 Matérn functions for various choices of β .

$\beta = \frac{s+1}{2}$	$\beta = \frac{s+3}{2}$	$\beta = \frac{s+5}{2}$
$e^{-\ \mathbf{x}\ }$	$(1 + \ \mathbf{x}\)e^{-\ \mathbf{x}\ }$	$(3 + 3\ \mathbf{x}\ + \ \mathbf{x}\ ^2)e^{-\ \mathbf{x}\ }$

4.5 Example 5: Generalized Inverse Multiquadratics

Since both Φ and $\hat{\Phi}$ in the previous example are positive radial functions we can use the Hankel inversion theorem (see Appendix B) to reverse their roles and see that the so-called *generalized inverse multiquadratics*

$$\Phi(\mathbf{x}) = (1 + \|\mathbf{x}\|^2)^{-\beta}, \quad \beta > \frac{s}{2}, \quad (4.6)$$

are strictly positive definite on \mathbb{R}^s for $s < 2\beta$. Generalized inverse multiquadratics are infinitely differentiable. By using another argument based on completely monotone

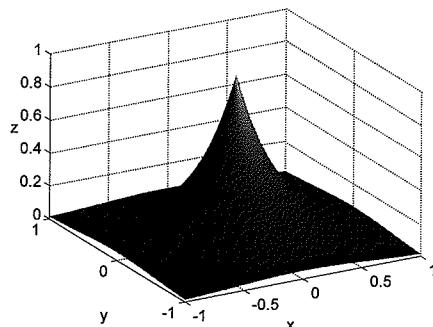


Fig. 4.3 Matérn functions with $\beta = \frac{s+1}{2}$ (left) and $\beta = \frac{s+5}{2}$ (right) centered at the origin in \mathbb{R}^2 .

functions we will be able to show that in fact we need to require only $\beta > 0$, and therefore the generalized inverse multiquadratics are strictly positive definite on \mathbb{R}^s for any s .

The “original” inverse multiquadric was introduced by Hardy in the early 1970s and corresponds to the value $\beta = 1/2$. The special choice $\beta = 1$ was referred to as *inverse quadratic* in various papers of Fornberg and co-workers (see, e.g., [Fornberg and Wright (2004)]). These two functions are displayed in Figure 4.4 using a shape parameter $\varepsilon = 5$.

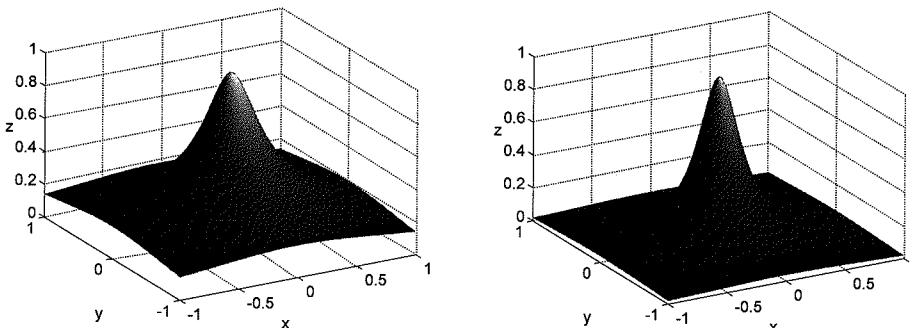


Fig. 4.4 Inverse multiquadric ($\beta = \frac{1}{2}$, left) and inverse quadratic ($\beta = 1$, right) centered at the origin in \mathbb{R}^2 .

4.6 Example 6: Truncated Power Functions

We now present an example of a family of strictly positive definite functions with *compact support*. Note that due to the observation made in Theorem 3.9 at the end of the previous chapter, they can not be strictly positive definite on \mathbb{R}^s for all s .

The *truncated power functions*

$$\varphi_\ell(r) = (1 - r)_+^\ell \quad (4.7)$$

give rise to strictly positive definite and radial functions on \mathbb{R}^s provided ℓ satisfies $\ell \geq \lfloor \frac{s}{2} \rfloor + 1$. Finding the Fourier transform of the truncated power function is rather involved. For details we refer to [Wendland (2005a)]. We will later use a simpler test based on multiply monotone functions to establish the strict positive definiteness of the truncated power functions. In (4.7) we used the *cutoff function* $(\cdot)_+$ which is defined by

$$(x)_+ = \begin{cases} x, & \text{for } x \geq 0, \\ 0, & \text{for } x < 0. \end{cases}$$

The cutoff function can be implemented conveniently in MATLAB using the `max` function, i.e., if `fx` is a vector of function values of f for different choices of x , then `max(fx,0)` computes $(f(x))_+$. We also point out that the expressions of the form $(1 - r)_+^\ell$ are to be interpreted as $((1 - r)_+)^{\ell}$, i.e., we first apply the cutoff function, and then the power.

Two different truncated power functions (with $\ell = 2, 4$) are displayed in Figure 4.5. While none of the truncated power functions are differentiable at the origin, the smoothness at the boundary of the support increases with ℓ .

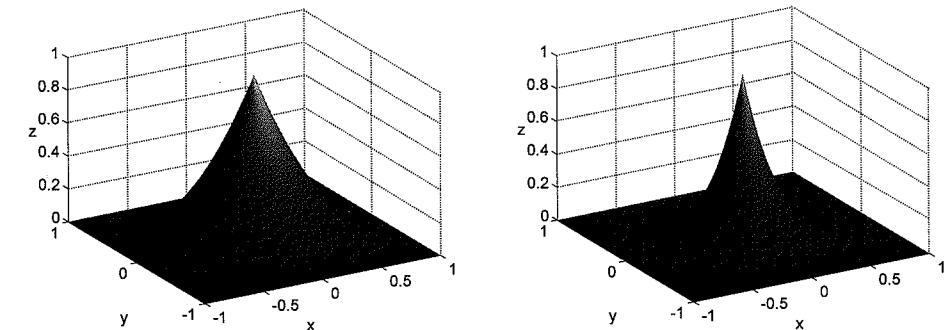


Fig. 4.5 Truncated power function with $\ell = 2$ (left) and $\ell = 4$ (right) centered at the origin in \mathbb{R}^2 .

4.7 Example 7: Potentials and Whittaker Radial Functions

Let $f \in C[0, \infty)$ be non-negative and not identically equal to zero, and define the function φ by

$$\varphi(r) = \int_0^\infty (1 - rt)_+^{k-1} f(t) dt. \quad (4.8)$$

Then $\Phi = \varphi(\|\cdot\|)$ is strictly positive definite and radial on \mathbb{R}^s provided $k \geq \lfloor \frac{s}{2} \rfloor + 2$ (see also Theorem 5.5 below). This can be verified by considering the quadratic form

$$\sum_{j=1}^N \sum_{k=1}^N c_j c_k \varphi(\|x_j - x_k\|) = \int_0^\infty \sum_{j=1}^N \sum_{k=1}^N c_j c_k \varphi_{k-1}(t \|x_j - x_k\|) f(t) dt$$

which is non-negative since the truncated power function $\varphi_{k-1}(\|\cdot\|)$ is strictly positive definite by Example 6, and f is non-negative. Since f is also assumed to be not identically equal to zero, the only way for the quadratic form to equal zero is if $c = \mathbf{0}$, and therefore φ is strictly positive definite.

For example, if we take $f(t) = t^\beta$, $\beta \geq 0$, then we get

$$\Phi(x) = \frac{\Gamma(k)\Gamma(\beta+1)}{\Gamma(k+\beta+1)\|x\|^{\beta+1}}. \quad (4.9)$$

While these functions are strictly positive definite and radial they are also singular at the origin and therefore not useful for our purposes. However, these functions are — up to scaling — generalizations of the *Coulomb potential* (for $\beta = 0$), and can therefore be given a physical interpretation.

Another possibility is to take $f(t) = t^\alpha e^{-\beta t}$, $\alpha \geq 0, \beta > 0$. Then we get

$$\Phi(x) = \frac{\|x\|^{(k-\alpha)/2}\Gamma(1+\alpha)\Gamma(k)}{\beta^{1+(k+\alpha)/2}\Gamma(k+\alpha+2)} e^{-\frac{\beta}{2\|x\|}} \times \quad (4.10)$$

$$\left(k M_{(\alpha-k)/2, (k+\alpha+1)/2} \left(\frac{\beta}{\|x\|} \right) + (1+\alpha) M_{1-(k-\alpha)/2, (k+\alpha+1)/2} \left(\frac{\beta}{\|x\|} \right) \right).$$

Here $M_{\mu,\nu}$ is the Whittaker- M function, a confluent hypergeometric function (see, e.g., Chapter 13 of [Abramowitz and Stegun (1972)]). When ν is a half-integer (which is, e.g., the case for integer k and α) formula (4.10) simplifies significantly. Examples for various integer values of k and α are listed in Table 4.4. Note that these functions are not defined at the origin. However, they can be made (only) continuous at the origin. Plots of two of these functions are provided in Figure 4.6. Note that only the functions for $k \geq 3$ are guaranteed to be strictly positive definite and radial on \mathbb{R}^3 .

Table 4.4 Whittaker radial functions Φ for various choices of k and α .

α	$k = 2$	$k = 3$
0	$\frac{\beta - \ x\ + \ x\ e^{-\frac{\beta}{\ x\ }}}{\beta^2}$	$\frac{\beta^2 - 2\beta\ x\ + 2\ x\ ^2 - 2\ x\ ^2e^{-\frac{\beta}{\ x\ }}}{\beta^3}$
1	$\frac{\beta - 2\ x\ + (\beta + 2\ x\)e^{-\frac{\beta}{\ x\ }}}{\beta^3}$	$\frac{\beta^2 - 4\beta\ x\ + 6\ x\ ^2 - (2\beta\ x\ + 6\ x\ ^2)e^{-\frac{\beta}{\ x\ }}}{\beta^4}$

Equation (4.8) amounts to another integral transform of f (not listed in Appendix B) with the compactly supported truncated power function as integration kernel. We will take another look at these functions in the context of multiply monotone functions below.

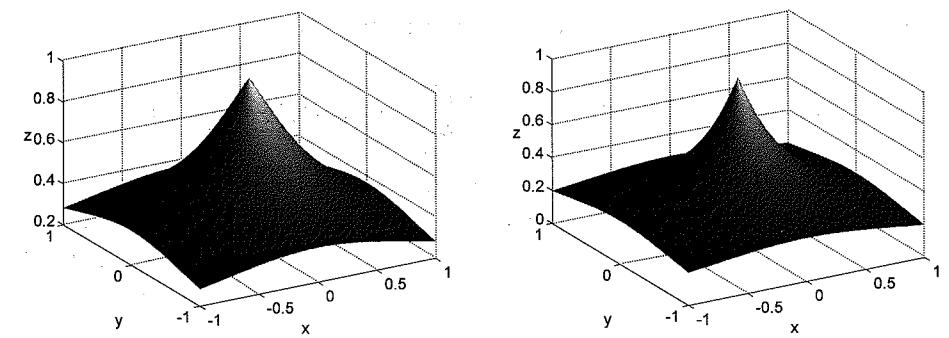


Fig. 4.6 Whittaker radial functions for $\alpha = 0$ and $\beta = 1$ with $k = 2$ (left) and $k = 3$ (right) centered at the origin in \mathbb{R}^2 .

4.8 Example 8: Integration Against Strictly Positive Definite Kernels

In fact, in [Wendland (2005a)] it is shown that integration of any non-negative function f that is not identically equal to zero against a function $K(t, \cdot)$ that is strictly positive definite on \mathbb{R}^s leads to another function that is strictly positive definite on \mathbb{R}^s , i.e.,

$$\varphi(r) = \int_0^\infty K(t, r) f(t) dt$$

gives rise to $\Phi = \varphi(\|\cdot\|)$ being strictly positive definite on \mathbb{R}^s . By choosing f and K appropriately we can obtain both globally supported and compactly supported functions.

For example, the multiply monotone functions in Williamson's characterization Theorem 5.4 are covered by this general theorem by taking $K(t, r) = (1-rt)_+^{k-1}$ and f an arbitrary positive function in L_1 so that $d\mu(t) = f(t)dt$. Also, functions that are strictly positive definite and radial on \mathbb{R}^s for all s (or equivalently completely monotone functions) are covered by choosing $K(t, r) = e^{-rt}$.

4.9 Summary

To summarize the theory surveyed thus far we can say that any multivariate (radial) function Φ whose Fourier transform is non-negative can be used to generate a basis for the scattered data interpolation problem by shifting it to the data sites. The function Φ can be positive, oscillatory, or have compact support. However, if Φ has any zeros then it cannot be strictly positive definite on \mathbb{R}^s for all choices of s .

Chapter 5

Completely Monotone and Multiply Monotone Functions

Since Fourier transforms are not always easy to compute, we now present two alternative criteria that allow us to decide whether a function is strictly positive definite and radial on \mathbb{R}^s (one for the case of all s , and one for only limited choices of s).

5.1 Completely Monotone Functions

We begin with the former case. To this end we now introduce a class of functions that is very closely related to positive definite radial functions and leads to a simple characterization of such functions.

Definition 5.1. A function $\varphi : [0, \infty) \rightarrow \mathbb{R}$ that is in $C[0, \infty) \cap C^\infty(0, \infty)$ and satisfies

$$(-1)^\ell \varphi^{(\ell)}(r) \geq 0, \quad r > 0, \ell = 0, 1, 2, \dots,$$

is called *completely monotone on $[0, \infty)$* .

Example 5.1. The function $\varphi(r) = \varepsilon$, $\varepsilon \geq 0$, is completely monotone on $[0, \infty)$.

Example 5.2. The function $\varphi(r) = e^{-\varepsilon r}$, $\varepsilon \geq 0$, is completely monotone on $[0, \infty)$ since

$$(-1)^\ell \varphi^{(\ell)}(r) = \varepsilon^\ell e^{-\varepsilon r} \geq 0, \quad \ell = 0, 1, 2, \dots$$

Example 5.3. The function $\varphi(r) = \frac{1}{(1+r)^\beta}$, $\beta \geq 0$, is completely monotone on $[0, \infty)$ since

$$(-1)^\ell \varphi^{(\ell)}(r) = (-1)^{2\ell} \beta(\beta+1)\cdots(\beta+\ell-1)(1+r)^{-\beta-\ell} \geq 0, \quad \ell = 0, 1, 2, \dots$$

Some properties of completely monotone functions that can be found in [Cheney and Light (1999); Feller (1966); Widder (1941)] are:

- (1) A non-negative finite linear combination of completely monotone functions is completely monotone.
- (2) The product of two completely monotone functions is completely monotone.

- (3) If φ is completely monotone and ψ is absolutely monotone (i.e., $\psi^{(\ell)} \geq 0$ for all $\ell \geq 0$), then $\psi \circ \varphi$ is completely monotone.
- (4) If φ is completely monotone and ψ is a positive function such that its derivative is completely monotone, then $\varphi \circ \psi$ is completely monotone.

Note that the functions in the second and third example above are, except for a variable substitution $r \mapsto r^2$, similar to the Gaussian and inverse multiquadratics mentioned earlier. In order to see how completely monotone functions are related to strictly positive definite radial functions we require an integral characterization of completely monotone functions.

Theorem 5.1 (Hausdorff-Bernstein-Widder). *A function $\varphi : [0, \infty) \rightarrow \mathbb{R}$ is completely monotone on $[0, \infty)$ if and only if it is the Laplace transform of a finite non-negative Borel measure μ on $[0, \infty)$, i.e., φ is of the form*

$$\varphi(r) = \mathcal{L}\mu(r) = \int_0^\infty e^{-rt} d\mu(t).$$

Proof. Widder's proof of this theorem can be found in [Widder (1941)], p. 160, where he reduces the proof of this theorem to another theorem by Hausdorff on completely monotone sequences. A detailed proof can also be found in the books [Cheney and Light (1999); Wendland (2005a)]. \square

Theorem 5.1 shows that, in the spirit of our earlier remarks, the function $\varphi(r) = e^{-\varepsilon r}$ can be viewed as the fundamental completely monotone function.

The following connection between positive definite radial and completely monotone functions was first pointed out by Schoenberg in 1938.

Theorem 5.2. *A function φ is completely monotone on $[0, \infty)$ if and only if $\Phi = \varphi(\|\cdot\|^2)$ is positive definite and radial on \mathbb{R}^s for all s .*

Note that the function Φ is now defined via the *square* of the norm. This differs from our definition of radial functions (see Definition 2.1).

Proof. One possibility is to use a change of variables to combine Schoenberg's characterization of functions that are positive definite and radial on any \mathbb{R}^s , Theorem 3.8, with the Hausdorff-Bernstein-Widder characterization of completely monotone functions. To get more insight we present an alternative proof of the claim that the completely monotone function φ gives rise to a Φ that is positive definite and radial on any \mathbb{R}^s . Details for the other direction can be found, e.g., in [Wendland (2005a)].

The Hausdorff-Bernstein-Widder theorem implies that we can write φ as

$$\varphi(r) = \int_0^\infty e^{-rt} d\mu(t)$$

with a finite non-negative Borel measure μ . Therefore, $\Phi(\mathbf{x}) = \varphi(\|\mathbf{x}\|^2)$ has the representation

$$\Phi(\mathbf{x}) = \int_0^\infty e^{-\|\mathbf{x}\|^2 t} d\mu(t).$$

To see that this function is positive definite on any \mathbb{R}^s we consider the quadratic form

$$\sum_{j=1}^N \sum_{k=1}^N c_j c_k \Phi(\mathbf{x}_j - \mathbf{x}_k) = \int_0^\infty \sum_{j=1}^N \sum_{k=1}^N c_j c_k e^{-t \|\mathbf{x}_j - \mathbf{x}_k\|^2} d\mu(t).$$

Since we saw earlier that the Gaussians are strictly positive definite and radial on any \mathbb{R}^s it follows that the quadratic form is non-negative. \square

We can see from the previous proof that if the measure μ is not concentrated at the origin, then Φ is even strictly positive definite and radial on any \mathbb{R}^s . This condition on the measure is equivalent with φ not being constant. With this additional restriction on φ we can apply the notion of a completely monotone function to the scattered data interpolation problem. The following *interpolation theorem* originates in the work of Schoenberg ([Schoenberg (1938a)], p. 823) who showed that complete monotonicity implies strict positive definiteness, thus providing a very simple test for verifying the well-posedness of many scattered data interpolation problems. A proof that the converse also holds can be found in [Wendland (2005a)].

Theorem 5.3. *A function $\varphi : [0, \infty) \rightarrow \mathbb{R}$ is completely monotone but not constant if and only if $\varphi(\|\cdot\|^2)$ is strictly positive definite and radial on \mathbb{R}^s for any s .*

Example 5.4. Since we showed above that the functions $\varphi(r) = e^{-\varepsilon r}$, $\varepsilon > 0$, and $\varphi(r) = 1/(1+r)^\beta$, $\beta \geq 0$, are completely monotone on $[0, \infty)$, and since they are also not constant we know from Theorem 5.3 that the Gaussians $\Phi(\mathbf{x}) = \varphi(\|\mathbf{x}\|^2) = e^{-\varepsilon^2 \|\mathbf{x}\|^2}$, $\varepsilon > 0$, and inverse multiquadratics $\Phi(\mathbf{x}) = \varphi(\|\mathbf{x}\|^2) = 1/(1+\|\mathbf{x}\|^2)^\beta$, $\beta \geq 0$, are strictly positive definite and radial on \mathbb{R}^s for all s . Not only is the test for complete monotonicity a simpler one than calculation of the Fourier transforms, but we also are able to verify strict positive definiteness of the inverse multiquadratics without any dependence of s on β .

5.2 Multiply Monotone Functions

We can also use monotonicity to test for strict positive definiteness of radial functions on \mathbb{R}^s for some fixed value of s . To this end we introduce the concept of a *multiply monotone function*.

Definition 5.2. A function $\varphi : (0, \infty) \rightarrow \mathbb{R}$ which is in $C^{k-2}(0, \infty)$, $k \geq 2$, and for which $(-1)^l \varphi^{(l)}(r)$ is non-negative, non-increasing, and convex for $l = 0, 1, 2, \dots, k-2$

2 is called k -times monotone on $(0, \infty)$. In case $k = 1$ we only require $\varphi \in C(0, \infty)$ to be non-negative and non-increasing.

Since convexity of φ means that $\varphi\left(\frac{r_1+r_2}{2}\right) \leq \frac{\varphi(r_1)+\varphi(r_2)}{2}$, or simply $\varphi''(r) \geq 0$ if φ'' exists, a multiply monotone function is in essence just a completely monotone function whose monotonicity is “truncated”.

Example 5.5. The truncated power function (c.f. (4.7))

$$\varphi_\ell(r) = (1 - r)_+^\ell$$

is ℓ -times monotone for any ℓ since

$$(-1)^l \varphi_\ell^{(l)}(r) = \ell(\ell - 1) \dots (\ell - l + 1)(1 - r)_+^{\ell-l} \geq 0, \quad l = 0, 1, 2, \dots, \ell.$$

We saw in Section 4.6 that the truncated power functions lead to radial functions that are strictly positive definite on \mathbb{R}^s provided $\ell \geq \lfloor s/2 \rfloor + 1$.

Example 5.6. If we define the integral operator I by

$$(If)(r) = \int_r^\infty f(t) dt, \quad r \geq 0, \quad (5.1)$$

and f is ℓ -times monotone, then If is $\ell + 1$ -times monotone. This follows immediately from the fundamental theorem of calculus. As we will see later, the operator I plays an important role in the construction of compactly supported radial basis functions.

To make the connection to strictly positive definite radial functions we require an integral representation for the class of multiply monotone functions. This was given in [Williamson (1956)] but apparently already known to Schoenberg in 1940.

Theorem 5.4 (Williamson). A continuous function $\varphi : (0, \infty) \rightarrow \mathbb{R}$ is k -times monotone on $(0, \infty)$ if and only if it is of the form

$$\varphi(r) = \int_0^\infty (1 - rt)_+^{k-1} d\mu(t), \quad (5.2)$$

where μ is a non-negative Borel measure on $(0, \infty)$.

Proof. To see that a function of the form (5.2) is indeed multiply monotone we just need to differentiate under the integral (since derivatives up to order $k - 2$ of $(1 - rt)_+^{k-1}$ are continuous and bounded). The other direction can be found in [Williamson (1956)]. \square

Williamson's characterization shows us that — just like the truncated power functions — the Whittaker radial functions (4.10) in Section 4.7 are based on multiply monotone functions.

For $k \rightarrow \infty$ the Williamson characterization corresponds to the Hausdorff-Bernstein-Widder characterization Theorem 5.1 of completely monotone functions

(and is equivalent provided we extend Williamson's work to include continuity at the origin).

We can see from Sections 4.6 and 4.7 that multiply monotone functions give rise to positive definite radial functions. Such a connection was first noted in [Askey (1973)] (and in the one-dimensional case by Pólya) using the truncated power functions of Section 4.6.

In the RBF literature the following theorem was stated in [Micchelli (1986)], and then refined in [Buhmann (1993a)]:

Theorem 5.5 (Micchelli). Let $k = \lfloor s/2 \rfloor + 2$ be a positive integer. If $\varphi : [0, \infty) \rightarrow \mathbb{R}$, $\varphi \in C[0, \infty)$, is k -times monotone on $(0, \infty)$ but not constant, then φ is strictly positive definite and radial on \mathbb{R}^s for any s such that $\lfloor s/2 \rfloor \leq k - 2$.

We would like to mention that several versions of Theorem 5.5 contain misprints in the literature. The correct form should be as stated above (c.f. also the generalization for strictly conditionally positive definite functions, Theorem 9.3).

Using Theorem 5.5 we can now verify the strict positive definiteness of the truncated power functions and Whittaker radial functions of Sections 4.6 and 4.7 without the use of Fourier transforms. Again, as for Gaussians and the Poisson radial functions, we can view the truncated power function as the fundamental compactly supported strictly positive definite radial function since it is obtained using the point evaluation measure in Williamson's characterization of a multiply monotone function.

It is interesting to observe a certain lack of symmetry in the theory for completely monotone and multiply monotone functions. First, in the completely monotone case we can use Theorem 5.3 to conclude that if φ is completely monotone and not constant then $\varphi^{(2)}$ is strictly positive definite on \mathbb{R}^s for any s . In the multiply monotone case (see Theorem 5.5) the square is missing. Now it is clear that we cannot expect the statement with a square to be true in the multiply monotone case. To see this we consider the truncated power function $\varphi(r) = (1 - r)_+^\ell$ (which we know — according to Example 5.1 above — to be ℓ -times multiply monotone for any ℓ). However, the function $\psi(r) = (1 - r^2)_+^\ell$ is not strictly positive definite and radial on \mathbb{R}^s for any s since it is not even strictly positive definite and radial on \mathbb{R} (and therefore even much less so on any higher-dimensional space). We can see this from the univariate radial Fourier transform of ψ (see Theorem B.1 of Appendix B with $s = 1$)

$$\begin{aligned} \mathcal{F}_1 \psi(r) &= \frac{1}{\sqrt{r^{-1}}} \int_0^\infty (1 - t^2)_+^\ell t^{\frac{1}{2}} J_{-1/2}(rt) dt \\ &= \sqrt{\frac{2}{\pi}} \int_0^1 (1 - t^2)^\ell \cos(rt) dt \\ &= 2^\ell \Gamma(\ell + 1) \frac{J_{\ell+1/2}(r)}{r^{\ell+1/2}}. \end{aligned}$$

Here we used the compact support of ψ and the fact that $J_{-1/2}(r) = \sqrt{2/\pi r} \cos r$. The function $\mathcal{F}_1\psi$ is oscillatory, and therefore ψ cannot be strictly positive definite (*c.f.* Theorem 3.5). In fact, the Fourier transform $\mathcal{F}_1\psi$ is closely related to the Poisson radial functions of Section 4.3.

Moreover, in the completely monotone case we have an equivalence between completely monotone and strictly positive definite functions that are radial on any \mathbb{R}^s (see Theorem 5.3). Again, we cannot expect such an equivalence to hold in the multiply monotone case, *i.e.*, the converse of Theorem 5.5 cannot be true. This is clear since we have already seen a number of functions that are strictly positive definite and radial, but not monotone at all — namely the oscillatory Laguerre-Gaussians of Section 4.2 and the Poisson radial functions of Section 4.3.

However, it is interesting to combine the Schoenberg Theorem 5.3 and Theorem 5.5 based on Williamson's characterization. If one starts with the strictly positive definite radial Gaussian $\varphi(r) = e^{-\varepsilon^2 r^2}$, then Theorem 5.3 tells us that $\phi(r) = \varphi(\sqrt{r}) = e^{-\varepsilon^2 r}$ is completely monotone. Now, any function that is completely monotone is also multiply monotone of any order, so that we can use Theorem 5.5 and conclude that the function $\phi(r) = e^{-\varepsilon^2 r}$ is also strictly positive definite and radial on \mathbb{R}^s for all s . Of course, now we can repeat the argument and conclude that $\psi(r) = e^{-\varepsilon^2 \sqrt{r}}$ is strictly positive definite and radial on \mathbb{R}^s for all s , and so on (see [Wendland (2005c)]). This result was already known to Schoenberg (at least in the non-strict case).

As a final remark in this chapter we mention that we are a long way from having a complete characterization of (radial) functions for which the scattered data interpolation problem has a unique solution. As we will see later, such an (as of now unknown) characterization will involve also functions which are not strictly positive definite. For example, we will mention a result of Micchelli's according to which *conditionally* positive definite functions of order one can be used for the scattered data interpolation problem. Furthermore, all of the results dealt with so far involve radial basis functions that are centered at the given data sites. There are only limited results addressing the situation in which the centers for the basis functions and the data sites may differ.

Chapter 6

Scattered Data Interpolation with Polynomial Precision

6.1 Interpolation with Multivariate Polynomials

As we mentioned in the introduction it is not an easy matter to use polynomials to perform multivariate scattered data interpolation. Only if the data sites are in certain special locations can we guarantee well-posedness of multivariate polynomial interpolation. We now address this problem.

Definition 6.1. We call a set of points $\mathcal{X} = \{x_1, \dots, x_N\} \subset \mathbb{R}^s$ *m-unisolvant* if the only polynomial of total degree at most m interpolating zero data on \mathcal{X} is the zero polynomial.

This definition guarantees a unique solution for interpolation to given data at a subset of cardinality $M = \binom{m+s}{m}$ of the points x_1, \dots, x_N by a polynomial of degree m . Here M is the dimension of the linear space Π_m^s of polynomials of total degree less than or equal to m in s variables.

For polynomial interpolation at N distinct data sites in \mathbb{R}^s to be a well-posed problem, the polynomial degree needs to be chosen accordingly, *i.e.*, we need $M = N$, and the data sites need to form an *m-unisolvant* set. This is rather restrictive. For example, this implies that polynomial interpolation at $N = 7$ points in \mathbb{R}^2 can not be done in a unique way since we could either attempt to use bivariate quadratic polynomials (for which $M = 6$), or bivariate cubic polynomials (with $M = 10$). There exists no space of bivariate polynomials for which $M = 7$.

We will see in the next chapter that *m-unisolvant* sets play an important role in the context of conditionally positive definite functions. There, however, even though we will be interested in interpolating N pieces of data, the polynomial degree will be small (usually $m = 1, 2, 3$), and the restrictions imposed on the locations of the data sites by the unisolvency conditions will be rather mild.

A sufficient condition (to be found in [Chui (1988)], Ch. 9) on the points x_1, \dots, x_N to form an *m-unisolvant* set in \mathbb{R}^2 is

Theorem 6.1. Suppose $\{L_0, \dots, L_m\}$ is a set of $m+1$ distinct lines in \mathbb{R}^2 , and that $\mathcal{U} = \{u_1, \dots, u_M\}$ is a set of $M = (m+1)(m+2)/2$ distinct points such that the

first point lies on L_0 , the next two points lie on L_1 but not on L_0 , and so on, so that the last $m+1$ points lie on L_m but not on any of the previous lines L_0, \dots, L_{m-1} . Then there exists a unique interpolation polynomial of total degree at most m to arbitrary data given at the points in \mathcal{U} . Furthermore, if the data sites $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ contain \mathcal{U} as a subset then they form an m -unisolvant set on \mathbb{R}^2 .

Proof. We use induction on m . For $m = 0$ the result is trivial. Take R to be the matrix arising from polynomial interpolation at the points in \mathcal{U} , i.e.,

$$R_{jk} = p_k(\mathbf{u}_j), \quad j, k = 1, \dots, M,$$

where the p_k form a basis of Π_m^2 . We want to show that the only possible solution to $Rc = \mathbf{0}$ is $c = \mathbf{0}$. This is equivalent to showing that if $p \in \Pi_m^2$ satisfies

$$p(\mathbf{u}_i) = 0, \quad i = 1, \dots, M,$$

then p is the zero polynomial.

For each $i = 1, \dots, m$, let the equation of the line L_i be given by

$$\alpha_i x + \beta_i y = \gamma_i,$$

where $\mathbf{x} = (x, y) \in \mathbb{R}^2$.

Suppose now that p interpolates zero data at all the points \mathbf{u}_i as stated above. Since p reduces to a univariate polynomial of degree m on L_m which vanishes at $m+1$ distinct points on L_m , it follows that p vanishes identically on L_m , and so

$$p(x, y) = (\alpha_m x + \beta_m y - \gamma_m)q(x, y),$$

where q is a polynomial of degree $m-1$. But now q satisfies the hypothesis of the theorem with m replaced by $m-1$ and \mathcal{U} replaced by $\tilde{\mathcal{U}}$ consisting of the first $\binom{m+1}{2}$ points of \mathcal{U} . By induction, therefore $q \equiv 0$, and thus $p \equiv 0$. This establishes the uniqueness of the interpolation polynomial. The last statement of the theorem is obvious. \square

A similar theorem was already proved in [Chung and Yao (1977)]. Theorem 6.1 can be generalized to \mathbb{R}^s by using hyperplanes. The proof is constructed with the help of an additional induction on s . Chui also gives an explicit expression for the determinant of the matrix associated with (polynomial) interpolation at the set of points \mathcal{U} .

Remark 6.1. For later reference we note that $(m-1)$ -unisolvency of the points $\mathbf{x}_1, \dots, \mathbf{x}_N$ is equivalent to the fact that the matrix P with

$$P_{jl} = p_l(\mathbf{x}_j), \quad j = 1, \dots, N, \quad l = 1, \dots, M,$$

has full (column-)rank. For $N = M$ this is the polynomial interpolation matrix.

Example 6.1. As can easily be verified, three collinear points in \mathbb{R}^2 are not 1-unisolvant, since a linear interpolant, i.e., a plane through three arbitrary heights at these three collinear points is not uniquely determined. On the other hand, if a set of points in \mathbb{R}^2 contains three non-collinear points, then it is 1-unisolvant.

We used the difficulties associated with multivariate polynomial interpolation as one of the motivations for the use of radial basis functions. However, sometimes it is desirable to have an interpolant that exactly reproduces certain types of functions. For example, if the data are constant, or come from a linear function, then it would be nice if our interpolant were also constant or linear, respectively. Unfortunately, the methods we have presented thus far (except for the distance matrix fit in the $s = 1$ case) do not reproduce these simple polynomial functions. Moreover, later on we will be interested in applying our interpolation methods to the numerical solution of partial differential equations, and practitioners (especially of finite element methods) often judge an interpolation method by its ability to pass the so-called *patch test*. An interpolation method passes the standard patch test if it can reproduce linear functions. In engineering applications this translates into exact calculation of constant stress and strain. We will see later that in order to prove error estimates for meshfree approximation methods it is not necessary to be able to reproduce polynomials globally (but local polynomial reproduction is an essential ingredient). Thus, if we are only concerned with the approximation power of a numerical method there is really no need for the standard patch test to hold.

6.2 Example: Reproduction of Linear Functions Using Gaussian RBFs

If we do insist on reproduction of linear functions then the top part of Figure 6.1 shows a Gaussian RBF interpolant ($\varepsilon = 6$) to the bivariate linear function $f(x, y) = (x + y)/2$ based on 1089 uniformly spaced points in the unit square along with the absolute error. Clearly the interpolant is not completely planar — not even to machine precision.

Fortunately, there is a simple remedy for this problem. All we need to do is add the polynomial functions $\mathbf{x} \mapsto 1$, $\mathbf{x} \mapsto x$, and $\mathbf{x} \mapsto y$ to the basis $\{e^{-\varepsilon^2 \|\cdot - \mathbf{x}_1\|^2}, \dots, e^{-\varepsilon^2 \|\cdot - \mathbf{x}_N\|^2}\}$ we have thus far been using to obtain our interpolant. However, now we have $N+3$ unknowns, namely the coefficients c_k , $k = 1, \dots, N+3$, in the expansion

$$\mathcal{P}_f(\mathbf{x}) = \sum_{k=1}^N c_k e^{-\varepsilon^2 \|\mathbf{x} - \mathbf{x}_k\|^2} + c_{N+1} + c_{N+2}x + c_{N+3}y, \quad \mathbf{x} = (x, y) \in \mathbb{R}^2,$$

and we have only N conditions to determine them, namely the interpolation conditions

$$\mathcal{P}_f(\mathbf{x}_j) = f(\mathbf{x}_j) = (x_j + y_j)/2, \quad j = 1, \dots, N.$$

What can we do to obtain a (non-singular) square system? As we will see below, we can add the following three conditions:

$$\sum_{k=1}^N c_k = 0,$$

$$\sum_{k=1}^N c_k x_k = 0,$$

$$\sum_{k=1}^N c_k y_k = 0.$$

How do we have to modify our existing MATLAB program for scattered data interpolation to incorporate these modifications? If we previously dealt with the solution of

$$Ac = y,$$

with $A_{jk} = e^{-\varepsilon^2 \|x_j - x_k\|^2}$, $j, k = 1, \dots, N$, $c = [c_1, \dots, c_N]^T$, and $y = [f(x_1), \dots, f(x_N)]^T$, then we now have to solve the *augmented system*

$$\begin{bmatrix} A & P \\ P^T & O \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} y \\ 0 \end{bmatrix}, \quad (6.1)$$

where A , c , and y are as before, and $P_{jl} = p_l(x_j)$, $j = 1, \dots, N$, $l = 1, \dots, 3$, with $p_1(x) = 1$, $p_2(x) = x$, and $p_3(x) = y$. Moreover, $\mathbf{0}$ is a zero vector of length 3, and O is a zero matrix of size 3×3 .

The MATLAB script `RBFInterpolation2Dlinear.m` shows an implementation of this approach for Gaussians (although they can easily be replaced by any other RBF) and test function $f(x, y) = (x + y)/2$. The resulting interpolant using $N = 9$ equally spaced data points and $\varepsilon = 6$ is shown in the bottom part of Figure 6.1. Now, while still not perfectly linear, the error is on the level of machine accuracy.

Program 6.1. `RBFInterpolation2Dlinear.m`

```
% RBFInterpolation2Dlinear
% Script that performs 2D RBF interpolation with reproduction of
% linear functions
% Calls on: DistanceMatrix
    % Define the Gaussian RBF and shape parameter
1 rbf = @(e,r) exp(-(e*r).^2); ep = 6;
    % Define linear test function
2 testfunction = @(x,y) (x+y)/2;
    % Number and type of data points
3 N = 9; gridtype = 'u';
    % Load data points
4 name = sprintf('Data2D_%d%s',N,gridtype); load(name)
5 ctrs = dsites;
6 neval = 40; M = neval^2; grid = linspace(0,1,neval);
7 [xe,ye] = meshgrid(grid); epoints = [xe(:) ye(:)];
    % Evaluate the test function at the data points.
8 rhs = testfunction(dsites(:,1),dsites(:,2));
```

```
% Add zeros for linear (2D) reproduction
9 rhs = [rhs; zeros(3,1)];
    % Compute distance matrix between the data sites and centers
10 DM_data = DistanceMatrix(dsites,ctrs);
    % Compute interpolation matrix
11 IM = rbf(ep,DM_data);
    % Define 3-column matrix P for linear reproduction
12 PM = [ones(N,1) dsites];
    % Augment interpolation matrix
13 IM = [IM PM; [PM' zeros(3,3)]]; % Compute distance matrix between evaluation points and centers
14 DM_eval = DistanceMatrix(epoints,ctrs);
    % Compute evaluation matrix
15 EM = rbf(ep,DM_eval);
    % Add column for constant reproduction
16 PM = [ones(M,1) epoints]; EM = [EM PM];
    % Compute RBF interpolant
    % (evaluation matrix * solution of interpolation system)
17 Pf = EM * (IM\rhs);
    % Compute maximum error on evaluation grid
18 exact = testfunction(epoints(:,1),epoints(:,2));
19 maxerr = norm(Pf-exact,inf);
20 rms_err = norm(Pf-exact)/neval;
21 fprintf('RMS error: %e\n', rms_err)
22 fprintf('Maximum error: %e\n', maxerr)
23 fview = [-30,30];
24 plotsurf(xe,ye,Pf,neval,exact,maxerr,fview);
25 ploterror2D(xe,ye,Pf,exact,maxerr,neval,fview);
```

Note that Program 6.1 is almost the same as Program 2.1. The only difference are lines 9, 12, 13, and 16 that have been added to deal with the augmented problem. In Program 6.1 we also modified the definition of the test function.

6.3 Scattered Data Interpolation with More General Polynomial Precision

As we just saw for a specific example, we may want to modify the assumption on the form (1.1) of the solution to the scattered data interpolation Problem 1.1 by adding certain polynomials to the expansion, *i.e.*, \mathcal{P}_f is now assumed to be of the form

$$\mathcal{P}_f(x) = \sum_{k=1}^N c_k \varphi(\|x - x_k\|) + \sum_{l=1}^M d_l p_l(x), \quad x \in \mathbb{R}^s, \quad (6.2)$$

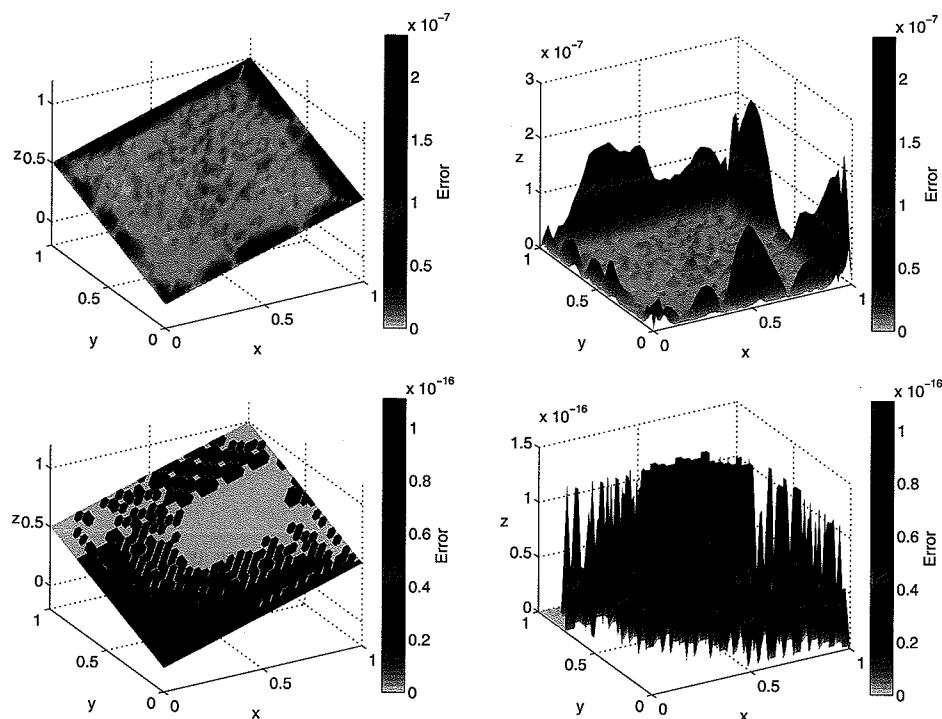


Fig. 6.1 Top: Gaussian interpolant to bivariate linear function with $N = 1089$ (left) and associated absolute error (right). Bottom: Interpolant based on linearly augmented Gaussians to bivariate linear function with $N = 9$ (left) and associated absolute error (right).

where p_1, \dots, p_M form a basis for the $M = \binom{m-1+s}{m-1}$ -dimensional linear space Π_{m-1}^s of polynomials of total degree less than or equal to $m - 1$ in s variables. It seems awkward to formulate this setup with polynomials in Π_{m-1}^s instead of degree m polynomials. However, in light of our discussion of conditionally positive definite functions in the next chapter this choice is quite natural.

Since enforcing the interpolation conditions $\mathcal{P}_f(\mathbf{x}_j) = f(\mathbf{x}_j)$, $j = 1, \dots, N$, leads to a system of N linear equations in the $N + M$ unknowns c_k and d_l one usually adds the M additional conditions

$$\sum_{k=1}^N c_k p_l(\mathbf{x}_k) = 0, \quad l = 1, \dots, M,$$

to ensure a unique solution. The example in the previous section represents the particular case $s = m = 2$.

While the use of polynomials is somewhat arbitrary (any other set of M linearly independent functions could also be used), it is obvious that the addition of polynomials of total degree at most $m - 1$ guarantees polynomial precision provided the points in \mathcal{X} form an $(m - 1)$ -unisolvant set. In other words, if the data come from a

polynomial of total degree less than or equal to $m - 1$, then they are fitted exactly by the expansion (6.2).

In general, solving the interpolation problem based on the extended expansion (6.2) now amounts to solving a system of linear equations of the form

$$\begin{bmatrix} A & P \\ P^T & O \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix}, \quad (6.3)$$

where the pieces are given by $A_{jk} = \varphi(\|\mathbf{x}_j - \mathbf{x}_k\|)$, $j, k = 1, \dots, N$, $P_{jl} = p_l(\mathbf{x}_j)$, $j = 1, \dots, N$, $l = 1, \dots, M$, $\mathbf{c} = [c_1, \dots, c_N]^T$, $\mathbf{d} = [d_1, \dots, d_M]^T$, $\mathbf{y} = [y_1, \dots, y_N]^T$, $\mathbf{0}$ is a zero vector of length M , and O is an $M \times M$ zero matrix. Below we will study the invertibility of this matrix in two steps. First for the case $m = 1$ in Theorem 6.2, and then for the case of general m in Theorem 7.2.

Note that we can easily modify the MATLAB program listed above to deal with reproduction of polynomials of other degrees. For example, if we want to reproduce constants then we need to replace lines 9, 12, 13, and 16 by

```

9  rhs = [rhs; 0];
12 PM = ones(N,1);
13 IM = [IM PM; [PM' 0]];
16 PM = ones(M,1); EM = [EM PM];

```

and for reproduction of bivariate quadratic polynomials we can use

```

9  rhs = [rhs; zeros(6,1)];
12a PM = [ones(N,1) dsites dsites(:,1).^2 ...
12b           dsites(:,2).^2 dsites(:,1).*dsites(:,2)];
13  IM = [IM PM; [PM' zeros(6,6)]]; 
16a PM = [ones(M,1) epoints epoints(:,1).^2 ...
16b           epoints(:,2).^2 epoints(:,1).*epoints(:,2)];
16c EM = [EM PM];

```

Of course, these specific examples work only for the case $s = 2$. The generalization to higher dimensions, however, is obvious but more cumbersome.

6.4 Conditionally Positive Definite Matrices and Reproduction of Constant Functions

We now need to investigate whether the augmented system matrix in (6.3) is non-singular. The special case $m = 1$ (in any space dimension s), *i.e.*, reproduction of constants, is covered by standard results from linear algebra, and we discuss it first.

Definition 6.2. A real symmetric matrix A is called *conditionally positive semi-*

definite of order one if its associated quadratic form is non-negative, i.e.

$$\sum_{j=1}^N \sum_{k=1}^N c_j c_k A_{jk} \geq 0 \quad (6.4)$$

for all $\mathbf{c} = [c_1, \dots, c_N]^T \in \mathbb{R}^N$ that satisfy

$$\sum_{j=1}^N c_j = 0.$$

If $\mathbf{c} \neq \mathbf{0}$ implies strict inequality in (6.4) then A is called *conditionally positive definite of order one*.

In the linear algebra literature the definition usually is formulated using “ \leq ” in (6.4), and then A is referred to as (conditionally or almost) *negative* definite. Obviously, conditionally positive definite matrices of order one exist only for $N > 1$.

We can interpret a matrix A that is conditionally positive definite of order one as one that is positive definite on the space of vectors \mathbf{c} such that

$$\sum_{j=1}^N c_j = 0.$$

Thus, in this sense, A is positive definite on the space of vectors \mathbf{c} “perpendicular” to constant functions.

Now we are ready to formulate and prove

Theorem 6.2. *Let A be a real symmetric $N \times N$ matrix that is conditionally positive definite of order one, and let $P = [1, \dots, 1]^T$ be an $N \times 1$ matrix (column vector). Then the system of linear equations*

$$\begin{bmatrix} A & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ d \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix},$$

is uniquely solvable.

Proof. Assume $[\mathbf{c}, d]^T$ is a solution of the homogeneous linear system, i.e., with $\mathbf{y} = \mathbf{0}$. We show that $[\mathbf{c}, d]^T = \mathbf{0}^T$ is the only possible solution.

Multiplication of the top block of the (homogeneous) linear system by \mathbf{c}^T yields

$$\mathbf{c}^T A \mathbf{c} + d \mathbf{c}^T P = 0.$$

From the bottom block of the system we know $P^T \mathbf{c} = \mathbf{c}^T P = 0$, and therefore

$$\mathbf{c}^T A \mathbf{c} = 0.$$

Since the matrix A is conditionally positive definite of order one by assumption we get that $\mathbf{c} = \mathbf{0}$. Finally, the top block of the homogeneous linear system under consideration states that

$$A\mathbf{c} + dP = \mathbf{0},$$

so that $\mathbf{c} = \mathbf{0}$ and the fact that P is a vector of ones imply $d = 0$. \square

Since Gaussians (and any other strictly positive definite radial function) give rise to positive definite matrices, and since positive definite matrices are also conditionally positive definite of order one, Theorem 6.2 establishes the nonsingularity of the (augmented) radial basis function interpolation matrix for constant reproduction.

In order to cover radial basis function interpolation with reproduction of higher-order polynomials we will now introduce (strictly) conditionally positive definite functions of order m .

Chapter 7

Conditionally Positive Definite Functions

7.1 Conditionally Positive Definite Functions Defined

In analogy to our earlier discussion of interpolation with positive definite functions we will now introduce conditionally positive definite and strictly conditionally positive definite functions of order m . We will realize that these functions provide the natural generalization of RBF interpolation with polynomial reproduction discussed in the previous chapter. Examples of strictly conditionally positive definite (radial) functions are presented in the next chapter.

Definition 7.1. A complex-valued continuous function Φ is called *conditionally positive definite of order m on \mathbb{R}^s* if

$$\sum_{j=1}^N \sum_{k=1}^N c_j \overline{c_k} \Phi(\mathbf{x}_j - \mathbf{x}_k) \geq 0 \quad (7.1)$$

for any N pairwise distinct points $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^s$, and $\mathbf{c} = [c_1, \dots, c_N]^T \in \mathbb{C}^N$ satisfying

$$\sum_{j=1}^N c_j p(\mathbf{x}_j) = 0,$$

for any complex-valued polynomial p of degree at most $m - 1$. The function Φ is called *strictly conditionally positive definite of order m on \mathbb{R}^s* if the quadratic form (7.1) is zero only for $\mathbf{c} \equiv \mathbf{0}$.

An immediate observation is

Lemma 7.1. *A function that is (strictly) conditionally positive definite of order m on \mathbb{R}^s is also (strictly) conditionally positive definite of any higher order. In particular, a (strictly) positive definite function is always (strictly) conditionally positive definite of any order.*

Proof. The first statement follows immediately from Definition 7.1. The second statement is true since the case $m = 0$ yields the class of (strictly) positive definite functions, i.e., (strictly) conditionally positive definite functions of order zero are (strictly) positive definite. \square

As for positive definite functions earlier, we can restrict ourselves to real-valued, even functions Φ and real coefficients. A detailed discussion is presented in [Wendland (2005a)].

Theorem 7.1. *A real-valued continuous even function Φ is called conditionally positive definite of order m on \mathbb{R}^s if*

$$\sum_{j=1}^N \sum_{k=1}^N c_j c_k \Phi(\mathbf{x}_j - \mathbf{x}_k) \geq 0 \quad (7.2)$$

for any N pairwise distinct points $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^s$, and $\mathbf{c} = [c_1, \dots, c_N]^T \in \mathbb{R}^N$ satisfying

$$\sum_{j=1}^N c_j p(\mathbf{x}_j) = 0,$$

for any real-valued polynomial p of degree at most $m - 1$. The function Φ is called strictly conditionally positive definite of order m on \mathbb{R}^s if the quadratic form (7.2) is zero only for $\mathbf{c} \equiv \mathbf{0}$.

The matrix A with entries $A_{jk} = \Phi(\mathbf{x}_j - \mathbf{x}_k)$ corresponding to a real and even strictly conditionally positive definite function Φ of order m can also be interpreted as being positive definite on the space of vectors \mathbf{c} such that

$$\sum_{j=1}^N c_j p(\mathbf{x}_j) = 0, \quad p \in \Pi_{m-1}^s.$$

Thus, in this sense, A is positive definite on the space of vectors \mathbf{c} “perpendicular” to s -variate polynomials of degree at most $m - 1$.

We can now generalize the interpolation Theorem 6.2 to the case of general polynomial reproduction:

Theorem 7.2. *If the real-valued even function Φ is strictly conditionally positive definite of order m on \mathbb{R}^s and the points $\mathbf{x}_1, \dots, \mathbf{x}_N$ form an $(m - 1)$ -unisolvant set, then the system of linear equations (6.3) is uniquely solvable.*

Proof. The proof is almost identical to the proof of Theorem 6.2. Assume $[\mathbf{c}, \mathbf{d}]^T$ is a solution of the homogeneous linear system, i.e., with $\mathbf{y} = \mathbf{0}$. We show that $[\mathbf{c}, \mathbf{d}]^T = \mathbf{0}$ is the only possible solution.

Multiplication of the top block by \mathbf{c}^T yields

$$\mathbf{c}^T \mathbf{A} \mathbf{c} + \mathbf{c}^T \mathbf{P} \mathbf{d} = 0.$$

From the bottom block of (6.3) we know $\mathbf{P}^T \mathbf{c} = \mathbf{0}$. This implies $\mathbf{c}^T \mathbf{P} = \mathbf{0}^T$, and therefore

$$\mathbf{c}^T \mathbf{A} \mathbf{c} = 0. \quad (7.3)$$

Since the function Φ is strictly conditionally positive definite of order m by assumption we know that the quadratic form of A (with coefficients such that $\mathbf{P}^T \mathbf{c} = \mathbf{0}$) above is zero only for $\mathbf{c} = \mathbf{0}$. Therefore (7.3) tells us that $\mathbf{c} = \mathbf{0}$. The unisolvency of the data sites, i.e., the linear independence of the columns of P (c.f. Remark 6.1), and the fact that $\mathbf{c} = \mathbf{0}$ guarantee $\mathbf{d} = \mathbf{0}$ from the top block

$$\mathbf{A} \mathbf{c} + \mathbf{P} \mathbf{d} = \mathbf{0}$$

of (6.3). \square

7.2 Conditionally Positive Definite Functions and Generalized Fourier Transforms

As before, integral characterizations help us identify functions that are strictly conditionally positive definite of order m on \mathbb{R}^s . An integral characterization of conditionally positive definite functions of order m , i.e., a generalization of Bochner’s theorem, can be found in the paper [Sun (1993b)]. However, since the subject matter is rather complicated, and since it does not really help us solve the scattered data interpolation problem, we do not mention any details here.

The Fourier transform characterization of strictly conditionally positive definite functions of order m on \mathbb{R}^s also makes use of some advanced tools from analysis. However, since this characterization is relevant for our purposes we state the result (due to [Iske (1994)]) and collect some of the most relevant concepts from distribution theory in Appendix B.

This distributional approach originated in the manuscript [Madyach and Nelson (1983)]. Many more details can be found in the original papers mentioned above as well as in the book [Wendland (2005a)].

Theorem 7.3. *Suppose the complex-valued function $\Phi \in \mathcal{B}$ possesses a generalized Fourier transform $\hat{\Phi}$ of order m which is continuous on $\mathbb{R}^s \setminus \{\mathbf{0}\}$. Then Φ is strictly conditionally positive definite of order m if and only if $\hat{\Phi}$ is non-negative and non-vanishing.*

Theorem 7.3 states that strictly conditionally positive definite functions on \mathbb{R}^s are characterized by the order of the singularity of their generalized Fourier transform at the origin, provided that this generalized Fourier transform is non-negative and non-zero.

Since integral characterizations similar to Schoenberg’s Theorems 3.6 and 3.8 are so complicated in the conditionally positive definite case we do not pursue the concept of a conditionally positive definite radial function here. The interested reader is referred to [Guo et al. (1993a)] for details. We will discuss some examples of radial functions via the Fourier transform approach in the next chapter, and in Chapter 9 we will explore the connection between completely and multiply monotone functions and conditionally positive definite radial functions.

Chapter 8

Examples of Conditionally Positive Definite Functions

We now present a number of strictly conditionally positive definite (radial) functions that are covered by the Fourier transform characterization Theorem 7.3. The generalized Fourier transforms for these examples are explicitly computed in [Wendland (2005a)]. We will establish the strict conditional positive definiteness of these functions again in detail in the next chapter with the help of completely monotone functions. Included in the examples below are several of the best known radial basic functions such as the multiquadric due to [Hardy (1971)] and the thin plate spline due to [Duchon (1976)].

8.1 Example 1: Generalized Multiquadratics

The *generalized multiquadratics*

$$\Phi(\mathbf{x}) = (1 + \|\mathbf{x}\|^2)^\beta, \quad \mathbf{x} \in \mathbb{R}^s, \beta \in \mathbb{R} \setminus \mathbb{N}_0, \quad (8.1)$$

have generalized Fourier transforms

$$\hat{\Phi}(\omega) = \frac{2^{1+\beta}}{\Gamma(-\beta)} \|\omega\|^{-\beta-s/2} K_{\beta+s/2}(\|\omega\|), \quad \omega \neq 0,$$

of order $m = \max(0, \lceil \beta \rceil)$, where $\lceil \beta \rceil$ denotes the smallest integer greater than or equal to β . Here the K_ν are again the modified Bessel functions of the second kind of order ν (*c.f.* Section 4.5). Note that we need to exclude positive integer values of β since this would lead to polynomials of even degree (see the related discussion in Example 2 below).

Since the generalized Fourier transforms are positive with a singularity of order m at the origin, Theorem 7.3 tells us that the functions

$$\Phi(\mathbf{x}) = (-1)^{\lceil \beta \rceil} (1 + \|\mathbf{x}\|^2)^\beta, \quad 0 < \beta \notin \mathbb{N},$$

are strictly conditionally positive definite of order $m = \lceil \beta \rceil$ (and higher).

For $\beta < 0$ the Fourier transform is a classical one and we are back to the generalized inverse multiquadratics of Section 4.5. These functions are again shown to be strictly conditionally positive definite of order $m = 0$, *i.e.*, strictly positive definite.

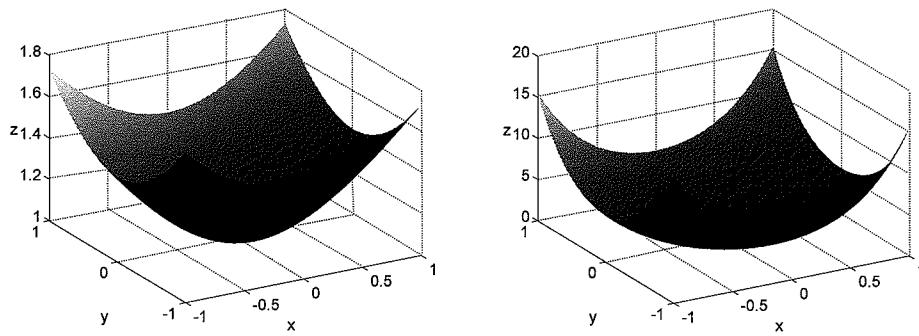


Fig. 8.1 Hardy's multiquadric with $\beta = \frac{1}{2}$ (left) and a generalized multiquadric with $\beta = \frac{5}{2}$ (right) centered at the origin in \mathbb{R}^2 .

Figure 8.1 shows Hardy's “original” multiquadric (with $\beta = 1/2$, i.e., strictly conditionally positive definite of order 1) and a generalized multiquadric with $\beta = 5/2$ (i.e., strictly conditionally positive definite of order 3). Note that the generalized multiquadratics are no longer “bump” functions (as most of the strictly positive definite functions were), but functions that grow with the distance from the origin.

The arguments above together with Theorem 7.2 show that we can use Hardy's multiquadratics in the form

$$\mathcal{P}_f(\mathbf{x}) = \sum_{k=1}^N c_k \sqrt{1 + \|\mathbf{x} - \mathbf{x}_k\|^2} + d, \quad \mathbf{x} \in \mathbb{R}^s,$$

together with the constraint

$$\sum_{k=1}^N c_k = 0$$

to solve the scattered data interpolation problem. The resulting interpolant will be exact for constant data. As in our earlier discussions we can scale the basis functions with a shape parameter ε by replacing $\|\mathbf{x}\|$ by $|\varepsilon| \|\mathbf{x}\|$. This does not affect the well-posedness of the interpolation problem. However, a small value of ε gives rise to “flat” basis functions, whereas a large value of ε produces very steep functions. As before, the accuracy of the fit will improve with decreasing ε while the stability will decrease, and the numerical results will become increasingly less reliable. For Figure 8.1 we used the shape parameter $\varepsilon = 1$.

By Theorem 9.7 below we can also solve the scattered data interpolation problem using the simpler expansion

$$\mathcal{P}_f(\mathbf{x}) = \sum_{k=1}^N c_k \sqrt{1 + \|\mathbf{x} - \mathbf{x}_k\|^2}, \quad \mathbf{x} \in \mathbb{R}^s.$$

This is what Hardy proposed to do in his work in the early 1970s (see, e.g., [Hardy (1971)]).

8.2 Example 2: Radial Powers

The *radial powers*

$$\Phi(\mathbf{x}) = \|\mathbf{x}\|^\beta, \quad \mathbf{x} \in \mathbb{R}^s, \quad 0 < \beta \notin 2\mathbb{N}, \quad (8.2)$$

have generalized Fourier transforms

$$\hat{\Phi}(\boldsymbol{\omega}) = \frac{2^{\beta+s/2} \Gamma(\frac{s+\beta}{2})}{\Gamma(-\beta/2)} \|\boldsymbol{\omega}\|^{-\beta-s}, \quad \boldsymbol{\omega} \neq 0,$$

of order $m = \lceil \beta/2 \rceil$. Therefore, the functions

$$\Phi(\mathbf{x}) = (-1)^{\lceil \beta/2 \rceil} \|\mathbf{x}\|^\beta, \quad 0 < \beta \notin 2\mathbb{N},$$

are strictly conditionally positive definite of order $m = \lceil \beta/2 \rceil$ (and higher).

This shows that the basic function $\Phi(\mathbf{x}) = \|\mathbf{x}\|_2$ used for the distance matrix fits in the introductory chapter are strictly conditionally positive definite of order one. According to Theorem 7.2 we should have used these basic functions together with an appended constant. However, Theorem 9.7 below provides the justification for their use as a pure distance matrix.

In Figure 8.2 we show radial cubics ($\beta = 3$, i.e., strictly conditionally positive definite of order 2) and quintics ($\beta = 5$, i.e., strictly conditionally positive definite of order 3).

Note that we had to exclude even powers in (8.2). This is clear since an even power combined with the square root in the definition of the Euclidean norm results in a polynomial — and we have already decided that polynomials cannot be used for interpolation at arbitrarily scattered multivariate sites.

Note that radial powers are not affected by a scaling of their argument. In other words, radial powers are *shape parameter free*. This has the advantage that the user need not worry about finding a “good” value of ε . On the other hand, we will see below that radial powers will not be able to achieve the spectral convergence rates that are possible with some of the other basic functions such as Gaussians and generalized (inverse) multiquadratics.

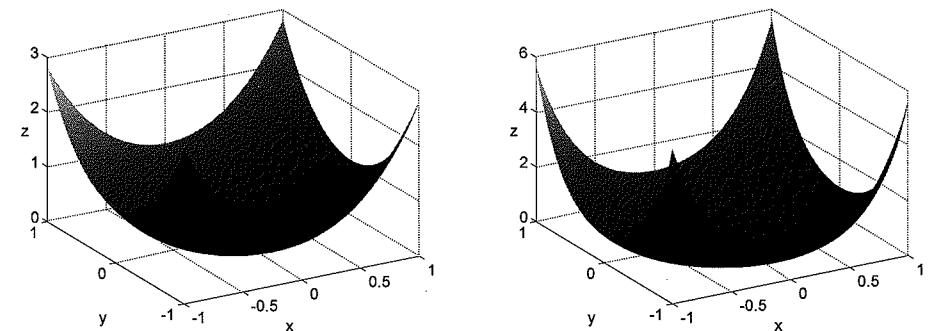


Fig. 8.2 Radial cubic (left) and quintic (right) centered at the origin in \mathbb{R}^2 .

8.3 Example 3: Thin Plate Splines

In the previous example we had to rule out even powers. However, if the even radial powers are multiplied by a log term, then we are back in business.

Duchon's *thin plate splines* (or Meinguet's *surface splines*)

$$\Phi(\mathbf{x}) = \|\mathbf{x}\|^{2\beta} \log \|\mathbf{x}\|, \quad \mathbf{x} \in \mathbb{R}^s, \beta \in \mathbb{N}, \quad (8.3)$$

have generalized Fourier transforms

$$\hat{\Phi}(\boldsymbol{\omega}) = (-1)^{\beta+1} 2^{2\beta-1+s/2} \Gamma(\beta + s/2) \beta! \|\boldsymbol{\omega}\|^{-s-2\beta}$$

of order $m = \beta + 1$. Therefore, the functions

$$\Phi(\mathbf{x}) = (-1)^{\beta+1} \|\mathbf{x}\|^{2\beta} \log \|\mathbf{x}\|, \quad \beta \in \mathbb{N},$$

are strictly conditionally positive definite of order $m = \beta + 1$. In particular, we can use

$$\mathcal{P}_f(\mathbf{x}) = \sum_{k=1}^N c_k \|\mathbf{x} - \mathbf{x}_k\|^2 \log \|\mathbf{x} - \mathbf{x}_k\| + d_1 + d_2 x + d_3 y, \quad \mathbf{x} = (x, y) \in \mathbb{R}^2,$$

together with the constraints

$$\sum_{k=1}^N c_k = 0, \quad \sum_{k=1}^N c_k x_k = 0, \quad \sum_{k=1}^N c_k y_k = 0,$$

to solve the scattered data interpolation problem in \mathbb{R}^2 provided the data sites are not all collinear. The resulting interpolant will be exact for data coming from a bivariate linear function.

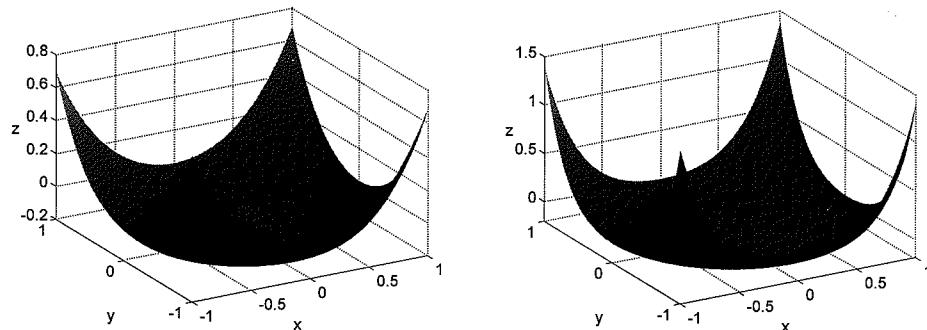


Fig. 8.3 “Classical” thin plate spline (left) and order 3 thin plate spline (right) centered at the origin in \mathbb{R}^2 .

Figure 8.3 shows the “classical” thin plate spline (with $\beta = 1$, i.e., strictly conditionally positive definite of order 2) and the order 3 spline $\Phi(\mathbf{x}) = \|\mathbf{x}\|^4 \log \|\mathbf{x}\|$. Note that the thin plate spline basic functions are not monotone. Also, both graphs displayed in Figure 8.3 contain a portion with negative function values.

As with radial powers, use of a shape parameter ε in conjunction with thin plate splines is pointless. Finally, we note that the families of radial powers and thin plate splines are often referred to collectively as *polyharmonic splines*.

There is no result that states that interpolation with thin plate splines (or any other strictly conditionally positive definite function of order $m \geq 2$) without the addition of an appropriate degree $m - 1$ polynomial is well-posed. Theorem 9.7 quoted several times before covers only the case $m = 1$.

Chapter 9

Conditionally Positive Definite Radial Functions

As for strictly positive definite radial functions, we will be able to connect strictly conditionally positive definite radial functions to completely monotone and multiply monotone functions, and thus be able to obtain a criterion for checking conditional positive definiteness of radial functions that is easier to use than the generalized Fourier transform in the previous chapters.

9.1 Conditionally Positive Definite Radial Functions and Completely Monotone Functions

In analogy to the discussion in Section 3.3 we now focus on conditionally positive definite functions that are radial on \mathbb{R}^s for all s . The paper [Guo *et al.* (1993a)] by Guo, Hu and Sun contains an integral characterization for such functions. This characterization is too technical to be included here.

Another important result in [Guo *et al.* (1993a)] is a characterization of conditionally positive definite radial functions on \mathbb{R}^s for all s in terms of completely monotone functions.

Theorem 9.1. *Let $\varphi \in C[0, \infty) \cap C^\infty(0, \infty)$. Then the function $\Phi = \varphi(\|\cdot\|^2)$ is conditionally positive definite of order m and radial on \mathbb{R}^s for all s if and only if $(-1)^m \varphi^{(m)}$ is completely monotone on $(0, \infty)$.*

Proof. The fact that complete monotonicity implies conditional positive definiteness was proved in [Micchelli (1986)]. Micchelli also conjectured that the converse holds and gave a simple proof for this in the case $m = 1$. For $m = 0$ this is Schoenberg's characterization of positive definite radial functions on \mathbb{R}^s for all s in terms of completely monotone functions (Theorem 5.2). The remaining part of the theorem is shown in [Guo *et al.* (1993a)]. \square

In order to get strict conditional positive definiteness we need to generalize Theorem 5.3, *i.e.*, the fact that φ not be constant. This leads to (see [Wendland (2005a)])

Theorem 9.2. If φ is as in Theorem 9.1 and not a polynomial of degree at most m , then Φ is strictly conditionally positive definite of order m and radial on \mathbb{R}^s for all s .

We can now more easily verify the conditional positive definiteness of the functions listed in the previous chapter.

Example 9.1. The functions

$$\varphi(r) = (-1)^{\lceil \beta \rceil} (1+r)^\beta, \quad 0 < \beta \notin \mathbb{N}$$

imply

$$\varphi^{(\ell)}(r) = (-1)^{\lceil \beta \rceil} \beta(\beta-1)\cdots(\beta-\ell+1)(1+r)^{\beta-\ell}$$

so that

$$(-1)^{\lceil \beta \rceil} \varphi^{(\lceil \beta \rceil)}(r) = \beta(\beta-1)\cdots(\beta-\lceil \beta \rceil+1)(1+r)^{\beta-\lceil \beta \rceil}$$

is completely monotone. Moreover, $m = \lceil \beta \rceil$ is the smallest possible m such that $(-1)^m \varphi^{(m)}$ is completely monotone. Since $\beta \notin \mathbb{N}$ we know that φ is not a polynomial, and therefore the generalized multiquadratics (c.f. (8.1))

$$\Phi(\|\mathbf{x}\|) = (-1)^{\lceil \beta \rceil} (1 + \|\mathbf{x}\|^2)^\beta, \quad \beta > 0,$$

are strictly conditionally positive definite of order $m \geq \lceil \beta \rceil$ and radial on \mathbb{R}^s for all values of s .

Example 9.2. The functions

$$\varphi(r) = (-1)^{\lceil \beta/2 \rceil} r^{\beta/2}, \quad 0 < \beta \notin 2\mathbb{N},$$

imply

$$\varphi^{(\ell)}(r) = (-1)^{\lceil \beta/2 \rceil} \frac{\beta}{2} \left(\frac{\beta}{2} - 1 \right) \cdots \left(\frac{\beta}{2} - \ell + 1 \right) r^{\beta/2-\ell}$$

so that $(-1)^{\lceil \beta/2 \rceil} \varphi^{(\lceil \beta/2 \rceil)}$ is completely monotone and $m = \lceil \beta/2 \rceil$ is the smallest possible m such that $(-1)^m \varphi^{(m)}$ is completely monotone. Since β is not an even integer φ is not a polynomial, and therefore, the radial powers (c.f. (8.2))

$$\Phi(\|\mathbf{x}\|) = (-1)^{\lceil \beta/2 \rceil} \|\mathbf{x}\|^\beta, \quad \beta > 0, \beta \notin 2\mathbb{N},$$

are strictly conditionally positive definite of order $m \geq \lceil \beta/2 \rceil$ and radial on \mathbb{R}^s for all s .

Example 9.3. The thin plate splines (c.f. (8.3))

$$\Phi(\|\mathbf{x}\|) = (-1)^{\beta+1} \|\mathbf{x}\|^{2\beta} \log \|\mathbf{x}\|, \quad \beta \in \mathbb{N},$$

are strictly conditionally positive definite of order $m \geq \beta + 1$ and radial on \mathbb{R}^s for all s . To see this we observe that

$$2\Phi(\|\mathbf{x}\|) = (-1)^{\beta+1} \|\mathbf{x}\|^{2\beta} \log(\|\mathbf{x}\|^2).$$

Therefore, we let

$$\varphi(r) = (-1)^{\beta+1} r^\beta \log r, \quad \beta \in \mathbb{N},$$

which is obviously not a polynomial. Differentiating φ we get

$$\varphi^{(\ell)}(r) = (-1)^{\beta+1} \beta(\beta-1)\cdots(\beta-\ell+1) r^{\beta-\ell} \log r + p_\ell(r), \quad 1 \leq \ell \leq \beta,$$

with p_ℓ a polynomial of degree $\beta - \ell$. Therefore, taking $\ell = \beta$ we have

$$\varphi^{(\beta)}(r) = (-1)^{\beta+1} \beta! \log r + C$$

and

$$\varphi^{(\beta+1)}(r) = (-1)^{\beta+1} \frac{\beta!}{r},$$

which is completely monotone on $(0, \infty)$.

9.2 Conditionally Positive Definite Radial Functions and Multiply Monotone Functions

Finally, [Micchelli (1986)] proved a more general version of Theorem 5.5 relating conditionally positive definite radial functions of order m on \mathbb{R}^s (for some fixed value of s) and multiply monotone functions. We state a stronger version due to [Buhmann (1993a)] which ensures strict conditional positive definiteness.

Theorem 9.3. Let $k = \lfloor s/2 \rfloor - m + 2$ be a positive integer, and suppose $\varphi \in C^{m-1}[0, \infty)$ is not a polynomial of degree at most m . If $(-1)^m \varphi^{(m)}$ is k -times monotone on $(0, \infty)$ but not constant, then φ is strictly conditionally positive definite of order m and radial on \mathbb{R}^s for any s such that $\lfloor s/2 \rfloor \leq k + m - 2$.

Just as we showed earlier that compactly supported radial functions cannot be strictly positive definite on \mathbb{R}^s for all s , it is important to note that there are no truly conditionally positive definite functions with compact support. More precisely (see [Wendland (2005a)]),

Theorem 9.4. Assume that the complex-valued function $\Phi \in C(\mathbb{R}^s)$ has compact support. If Φ is strictly conditionally positive definite of (minimal) order m , then m is necessarily zero, i.e., Φ is already strictly positive definite.

Proof. The hypotheses on Φ ensure that it is integrable, and therefore it possesses a classical Fourier transform $\hat{\Phi}$ which is continuous. For integrable functions the generalized Fourier transform coincides with the classical Fourier transform. Theorem 7.3 ensures that $\hat{\Phi}$ is non-negative on $\mathbb{R}^s \setminus \{\mathbf{0}\}$ and not identically equal to zero. By continuity we also get $\hat{\Phi}(\mathbf{0}) \geq 0$, and Theorem 3.5 shows that Φ is strictly positive definite. \square

Theorem 9.3 together with Theorem 9.4 implies that if we perform m -fold anti-differentiation on a non-constant k -times monotone function, then we obtain a function that is strictly positive definite and radial on \mathbb{R}^s for $\lfloor s/2 \rfloor \leq k + m - 2$.

Example 9.4. The function $\varphi_k(r) = (1 - r)_+^k$ is k -times monotone (see Example 5.5 in Section 5.2). To avoid the integration constant for the compactly supported truncated power function we compute the anti-derivative via the integral operator I of Example 5.6 in Section 5.2, i.e.,

$$I\varphi_k(r) = \int_r^\infty \varphi_k(t) dt = \int_r^\infty (1 - t)_+^k dt = \frac{(-1)^k}{k+1} (1 - r)_+^{k+1}.$$

If we apply m -fold anti-differentiation we get

$$I^m \varphi_k(r) = II^{m-1} \varphi_k(r) = \frac{(-1)^{mk}}{(k+1)(k+2)\cdots(k+m)} (1 - r)_+^{k+m}.$$

Therefore, by Theorem 9.3, the function

$$\varphi(r) = (1 - r)_+^{k+m}$$

is strictly conditionally positive definite of order m and radial on \mathbb{R}^s for $\lfloor s/2 \rfloor \leq k + m - 2$, and by Theorem 9.4 it is even strictly positive definite and radial on \mathbb{R}^s . This was also observed in Example 6 of Chapter 4. In fact, we saw there that φ is strictly positive definite and radial on \mathbb{R}^s for $\lfloor s/2 \rfloor \leq k + m - 1$.

We see that we can construct strictly positive definite compactly supported radial functions by anti-differentiating the truncated power function. This is essentially the approach taken by Wendland to construct his popular compactly supported radial basis functions. We provide more details of his construction in Chapter 11.

9.3 Some Special Properties of Conditionally Positive Definite Functions of Order One

Since an $N \times N$ matrix that is conditionally positive definite of order one is positive definite on a subspace of dimension $N - 1$ it has the interesting property that at least $N - 1$ of its eigenvalues are positive. This follows immediately from the Courant-Fischer theorem of linear algebra (see e.g., p. 550 of [Meyer (2000)]):

Theorem 9.5 (Courant-Fischer). *Let A be a real symmetric $N \times N$ matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$, then*

$$\lambda_k = \max_{\dim V=k} \min_{\substack{x \in V \\ \|x\|=1}} x^T A x$$

and

$$\lambda_k = \min_{\dim V=N-k+1} \max_{\substack{x \in V \\ \|x\|=1}} x^T A x.$$

With an additional assumption on A we can make an even stronger statement.

Theorem 9.6. *An $N \times N$ matrix A which is conditionally positive definite of order one and has a non-positive trace possesses one negative and $N - 1$ positive eigenvalues.*

Proof. Let $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$ denote the eigenvalues of A . From the Courant-Fischer theorem we get

$$\lambda_{N-1} = \max_{\dim V=N-1} \min_{\substack{x \in V \\ \|x\|=1}} x^T A x \geq \min_{\substack{\mathbf{c} : \sum c_k = 0 \\ \|\mathbf{c}\|=1}} \mathbf{c}^T A \mathbf{c} > 0,$$

so that A has at least $N - 1$ positive eigenvalues. But since $\text{tr}(A) = \sum_{k=1}^N \lambda_k \leq 0$, A also must have at least one negative eigenvalue. \square

Note that the additional hypothesis of Theorem 9.6 is satisfied for the interpolation matrix resulting from (the negative) of RBFs such as Hardy's multiquadric or the linear radial function $\varphi(r) = r$ since its diagonal elements correspond to the value of the basic function at the origin.

Moreover, we will now use Theorem 9.6 to conclude that we can use radial functions that are strictly conditionally positive definite of order one (such as the multiquadric, $0 < \beta < 1$, and the norm basic function) *without* appending the constant term to solve the scattered data interpolation problem. This was first proved by [Micchelli (1986)] and motivated by Hardy's earlier work with multiquadratics and Franke's conjecture that the matrix A is non-singular in this case (see [Franke (1982a)]).

Theorem 9.7 (Interpolation). *Suppose Φ is strictly conditionally positive definite of order one and that $\Phi(\mathbf{0}) \leq 0$. Then for any distinct points $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^s$ the matrix A with entries $A_{jk} = \Phi(\mathbf{x}_j - \mathbf{x}_k)$ has $N - 1$ positive and one negative eigenvalue, and is therefore non-singular.*

Proof. Clearly, the matrix A is conditionally positive definite of order one. Moreover, the trace of A is given by $\text{tr}(A) = N\Phi(\mathbf{0}) \leq 0$. Therefore, Theorem 9.6 applies and the statement follows. \square

As mentioned above, this theorem covers the generalized multiquadratics $\Phi(\mathbf{x}) = -(1 + \|\mathbf{x}\|)^\beta$ with $0 < \beta < 1$ (which includes the Hardy multiquadric). The theorem also covers the radial powers $\Phi(\mathbf{x}) = -\|\mathbf{x}\|^\beta$ for $0 < \beta < 2$ (including the Euclidean distance function).

Another special property of a conditionally positive definite function of order one is

Lemma 9.1. *If C is an arbitrary real constant and the real even function Φ is (strictly) conditionally positive definite of order one, then $\Phi + C$ is also (strictly) conditionally positive definite of order one.*

Proof. Simply consider

$$\sum_{j=1}^N \sum_{k=1}^N c_j c_k [\Phi(\mathbf{x}_j - \mathbf{x}_k) + C] = \sum_{j=1}^N \sum_{k=1}^N c_j c_k \Phi(\mathbf{x}_j - \mathbf{x}_k) + \sum_{j=1}^N \sum_{k=1}^N c_j c_k C.$$

The second term on the right is zero since Φ is conditionally positive definite of order one, i.e., $\sum_{j=1}^N c_j = 0$, and thus the statement follows. \square

Chapter 10

Miscellaneous Theory: Other Norms and Scattered Data Fitting on Manifolds

10.1 Conditionally Positive Definite Functions and p -Norms

In Chapter 1 we used interpolation with distance matrices as a multivariate generalization of the piecewise linear approach. Our choice of the distance matrix approach was motivated by the fact that the associated basis functions, $\Phi_j(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_j\|$ would satisfy the dependence on the data sites imposed on a multivariate interpolation method by the Mairhuber-Curtis theorem. We made the (natural?) choice of using the Euclidean (2-norm) distance function, and then showed in subsequent chapters that the function $\Phi(\mathbf{x}) = -\|\mathbf{x}\|_2$ is strictly conditionally positive definite of order one and radial on \mathbb{R}^s , and thus our distance matrix approach was indeed well-posed via Micchelli's Theorem 9.7.

We now briefly consider solving the scattered data interpolation problem with radial functions based on other p -norms. These norms are defined as usual as

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^s |x_i|^p \right)^{1/p}, \quad \mathbf{x} \in \mathbb{R}^s, \quad 1 \leq p < \infty.$$

The content of this section is mostly the subject of the paper [Baxter (1991)].

If we consider only distance matrices, i.e., interpolation matrices generated by the basic function $\Phi(\mathbf{x}) = \|\mathbf{x}\|_p$, then it was shown in [Dyn *et al.* (1989)] that the choice $p = 1$ leads to a singular matrix already for very simple sets of distinct interpolation points. For example, if $\mathcal{X} = \{(0, 0), (1, 0), (1, 1), (0, 1)\}$ then the 1-norm distance matrix is given by

$$\begin{bmatrix} 0 & 1 & 2 & 1 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 \end{bmatrix},$$

and it is easy to verify that this matrix is singular. This result has discouraged people from using 1-norm radial basis functions.

However, if we use, e.g., N Halton points, then we have never encountered a singular 1-norm distance matrix in all of our numerical experiments. In fact, the

matrix seems to have $N - 1$ negative and one positive eigenvalue (just as predicted by Theorem 9.7 for the 2-norm case).

Figure 10.2 shows various interpolants to the linear function $f(x, y) = (x + y)/2$ on the unit square. The interpolant is false colored according to the maximum error. In the top row of the figure we used a 1-norm distance matrix based on 1089 Halton points. The MATLAB code for generating a p -norm distance matrix fit is virtually identical to our earlier code in Programs 1.1 and 1.2. The only change required is the replacement of lines 6 and 8 of Program 1.1 by

```
6 DM = DM + abs(dr-cc).^p;
8 DM = DM.^ (1/p);
```

We can also use this modification of Program 1.1 to produce more general RBF interpolants (see the example with p -norm Gaussians in the bottom row of Figure 10.2 below).

Similar to the 1-norm result from [Dyn *et al.* (1989)] quoted above it was shown in [Baxter (1991)] that for $p > 2$ we cannot in general guarantee non-singular distance matrices, either. On the other hand, a number of numerical experiments showed the p -norm matrices to be non-singular provided uniformly spaced or Halton points in $[0, 1]^2$ were used. The second row of Figure 10.2 shows distance matrix interpolants to $f(x, y) = (x + y)/2$ on the unit square using a p -norm distance matrix for $p = 10$ and $p = 100$ based on 25 uniformly spaced points.

These examples show that certainly not all is lost when using p -norm radial basis functions. The situation is similar as with the use of Kansa's method for the collocation solution of elliptic PDEs (see Chapter 38). There do exist configurations of data points for which the interpolation matrix becomes singular. However, these configurations may be rare, and therefore the use of p -norm radial basis functions may be justified in many cases. We point out that we used norms for $p > 2$ even though the Baxter result mentioned above guarantees existence of data sets \mathcal{X} for which the interpolation matrix will be singular. For our examples the interpolation matrix was far from singular. Using 25 uniformly spaced data sites the matrices again exhibited 24 negative and one positive eigenvalue. This use of p -norm radial basis functions certainly deserves further investigation.

The case $1 < p < 2$, however, is much better understood. In [Baxter (1991)] we find

Theorem 10.1. Suppose $1 < p < 2$ and let A be the p -norm distance matrix with entries

$$A_{jk} = \|\mathbf{x}_j - \mathbf{x}_k\|_p, \quad j, k = 1, \dots, N.$$

Then the matrix $-A$ is conditionally positive definite of order one. Moreover, it is strictly conditionally positive definite of order one if $N \geq 2$ and the points $\mathbf{x}_1, \dots, \mathbf{x}_N$ are distinct.

This theorem is derived from a much earlier theorem by Schoenberg relating conditionally positive definite matrices of order one and Euclidean distance matrices. When Schoenberg first studied conditionally positive definite matrices of order one this was in connection with isometric embeddings. Based on earlier work by Karl Menger [Menger (1928)] Schoenberg derived the following result characterizing certain conditionally positive definite matrices as Euclidean distance matrices (see [Schoenberg (1937)]).

Theorem 10.2 (Schoenberg-Menger). Let A be a real symmetric $N \times N$ matrix with all diagonal entries zero and all other elements positive. Then $-A$ is conditionally positive semi-definite of order one if and only if there exist N points $\mathbf{y}_1, \dots, \mathbf{y}_N \in \mathbb{R}^N$ for which

$$A_{jk} = \|\mathbf{y}_j - \mathbf{y}_k\|_2^2.$$

These points are the vertices of a simplex in \mathbb{R}^N .

In the third row of Figure 10.2 we display the interpolants to the test function $f(x, y) = (x + y)/2$ on $[0, 1]^2$ using distance matrix interpolation based on 25 equally spaced points and p -norms with $p = 1.001$ and $p = 2$. Since we use a plain distance interpolant, *i.e.*, $\Phi(\mathbf{x}) = \|\mathbf{x}\|_p$ it is remarkable that the error using the $p = 1.001$ -norm is about two orders of magnitude smaller than the next best p -norm distance matrix fit among our experiments (which we obtained for $p = 100$, *c.f.* Figure 10.2).

The use of different p -norms for different applications has not been studied carefully in the literature.

Two other results regarding interpolation with p -norm radial basis functions can also be found in the literature. In [Wendland (2005a)] we find a reference to [Zastavnyi (1993)] according to which — for space dimensions $s \geq 3$ — the only function that is positive definite and p -norm radial on \mathbb{R}^s is the zero function. Again, somewhat discouraging news. However, there is also good news. The following rather powerful theorem comes from [Baxter (1991)]. Baxter calls the matrix A of Theorem 10.2 an *almost negative definite* matrix (*c.f.* the remarks following Definition 6.2).

Theorem 10.3. Let $-A$ be an $N \times N$ matrix that is conditionally positive semi-definite of order one with all diagonal entries zero, and let $\varphi(\cdot^2)$ be a function that is conditionally positive definite of order one and radial on \mathbb{R}^s . Then the matrix defined by

$$B_{jk} = -\varphi(A_{jk}), \quad j, k = 1, \dots, N,$$

is conditionally positive semi-definite of order one. Moreover, if $N \geq 2$ and no off-diagonal elements of A vanish, then B is strictly conditionally positive definite of order one whenever $\varphi(\cdot^2)$ is strictly conditionally positive definite of order one.

Proof. By Schoenberg's Theorem 10.2 we can write $A_{jk} = \|\mathbf{y}_j - \mathbf{y}_k\|_2^2$ for appropriate points $\mathbf{y}_j \in \mathbb{R}^N$. By assumption $\varphi(\cdot^2)$ is conditionally positive definite of order one and radial, and therefore B is conditionally positive semi-definite of order one. Moreover, if $A_{jk} \neq 0$ for all off-diagonal elements, then $\mathbf{y}_1, \dots, \mathbf{y}_N$ are distinct, and therefore B is strictly conditionally positive definite of order one provided $\varphi(\cdot^2)$ is strictly conditionally positive definite of order one. \square

Since Baxter also shows that if A is a 1-norm distance matrix, then $-A$ is a conditionally positive semi-definite matrix of order one, Theorem 10.3 guarantees that we can use many "standard" radial basis functions in conjunction with the 1-norm for RBF interpolation. For example, the use of 1-norm Gaussians is justified by Theorem 10.3. In the literature one can also find an analog of Bochner's theorem for positive definite 1-norm radial functions due to [Cambanis *et al.* (1983)] (see also [Wendland (2005a)]).

Figure 10.1 shows p -norm Gaussians $\Phi(\mathbf{x}) = e^{-\varepsilon^2 \|\mathbf{x}\|_p^2}$ for $p = 1$ and $p = 10$. A shape parameter $\varepsilon = 3$ was used. Interpolants to the function $f(x, y) = (x+y)/2$ at 25 equally spaced points in $[0, 1]^2$ using these basic functions with $\varepsilon = 1$ are shown in the bottom row of Figure 10.2.

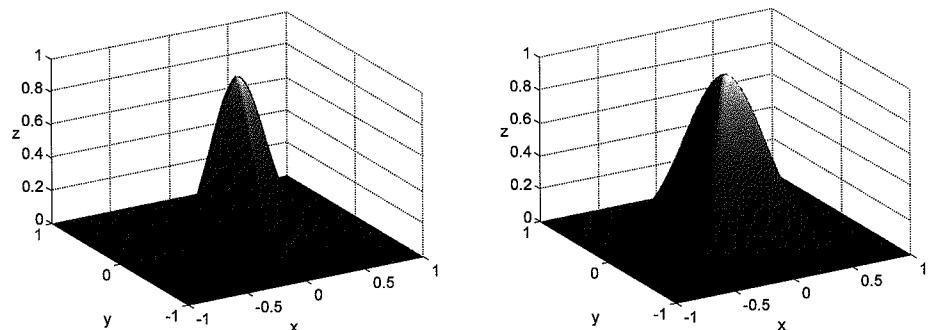


Fig. 10.1 p -norm Gaussians for $p = 1$ (left) and $p = 10$ (right) centered at the origin in \mathbb{R}^2 .

Another, closely related theorem by Baxter is

Theorem 10.4. Suppose $\varphi(\cdot^2)$ and $\psi(\cdot^2)$ are functions that are conditionally positive definite of order one and radial on \mathbb{R}^s with $\psi(0) = 0$. Then $\varphi \circ \psi(\cdot^2)$ is also conditionally positive definite of order one and radial on \mathbb{R}^s . Indeed, if $\varphi(\cdot^2)$ is strictly conditionally positive definite of order one and radial and ψ vanishes only at zero, then $\varphi \circ \psi(\cdot^2)$ is strictly conditionally positive definite of order one and radial.

This theorem is a generalization of a classical result in linear algebra by Schur (see, *e.g.*, [Horn and Johnson (1991); Micchelli (1986)], where Schur's theorem was extended to cover strictness).

10.2 Scattered Data Fitting on Manifolds

There exists a sizeable body of literature on the topic of scattered data interpolation on manifolds, especially the sphere S^{s-1} in \mathbb{R}^s . We will not mention any specific results here. Instead we refer the reader to the book [Freeden *et al.* (1998)], the survey papers [Cheney (1995a); Fasshauer and Schumaker (1998)], as well as many original papers such as [Baxter and Hubbert (2001); Bingham (1973); Fasshauer (1995a); Fasshauer (1999b); Hubbert and Morton (2004a); Hubbert and Morton (2004b); Levesley *et al.* (1999); Menegatto (1994b); Narcowich and Ward (2002); Ragozin and Levesley (1996); Ron and Sun (1996); Schoenberg (1942); Schreiner (1997); Wahba (1981); Wahba (1982); Xu and Cheney (1992b)].

Radial basis functions on more general Riemannian manifolds are studied in, *e.g.*, [Dyn *et al.* (1997); Dyn *et al.* (1999); Levesley and Ragozin (2002); Narcowich (1995); Narcowich *et al.* (2003); Schimmin and Belger (1991)].

There is also a "poor man's solution" to interpolation on manifolds, especially the sphere. One can use the Euclidean radial basis function methods discussed thus far, and simply restrict their evaluation to the manifold. This approach is outlined in Section 6 of [Fasshauer and Schumaker (1998)].

We will discuss another, related, interpolation problem later. Namely, interpolation to point cloud data in \mathbb{R}^3 . In this case, the underlying manifold is unknown, and another approach needs to be taken. See Chapter 30 for details.

10.3 Remarks

Many of the results given in the previous chapters can be generalized to vector-valued or even matrix-valued functions. Some results along these lines can be found in [Lowitzsch (2002); Lowitzsch (2005); Myers (1992); Narcowich and Ward (1994a); Schaback (1995a)].

We point out that the approach to solving the interpolation problems taken in the previous chapters always assumes that the centers, *i.e.*, the points \mathbf{x}_k , $k = 1, \dots, N$, at which the basis functions are centered, coincide with the data sites. This is a fairly severe restriction, and it has been shown in examples in the context of least squares approximation of scattered data (see, *e.g.*, [Franke *et al.* (1994); Franke *et al.* (1995)] or [Fasshauer (1995a)]) that better results can be achieved if the centers are chosen different from the data sites. Theoretical results in this direction are very limited, and are reported in [Quak *et al.* (1993)] and in [Sun (1993a)].

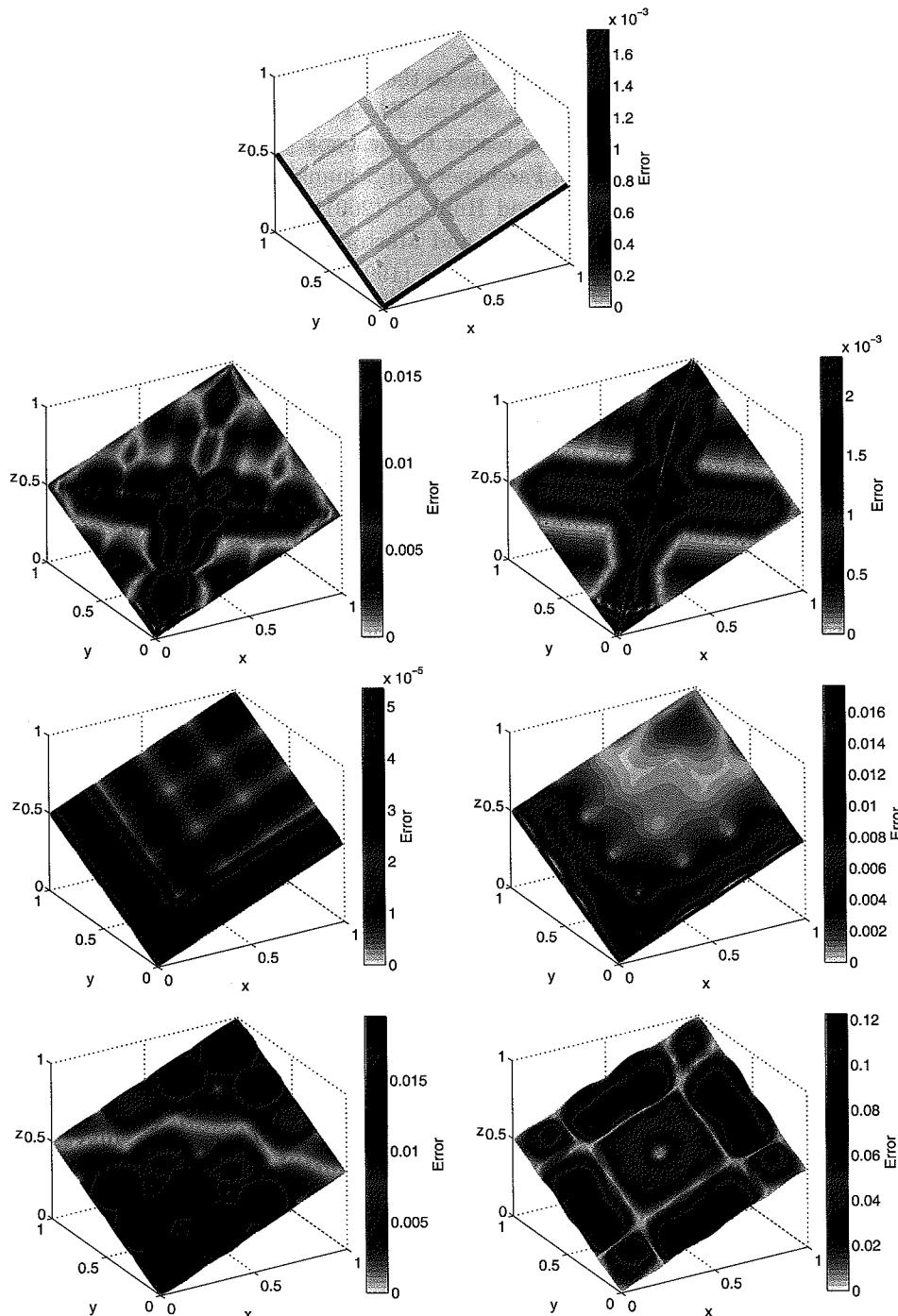


Fig. 10.2 p -norm distance matrix fits to $f(x, y) = (x + y)/2$ on a 5×5 grid in $[0, 1]^2$ unless noted otherwise. Top: $p = 1$ (1089 Halton points). 2nd row: $p = 10$ (left), $p = 100$ (right). 3rd row: $p = 1.001$ (left), $p = 2$ (right). Bottom: p -norm Gaussian fits for $p = 1$ (left) and $p = 10$ (right).

Chapter 11

Compactly Supported Radial Basis Functions

As we saw earlier (see Theorem 9.4), compactly supported functions Φ that are truly strictly conditionally positive definite of order $m > 0$ do not exist. The compact support automatically ensures that Φ is strictly positive definite. Another observation (see Theorem 3.9) was that compactly supported radial functions can be strictly positive definite on \mathbb{R}^s only for a fixed maximal s -value. It is not possible for a function to be strictly positive definite and radial on \mathbb{R}^s for all s and also have a compact support. Therefore we focus our attention on the characterization and construction of functions that are compactly supported, strictly positive definite and radial on \mathbb{R}^s for some fixed s .

According to our earlier work (Bochner's theorem and generalizations thereof), a function is strictly positive definite and radial on \mathbb{R}^s if its s -variate Fourier transform is non-negative. Theorem B.1 in the Appendix gives the Fourier transform of the radial function $\Phi = \varphi(\|\cdot\|)$ as another radial function

$$\hat{\Phi}(x) = \mathcal{F}_s \varphi(\|x\|) = \|x\|^{-(s-2)/2} \int_0^\infty \varphi(t) t^{s/2} J_{(s-2)/2}(t\|x\|) dt,$$

where J_ν is the Bessel function of the first kind of order ν .

11.1 Operators for Radial Functions and Dimension Walks

A certain integral operator and its inverse differential operator were defined in [Schaback and Wu (1996)]. In that paper an entire calculus was developed for how these operators act on radial functions. In fact, according to [Gneiting (2002)], these operators can be traced back to [Matheron (1965)] who called the integral operator *montée* and the differential operator *descente* motivated by an application related to mining.

In the following we define these operators and show how they facilitate the construction of compactly supported radial functions.

Definition 11.1.

- (1) Let φ be such that $t \mapsto t\varphi(t) \in L_1[0, \infty)$. Then we define the *integral operator* \mathcal{I} via

$$(\mathcal{I}\varphi)(r) = \int_r^\infty t\varphi(t)dt, \quad r \geq 0.$$

- (2) For even $\varphi \in C^2(\mathbb{R})$ we define the *differential operator* \mathcal{D} via

$$(\mathcal{D}\varphi)(r) = -\frac{1}{r}\varphi'(r), \quad r \geq 0.$$

In both cases the resulting functions are to be interpreted as even functions using even extensions.

Note that the integral operator \mathcal{I} differs from the operator I introduced earlier (see (5.1)) by a factor t in the integrand.

The most important properties of the montée and descente operators are (see, e.g., [Schaback and Wu (1996)] or [Wendland (1995)]):

Theorem 11.1.

- (1) Both \mathcal{D} and \mathcal{I} preserve compact support, i.e., if φ has compact support, then so do $\mathcal{D}\varphi$ and $\mathcal{I}\varphi$.
- (2) If $\varphi \in C(\mathbb{R})$ and $t \mapsto t\varphi(t) \in L_1[0, \infty)$, then $\mathcal{D}\mathcal{I}\varphi = \varphi$.
- (3) If $\varphi \in C^2(\mathbb{R})$ ($\varphi \not\equiv 1$) is even and $\varphi' \in L_1[0, \infty)$, then $\mathcal{I}\mathcal{D}\varphi = \varphi$.
- (4) If $t \mapsto t^{s-1}\varphi(t) \in L_1[0, \infty)$ and $s \geq 3$, then $\mathcal{F}_s(\varphi) = \mathcal{F}_{s-2}(\mathcal{I}\varphi)$.
- (5) If $\varphi \in C^2(\mathbb{R})$ is even and $t \mapsto t^s\varphi'(t) \in L_1[0, \infty)$, then $\mathcal{F}_s(\varphi) = \mathcal{F}_{s+2}(\mathcal{D}\varphi)$.

The operators \mathcal{I} and \mathcal{D} allow us to express s -variate Fourier transforms as $(s-2)$ - or $(s+2)$ -variate Fourier transforms, respectively. In particular, a direct consequence of the above properties and the characterization of strictly positive definite radial functions (Theorem 3.6) is

Theorem 11.2.

- (1) Suppose $\varphi \in C(\mathbb{R})$. If $t \mapsto t^{s-1}\varphi(t) \in L_1[0, \infty)$ and $s \geq 3$, then φ is strictly positive definite and radial on \mathbb{R}^s if and only if $\mathcal{I}\varphi$ is strictly positive definite and radial on \mathbb{R}^{s-2} .
- (2) If $\varphi \in C^2(\mathbb{R})$ is even and $t \mapsto t^s\varphi'(t) \in L_1[0, \infty)$, then φ is strictly positive definite and radial on \mathbb{R}^s if and only if $\mathcal{D}\varphi$ is strictly positive definite and radial on \mathbb{R}^{s+2} .

This allows us to construct new strictly positive definite radial functions from given ones by a “dimension-walk” technique that steps through multivariate Euclidean space in even increments. The examples presented in the following sections illustrate this technique.

11.2 Wendland’s Compactly Supported Functions

Probably the most popular family of compactly supported radial functions presently in use was constructed in [Wendland (1995)]. Wendland starts with the truncated power function $\varphi_\ell(r) = (1-r)_+^\ell$ (which we know to be strictly positive definite and radial on \mathbb{R}^s for $\ell \geq \lfloor \frac{s}{2} \rfloor + 1$ according to Section 4.6), and then he walks through dimensions by repeatedly applying the operator \mathcal{I} .

Definition 11.2. With $\varphi_\ell(r) = (1-r)_+^\ell$ we define

$$\varphi_{s,k} = \mathcal{I}^k \varphi_{\lfloor s/2 \rfloor + k + 1}.$$

It turns out that the functions $\varphi_{s,k}$ are all supported on $[0, 1]$ and have a polynomial representation there. More precisely,

Theorem 11.3. The functions $\varphi_{s,k}$ are strictly positive definite and radial on \mathbb{R}^s and are of the form

$$\varphi_{s,k}(r) = \begin{cases} p_{s,k}(r), & r \in [0, 1], \\ 0, & r > 1, \end{cases}$$

with a univariate polynomial $p_{s,k}$ of degree $\lfloor s/2 \rfloor + 3k + 1$. Moreover, $\varphi_{s,k} \in C^{2k}(\mathbb{R})$ are unique up to a constant factor, and the polynomial degree is minimal for given space dimension s and smoothness $2k$.

This theorem states that any other compactly supported C^{2k} polynomial function that is strictly positive definite and radial on \mathbb{R}^s will not have a smaller polynomial degree. Our other examples below (Wu’s functions, Gneiting’s functions) illustrate this fact. The strict positive definiteness of Wendland’s functions $\varphi_{s,k}$ starting with non-integer values of ℓ in Definition 11.2 was established in [Gneiting (1999)]. Note, however, that then the functions are no longer guaranteed to be polynomials on their support.

Wendland gave recursive formulas for the functions $\varphi_{s,k}$ for all s, k . We instead list the explicit formulas of [Fasshauer (1999a)].

Theorem 11.4. The functions $\varphi_{s,k}$, $k = 0, 1, 2, 3$, have the form

$$\begin{aligned} \varphi_{s,0}(r) &= (1-r)_+^\ell, \\ \varphi_{s,1}(r) &\doteq (1-r)_+^{\ell+1} [(\ell+1)r+1], \\ \varphi_{s,2}(r) &\doteq (1-r)_+^{\ell+2} [(\ell^2+4\ell+3)r^2+(3\ell+6)r+3], \\ \varphi_{s,3}(r) &\doteq (1-r)_+^{\ell+3} [(\ell^3+9\ell^2+23\ell+15)r^3+(6\ell^2+36\ell+45)r^2 \\ &\quad +(15\ell+45)r+15], \end{aligned}$$

where $\ell = \lfloor s/2 \rfloor + k + 1$, and the symbol \doteq denotes equality up to a multiplicative positive constant.

Proof. The case $k = 0$ follows directly from the definition. Application of the definition for the case $k = 1$ yields

$$\begin{aligned}\varphi_{s,1}(r) &= (\mathcal{I}\varphi_\ell)(r) = \int_r^\infty t\varphi_\ell(t)dt \\ &= \int_r^\infty t(1-t)_+^\ell dt \\ &= \int_r^1 t(1-t)^\ell dt \\ &= \frac{1}{(\ell+1)(\ell+2)}(1-r)^{\ell+1}[(\ell+1)r+1],\end{aligned}$$

where the compact support of φ_ℓ reduces the improper integral to a definite integral which can be evaluated using integration by parts. The other two cases are obtained similarly by repeated application of \mathcal{I} . \square

Example 11.1. For $s = 3$ we list some of the most commonly used functions in Table 11.1. These functions are strictly positive definite and radial on \mathbb{R}^s for $s \leq 3$. We also list their degree of smoothness $2k$. The functions were determined using the formulas from Theorem 11.4, i.e., for $k = 1, 2, 3$ they match Definition 11.2 only up to a positive constant factor.

For the MATLAB implementation in the next chapter it is better to express the compactly supported functions in a shifted form since we will be using a matrix version of $1-\varepsilon r$ in place of the code used earlier in *DistanceMatrix.m* for r . Thus we also list the appropriate functions $\tilde{\varphi}_{s,k} = \varphi_{s,k}(1-\cdot)$ so that $\tilde{\varphi}_{s,k}(1-\varepsilon r) = \varphi_{s,k}(\varepsilon r)$.

For clarification purposes we reiterate that expressions of the form $(x)_+^\ell$ are to be interpreted as $((x)_+)_+^\ell$, i.e., we first apply the cutoff function, and then the power.

Table 11.1 Wendland's compactly supported radial functions $\varphi_{s,k}$ and $\tilde{\varphi}_{s,k} = \varphi_{s,k}(1-\cdot)$ for various choices of k and $s = 3$.

k	$\varphi_{3,k}(r)$	$\tilde{\varphi}_{3,k}(r)$	smoothness
0	$(1-r)_+^2$	r_+^2	C^0
1	$(1-r)_+^4(4r+1)$	$r_+^4(5-4r)$	C^2
2	$(1-r)_+^6(35r^2+18r+3)$	$r_+^6(56-88r+35r^2)$	C^4
3	$(1-r)_+^8(32r^3+25r^2+8r+1)$	$r_+^8(66-154r+121r^2-32r^3)$	C^6

11.3 Wu's Compactly Supported Functions

In [Wu (1995b)] we find another way to construct strictly positive definite radial functions with compact support. Wu starts with the function

$$\psi(r) = (1-r^2)_+^\ell, \quad \ell \in \mathbb{N},$$

which in itself is not positive definite (see the discussion at the end of Chapter 5). However, Wu then uses convolution to construct another function that is strictly positive definite and radial on \mathbb{R} , i.e.,

$$\begin{aligned}\psi_\ell(r) &= (\psi * \psi)(2r) \\ &= \int_{-\infty}^\infty (1-t^2)_+^\ell (1-(2r-t)^2)_+^\ell dt \\ &= \int_{-1}^1 (1-t^2)_+^\ell (1-(2r-t)^2)_+^\ell dt.\end{aligned}$$

This function is strictly positive definite since its Fourier transform is essentially the square of the Fourier transform of ψ and therefore non-negative. Just like the Wendland functions, this function is a polynomial on its support. In fact, the degree of the polynomial is $4\ell + 1$, and $\psi_\ell \in C^{2\ell}(\mathbb{R})$.

Now, a family of strictly positive definite radial functions is constructed by a dimension walk using the \mathcal{D} operator.

Definition 11.3. With $\psi_\ell(r) = ((1-\cdot^2)_+^\ell * (1-\cdot^2)_+^\ell)(2r)$ we define

$$\psi_{k,\ell} = \mathcal{D}^k \psi_\ell.$$

The functions $\psi_{k,\ell}$ are strictly positive definite and radial on \mathbb{R}^s for $s \leq 2k+1$, are polynomials of degree $4\ell - 2k + 1$ on their support and in $C^{2(\ell-k)}$ in the interior of the support. On the boundary the smoothness increases to $C^{2\ell-k}$.

Example 11.2. For $\ell = 3$ we can compute the four functions

$$\psi_{k,3}(r) = \mathcal{D}^k \psi_3(r) = \mathcal{D}^k((1-\cdot^2)_+^3 * (1-\cdot^2)_+^3)(2r), \quad k = 0, 1, 2, 3.$$

They are listed in Table 11.2 along with their smoothness. The maximal space dimension s for which these functions are strictly positive definite and radial on \mathbb{R}^s is also listed. Just as with the Wendland functions, the functions in Table 11.2 match the definition only up to a positive multiplicative constant. Again, we also list the functions $\tilde{\psi}_{k,\ell} = \psi_{k,\ell}(1-\cdot)$ used in our MATLAB implementation in Chapter 12. This representation of the Wu functions is given in Table 11.3.

Table 11.2 Wu's compactly supported radial functions $\psi_{k,\ell}$ for various choices of k and $\ell = 3$.

k	$\psi_{k,3}(r)$	smoothness	s
0	$(1-r)_+^7(5+35r+101r^2+147r^3+101r^4+35r^5+5r^6)$	C^6	1
1	$(1-r)_+^6(6+36r+82r^2+72r^3+30r^4+5r^5)$	C^4	3
2	$(1-r)_+^5(8+40r+48r^2+25r^3+5r^4)$	C^2	5
3	$(1-r)_+^4(16+29r+20r^2+5r^3)$	C^0	7

Table 11.3 Shifted version $\tilde{\psi}_{k,\ell}$ of Wu's compactly supported radial functions $\psi_{k,\ell}$ for various choices of k and $\ell = 3$.

k	$\tilde{\psi}_{k,3}(r)$	smoothness	s
0	$r_+^7(429 - 1287r + 1573r^2 - 1001r^3 + 351r^4 - 65r^5 + 5r^6)$	C^6	1
1	$r_+^6(231 - 561r + 528r^2 - 242r^3 + 55r^4 - 5r^5)$	C^4	3
2	$r_+^5(126 - 231r + 153r^2 - 45r^3 + 5r^4)$	C^2	5
3	$r_+^4(70 - 84r + 35r^2 - 5r^3)$	C^0	7

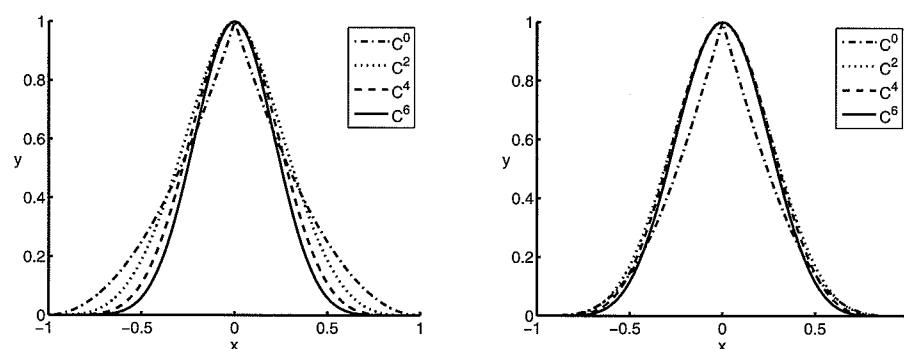


Fig. 11.1 Plot of Wendland's functions from Example 11.1 (left) and Wu's functions from Example 11.2 (right).

As predicted by Theorem 11.3, for a prescribed smoothness the polynomial degree of Wendland's functions is lower than that of Wu's functions. For example, both Wendland's function $\varphi_{3,2}$ and Wu's function $\psi_{1,3}$ are C^4 smooth and strictly positive definite and radial on \mathbb{R}^3 . However, the polynomial degree of Wendland's function is 8, whereas that of Wu's function is 11. Another comparable function is Gneiting's oscillatory function σ_2 (see Table 11.5), which is a C^4 polynomial of degree 9 that is strictly positive definite and radial on \mathbb{R}^3 .

While the two families of strictly positive definite compactly supported functions discussed above are both constructed via dimension walk, Wendland uses integration (and thus obtains a family of increasingly smoother functions), whereas Wu needs to start with a function of sufficient smoothness, and then obtains successively less smooth functions (via differentiation).

11.4 Oscillatory Compactly Supported Functions

Other strictly positive definite compactly supported radial functions have been proposed by Gneiting (see, e.g., [Gneiting (2002)]). He showed that a family of oscillatory compactly supported functions can be constructed using the so-called *turning*

bands operator of [Matheron (1973)]. Starting with a function φ_s that is strictly positive definite and radial on \mathbb{R}^s for $s \geq 3$ the turning bands operator produces

$$\varphi_{s-2}(r) = \varphi_s(r) + \frac{r\varphi'_s(r)}{s-2} \quad (11.1)$$

which is strictly positive definite and radial on \mathbb{R}^{s-2} .

Example 11.3. One such family of functions is generated if we start with the Wendland functions $\varphi_{s+2,1}(r) = (1-r)_+^{\ell+1} [(\ell+1)r+1]$ (ℓ non-integer allowed). Application of the turning bands operator results in the functions

$$\tau_{s,\ell}(r) = (1-r)_+^\ell \left(1 + \ell r - \frac{(\ell+1)(\ell+2+s)}{s} r^2 \right),$$

which are strictly positive definite and radial on \mathbb{R}^s provided $\ell \geq \frac{s+5}{2}$ (see [Gneiting (2002)]). Some specific functions from this family are listed in Table 11.4. All of the functions are in $C^2(\mathbb{R})$. If we want smoother functions, then we need to start with a smoother Wendland family as described below in Example 11.4.

Table 11.4 Gneiting's compactly supported radial functions $\tau_{s,\ell}$ for various choices of ℓ and $s = 2$.

ℓ	$\tau_{2,\ell}(r)$	smoothness
7/2	$(1-r)_+^{7/2} (1 + \frac{7}{2}r - \frac{135}{8}r^2)$	C^2
5	$(1-r)_+^5 (1 + 5r - 27r^2)$	C^2
15/2	$(1-r)_+^{15/2} (1 + \frac{15}{2}r - \frac{391}{8}r^2)$	C^2
12	$(1-r)_+^{12} (1 + 12r - 104r^2)$	C^2

The functions of Table 11.4 are shown in the left plot of Figure 11.2 with ℓ increasing from the outside in (as viewed near the origin).

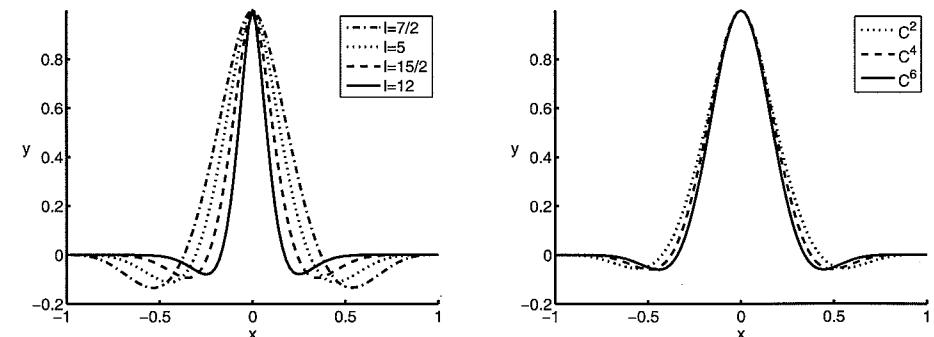


Fig. 11.2 Oscillatory functions of Table 11.4 (left) and Table 11.5 (right).

Example 11.4. Alternatively, we can obtain a set of oscillatory functions that are strictly positive definite and radial on \mathbb{R}^3 by applying the turning bands operator to the Wendland functions $\varphi_{5,k}$ that are strictly positive definite and radial on \mathbb{R}^5 for different choices of k . Then the resulting functions σ_k will have the same degree of smoothness $2k$ as the original functions and they will be strictly positive definite and radial on \mathbb{R}^3 . The results for $k = 1, 2, 3$ are listed in Table 11.5 and displayed in the right plot of Figure 11.2.

Table 11.5 Oscillatory compactly supported functions that are strictly positive definite and radial on \mathbb{R}^3 parametrized by smoothness.

k	$\sigma_k(r)$	smoothness
1	$(1-r)_+^4 (1+4r-15r^2)$	C^2
2	$(1-r)_+^6 (3+18r+3r^2-192r^3)$	C^4
3	$(1-r)_+^8 (15+120r+210r^2-840r^3-3465r^4)$	C^6

Gneiting also suggests the construction of strictly positive definite radial functions by taking the product of the (appropriately scaled) Poisson functions Ω_s (see either Theorem 3.6 or Section 4.3) with a certain compactly supported non-negative function (see [Gneiting (2002)] for more details). By Property (6) of Theorem 3.1 the resulting function will be strictly positive definite.

11.5 Other Compactly Supported Radial Basis Functions

There are many other ways in which one can construct compactly supported functions that are strictly positive definite and radial on \mathbb{R}^s . In [Schaback (1995a)] several such possibilities are described.

Example 11.5. *Euclid's hat* functions are constructed in analogy to *B-splines*. It is well known that the univariate function $\beta(r) = (1 - |r|)_+$ is a second-order *B-spline* with knots at $-1, 0, 1$, and it is obtained as the convolution of the characteristic function of the interval $[-1/2, 1/2]$ with itself. Euclid's hat functions are now obtained by convolving the characteristic function of the s -dimensional Euclidean unit ball with itself. The resulting functions can be written for $r \in [0, 1]$ in the form

$$\varphi_{2k+1}(2r) = \begin{cases} \frac{2\pi\varphi_{2k-1}(2r) - r(1-r^2)^k}{2k+1} & k = 1, 2, 3, \dots, \\ 2(1-r) & k = 0, \end{cases}$$

for odd space dimensions $s = 2k + 1$, and as

$$\varphi_{2k+2}(2r) = \begin{cases} \frac{2\pi\varphi_{2k}(2r) - r\sqrt{1-r^2}(1-r^2)^k}{2k+2} & k = 1, 2, 3, \dots, \\ 2(\arccos r - r\sqrt{1-r^2}) & k = 0, \end{cases}$$

for even space dimensions $s = 2k$. Note that these functions are zero outside the interval $[0, 2]$.

We have listed several of these functions in Table 11.6 where we have employed a substitution $2r \rightarrow r$ and a normalization factor such that the functions all have a value of one at the origin. The functions are also displayed in the left plot of Figure 11.3.

Table 11.6 Euclid's hat functions (defined for $0 \leq r \leq 2$) for different values of s .

s	$\varphi_s(r)$	smoothness
1	$1 - \frac{r}{2}$	C^0
2	$\frac{1}{2\pi} (4 \arccos(\frac{r}{2}) - r\sqrt{4-r^2})$	C^0
3	$1 - \frac{1}{32\pi} ((4 + 16\pi)r - r^3)$	C^0
4	$\frac{2}{\pi} \arccos(\frac{r}{2}) - \frac{1}{32\pi} \sqrt{4-r^2} (20r + r^3)$	C^0
5	$1 - \frac{1}{64\pi^2} ((12 + 8\pi + 32\pi^2)r - (3 + 2\pi)r^3)$	C^0

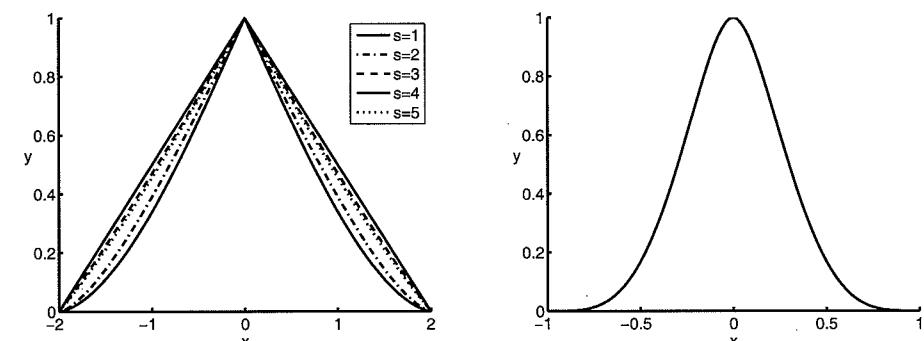


Fig. 11.3 Euclid's hat functions (left) of Table 11.6 and Buhmann's function of Example 11.6 (right).

Another construction described in [Schaback (1995a)] is the radialization of the s -fold tensor product of univariate *B-splines* of even order $2m$ with uniform knots. These functions do not seem to have a simple representation that lends itself to numerical computations. As can be seen from its radialized Fourier transform, the radialized *B-spline* itself is not strictly positive definite and radial on any \mathbb{R}^s with $s > 1$. For $s = 1$ only the *B-splines* of even order are strictly positive definite (see, e.g., [Schölkopf and Smola (2002)]).

The last family of compactly supported strictly positive definite radial functions we would like to mention is due to [Buhmann (1998)]. Buhmann's functions contain

a logarithmic term in addition to a polynomial. His functions have the general form

$$\varphi(r) = \int_0^\infty (1 - r^2/t)_+^\lambda t^\alpha (1 - t^\delta)_+^\rho dt.$$

Here $0 < \delta \leq \frac{1}{2}$, $\rho \geq 1$, and in order to obtain functions that are strictly positive definite and radial on \mathbb{R}^s for $s \leq 3$ the constraints for the remaining parameters are $\lambda \geq 0$, and $-1 < \alpha \leq \frac{\lambda-1}{2}$.

Example 11.6. An example with $\alpha = \delta = \frac{1}{2}$, $\rho = 1$ and $\lambda = 2$ is listed in [Buhmann (2000)]:

$$\varphi(r) \doteq 12r^4 \log r - 21r^4 + 32r^3 - 12r^2 + 1, \quad 0 \leq r \leq 1.$$

This function is in $C^2(\mathbb{R})$ and strictly positive definite and radial on \mathbb{R}^s for $s \leq 3$. It is displayed in the right plot of Figure 11.3.

While it is stated in [Buhmann (2000)] that the construction there encompasses both Wendland's and Wu's functions, an even more general theorem that shows that integration of a positive function $f \in L_1[0, \infty)$ against a strictly positive definite kernel K results in a strictly positive definite function can be found in [Wendland (2005a)] (see also Section 4.8). More specifically,

$$\varphi(r) = \int_0^\infty K(t, r)f(t)dt$$

is strictly positive definite. Buhmann's construction then corresponds to choosing $f(t) = t^\alpha(1 - t^\delta)_+^\rho$ and $K(t, r) = (1 - r^2/t)_+^\lambda$.

Chapter 12

Interpolation with Compactly Supported RBFs in MATLAB

We now have an alternative way to construct an RBF interpolant to scattered data in \mathbb{R}^s . If we use the compactly supported radial functions of the previous chapter then the main difference to our previous interpolants is that now the interpolation matrix can be made *sparse* by scaling the support of the basic function appropriately. To achieve this we use — as we did earlier — the basic functions $\varphi_\varepsilon(r) = \varphi(\varepsilon r)$. Thus, a large value of ε corresponds to a small support. In other words, if the support of φ is the interval $[0, 1]$, then the support radius ρ of φ_ε is given by $\rho = 1/\varepsilon$ so that $\varphi_\varepsilon(r) = 0$ for $r > \rho = 1/\varepsilon$.

Since we know that the interpolation matrix will be a sparse matrix, we want to write MATLAB code to efficiently assemble the matrix. Once we have defined a sparse matrix, MATLAB will automatically use state-of-the-art sparse matrix techniques to solve the linear system. Obviously, we do not want to compute the matrix entries for all pairs of points since we know all of the entries for far away points will be zero. Therefore, an efficient data structure is needed. We use *kd-trees* (implemented in a set of MATLAB MEX-files written by Guy Shechter that can be downloaded from the MATLAB Central File Exchange, see [MCFE]). Some information on *kd-trees* is provided in Appendix A. Data structures for the use with meshfree approximation methods are also discussed in [Wendland (2005a)].

12.1 Assembly of the Sparse Interpolation Matrix

We have structured the scattered data interpolation program in the compactly supported case analogous to the code for the global interpolants, *i.e.*, first construct a distance matrix, and then apply the anonymous function `rbf` to obtain the interpolation/evaluation matrix (as on lines 13–14 and 15–16 of Program 2.1). However, it turns out that it is easier to deal with the compact support if we compute the “distance matrix” corresponding to the $(1 - \varepsilon r)_+$ term since otherwise those entries of the distance matrix that are zero (since the mutual distance between two identical points is zero) would be “lost” in the sparse representation of the matrix.

The MATLAB code `DistanceMatrixCSRBF.m` (Program 12.1) contains two simi-

lar blocks that will be used depending on whether we have more centers than data sites or vice versa. For example, if there are more data sites than centers (cf. lines 7–16), then we build a *kd*-tree for the data sites and find — for each center x_j — those data sites within the support of the basis function centered at x_j , *i.e.*, we construct the (sparse) matrix column by column. In the other case (cf. lines 18–27) we start with a tree for the centers and build the matrix row by row. This is accomplished by determining — for each data site x_i — all centers whose associated basis function covers data site x_i .

The functions `kdtree` and `kdrangequery` are provided by the *kd*-tree library mentioned above. The call in line 7 (respectively 18) of Program 12.1 generates the *kd*-tree of all the centers (data sites), and with the call to `kdrangequery` in line 9 (respectively 20) we find all centers (data sites) that lie within a distance `support` of the j th center point (data site). The actual distances are returned in the vector `dist` and the indices into the list of all data sites are provided in `idx`. The distances for these points only are stored in the matrix `DM`. For maximum efficiency (in order to avoid dynamic memory allocation) it is important to have a good estimate of the number of nonzero entries in the matrix for the allocation statement in lines 4 and 5. The version of the code presented here has the best performance for larger problems since `sparse` is only invoked once.

Program 12.1. DistanceMatrixCSRBF.m

```
% DM = DistanceMatrixCSRBF(dsites,ctrs,ep)
% Forms the distance matrix of two sets of points in R^s
% for compactly supported radial basis functions, i.e.,
%   DM(i,j) = || datasite_i - center_j ||_2.
% The CSRBF used with this code must be given in shifted form
% rbf2(u) = rbf(r), u=1-e*r.
% For example, the Wendland C2
% rbf = @(e,r) max(1-e*r,0).^4.* (4*e*r+1);
% becomes
% rbf2 = @(u) u.^4.* (4*u+5);
% Input
% dsites: Nxs matrix representing a set of N data sites
%         in R^s (i.e., each row contains one
%         s-dimensional point)
% ctrs: Mxs matrix representing a set of M centers for
%         RBFs in R^s (also one center per row)
% ep: determines size of support of basis function.
%     Small ep yields wide function,
%     i.e., supportsize = 1/ep
% Output
% DM: NxM SPARSE matrix that contains the Euclidean
```

```
%           u-distance (u=1-e*r) between the i-th data
%           site and the j-th center in the i,j position
% Uses:    k-D tree package by Guy Shechter from
%           MATLAB Central File Exchange
1 function DM = DistanceMatrixCSRBF(dsites,ctrs,ep)
2 N = size(dsites,1); M = size(ctrs,1);
% Build k-D tree for data sites
% For each center (basis function), find the data sites
% in its support along with u-distance
3 support = 1/ep;
4 nzmax = 25*N; rowidx = zeros(1,nzmax); colidx = zeros(1,nzmax);
5 validx = zeros(1,nzmax); istart = 1; iend = 0;
6 if M > N % faster if more centers than data sites
7 [tmp,tmp,Tree] = kdtree(ctrs,[]);
8 for i = 1:N
9     [pts,dist,idx] = kdrangequery(Tree,dsites(i,:),support);
10    newentries = length(idx);
11    iend = iend + newentries;
12    rowidx(istart:iend) = repmat(i,1,newentries);
13    colidx(istart:iend) = idx';
14    validx(istart:iend) = 1-ep*dist';
15    istart = istart + newentries;
16 end
17 else
18 [tmp,tmp,Tree] = kdtree(dsites,[]);
19 for j = 1:M
20     [pts,dist,idx] = kdrangequery(Tree,ctrs(j,:),support);
21     newentries = length(idx);
22     iend = iend + newentries;
23     rowidx(istart:iend) = idx';
24     colidx(istart:iend) = repmat(j,1,newentries);
25     validx(istart:iend) = 1-ep*dist';
26     istart = istart + newentries;
27 end
28 end
29 idx = find(rowidx);
30 DM = sparse(rowidx(idx),colidx(idx),validx(idx),N,M);
% Free the k-D Tree from memory.
31 kdtree([],[],Tree);
```

The reason for coding `DistanceMatrixCSRBF.m` in two different ways is so that we will be able to speed up the program when dealing with non-square (evaluation) matrices (for example in the context of MLS approximation (*c.f.* Chapter 24)).

One could also implement the distance matrix routine for sparse matrices as follows:

```

1 function DM = DistanceMatrixCSRBF(dsites,ctrs,ep)
2 N = size(dsites,1); M = size(ctrs,1);
% Build k-D tree for data sites
% For each center (basis function), find the data sites
% in its support along with u-distance
3 support = 1/ep; nzmax = 25*N; DM = spalloc(N,M,nzmax);
4 [tmp,tmp,Tree] = kdtree(dsites,[]);
5 for j = 1:M
6     [pts,dist,idx] = kdrangequery(Tree,ctrs(j,:),support);
7     DM(idx,j) = 1-ep*dist;
8 end
% Free the k-D Tree from memory.
9 kdtree([],[],Tree);

```

This code is certainly easier to follow, but not as efficient as the one listed in Program 12.1. Note that we listed only one version of the code here. Clearly, the alternative version can be added analogously to the previous program.

The interpolation program is virtually identical to Program 2.1. The only changes are to replace lines 13 and 15 by the corresponding lines

```

13 DM_data = DistanceMatrixCSRBF(dsites,ctrs,ep);
15 DM_eval = DistanceMatrixCSRBF(epoints,ctrs,ep);

```

and to define the RBF in shifted form, *i.e.*, instead of representing, *e.g.*, the C^2 Wendland function $\varphi_{3,1}$ on line 1 by

```
1 rbf = @(e,r) max(1-e*r,0).^4.*((4*e*r+1); ep=0.7;
```

we now write

```
1 rbf = @(e,r) r.^4.*((5*spones(r)-4*r); ep=0.7;
```

Note the use of the sparse matrix of ones `spones`. Had we used `5-4*r` instead, then a *full* matrix would have been generated (with many additional — and unwanted — ones).

In order to speed up the solution of the (symmetric positive definite) sparse linear system we could use the preconditioned conjugate gradient algorithm (`pcg` in MATLAB) instead of the basic backslash \ (or matrix left division `mldivide`) operation, *i.e.*, we could replace line 17 of Program 2.1 by

```
17 c = pcg(IM,rhs); Pf = EM * c;
```

Note, however, that the \ operator also employs state-of-the-art *direct* sparse solvers by first applying a minimum degree preordering.

12.2 Numerical Experiments with CSRBFS

We now present two sets of interpolation experiments with compactly supported RBFs. In Table 12.2 we use the *non-stationary* approach to interpolation, *i.e.*, the support size remains fixed for increasingly denser sets \mathcal{X} of data sites. In this approach we will be able to observe convergence. However, the matrices become increasingly denser, and therefore the non-stationary approach is very inefficient. In Table 12.1, on the other hand, we use the *stationary* approach, *i.e.*, we scale the support size of the basis functions proportionally to the fill distance $h_{\mathcal{X},\Omega}$ (defined in (2.3)). Now the “bandwidth” of the interpolation matrix A is constant. This theoretically results in $\mathcal{O}(N)$ computational complexity, *i.e.*, a very efficient interpolation method. The stationary interpolation method is also numerically stable, but there will be essentially no convergence (see Table 12.1).

We use Wendland’s compactly supported function $\varphi_{3,1}(r) = (1-r)_+^4 (4r+1)$ to interpolate Franke’s function (2.2) on grids of equally spaced points in the unit square $[0, 1]^2$. In the stationary case (Table 12.1) the support of the basis function starts out with an initial scale parameter $\varepsilon = 0.7$ which is subsequently multiplied by a factor of two whenever the fill distance is halved, *i.e.*, when we repeat the experiment on the next finer grid. This corresponds to keeping a constant number of roughly 25 data sites within the support of any basis function. Therefore, the “bandwidth” of the interpolation matrix A is kept constant (at 25), so that A is very sparse for finer grids. We can observe nice convergence for the first few iterations, but once an RMS-error of approximately 5×10^{-3} is reached, there is not much further improvement. This behavior is not yet fully understood. However, it is similar to what happens in the *approximate approximation* method of Maz’ya (see, *e.g.*, [Maz’ya and Schmidt (2001)] and our discussion in Chapter 26). The rate listed in the table is the exponent of the observed RMS-convergence rate $\mathcal{O}(h^{\text{rate}})$. It is computed using the formula

$$\text{rate}_k = \frac{\ln(e_{k-1}/e_k)}{\ln(h_{k-1}/h_k)}, \quad k = 2, 3, \dots, \quad (12.1)$$

where e_k is the error for experiment number k , and h_k is the fill distance of the k th computational mesh. Note, that for uniformly spaced points the ratio of fill distances of two consecutive meshes will always be two, while for random points (such as Halton points) we estimate the fill distance via (2.4). The % nonzero column indicates the sparsity of the interpolation matrices, and the time is measured in seconds. Errors are computed on an evaluation grid of 40×40 equally spaced points in $[0, 1]^2$.

In the non-stationary case (Table 12.2) we use basis functions without adjusting their support size, *i.e.*, $\varepsilon = 0.7$ is kept fixed for all experiments. We have convergence — although it is not obvious what the rate might be. However, the matrices become increasingly dense and computation requires lots of system memory. Therefore, we left out the solution for the $N = 16641$ and $N = 66049$ cases in Table 12.2. The time

Table 12.1 Stationary interpolation at N equally spaced points in $[0, 1]^2$ (constant 25 points in support) with Wendland's function $\varphi(r) = (1 - r)_+^4 (4r + 1)$.

N	RMS-error	rate	% nonzero	time
9	1.562729e-001		100	0.23
25	2.690350e-002	2.5382	57.8	0.31
81	1.027881e-002	1.3881	23.2	0.33
289	6.589552e-003	0.6414	7.47	0.41
1089	3.891263e-003	0.7599	2.13	0.63
4225	3.726913e-003	0.0623	0.57	1.23
16641	2.638296e-003	0.4984	0.15	3.75
66049	2.467867e-003	0.0963	0.04	15.48

Table 12.2 Non-stationary interpolation at N equally spaced points in $[0, 1]^2$ ($\varepsilon = 0.7$ fixed) with Wendland's function $\varphi(r) = (1 - r)_+^4 (4r + 1)$.

N	RMS-error	rate	time
9	1.562729e-001		0.03
25	2.807706e-002	2.4766	0.04
81	4.853006e-003	2.5324	0.12
289	2.006041e-004	4.5965	0.45
1089	1.288000e-005	3.9611	2.75
4225	1.382497e-006	3.2198	47.92

comparison between the entries in Table 12.1 and Table 12.2 is not a straightforward one since we used the (dense) code Program 2.1 to do the experiments for Table 12.2 since there is no sparseness to be exploited and the kd -trees actually introduce additional overhead.

The interplay between computational efficiency and non-convergence in the stationary case and convergence and computational inefficiency in the non-stationary case is again a *trade-off principle* similar to the interplay between accuracy and ill-conditioning for globally supported RBFs (*c.f.* Chapter 2). These trade-off principles were explained theoretically as well as illustrated with numerical experiments in [Schaback (1997b)], and we will consider them in Chapter 16.

For comparison purposes we repeat the experiments with the oscillatory basic function

$$\varphi(r) = \sigma_2(r) = (1 - r)_+^6 (3 + 18r + 3r^2 - 192r^3),$$

which is also C^4 smooth and strictly positive definite and radial on \mathbb{R}^s for $s \leq 3$ (see Table 11.5). The results are listed in Table 12.3 for the stationary case and in Table 12.4 for the non-stationary case. Note that the function is implemented as

```
rbf = @(e,r) -r.^6.* (168*spones(r)-552*r+573*r.^2-192*r.^3);
```

Table 12.3 Stationary interpolation at N equally spaced points in $[0, 1]^2$ (constant 25 points in support) with the oscillatory function $\varphi(r) = (1 - r)_+^6 (3 + 18r + 3r^2 - 192r^3)$.

N	RMS-error	rate	% nonzero	time
9	1.655969e-001		100	0.28
25	3.941226e-002	2.0710	57.8	0.34
81	2.978973e-002	0.4038	23.2	0.36
289	2.914215e-002	0.0317	7.47	0.42
1089	3.063424e-002	-0.0720	2.13	0.64
4225	3.094308e-002	-0.0145	0.57	1.31
16641	3.089882e-002	0.0021	0.15	4.13
66049	3.086639e-002	0.0015	0.04	16.81

Table 12.4 Non-stationary interpolation at N equally spaced points in $[0, 1]^2$ ($\varepsilon = 0.7$ fixed) with the oscillatory function $\varphi(r) = (1 - r)_+^6 (3 + 18r + 3r^2 - 192r^3)$.

N	RMS-error	rate	time
9	1.655969e-001		0.03
25	3.097850e-002	2.4183	0.06
81	4.612941e-003	2.7475	0.20
289	1.305297e-004	5.1432	0.72
1089	4.780575e-006	4.7711	4.06
4225	2.687479e-007	4.1529	55.09

in the sparse setting and as

```
rbf = @(e,r) max(1-e*r,0).^6.* (3+18*e*r+3*(e*r).^2-192*(e*r).^3);
```

for the dense code.

While the performance of the oscillatory functions for the stationary experiment is even more disappointing than that of Wendland's functions, the situation is reversed in the non-stationary case. In fact, the errors obtained with the oscillatory basis functions are almost as good as those achieved with "optimally" scaled Gaussians (*c.f.* Table 2.2).

In order to overcome the problems due to the trade-off principle that are apparent in both the stationary and non-stationary approach to interpolation with compactly supported radial functions we will later consider using a multilevel stationary scheme (see Chapter 32).

Reproducing Kernel Hilbert Spaces and Native Spaces for Strictly Positive Definite Functions

In the next few chapters we will present some of the theoretical work on error bounds for approximation and interpolation with radial basis functions. Since the discussion for strictly positive definite functions will already be technical enough, we focus on this case, and only mention a few results for the conditionally positive definite case. The following discussion follows mostly the presentation in [Wendland (2005a)] where the interested reader can find many more details.

13.1 Reproducing Kernel Hilbert Spaces

Our first set of error bounds will come rather naturally once we associate with each (strictly positive definite) radial basic function a certain space of functions called its *native space*. We will then be able to establish a connection to reproducing kernel Hilbert spaces, which in turn will give us the desired error bounds as well as certain optimality results for radial basis function interpolation (see Chapter 18).

Reproducing kernels are a classical concept in analysis introduced by Nachman Aronszajn (see [Aronszajn (1950)]). We begin with

Definition 13.1. Let \mathcal{H} be a real Hilbert space of functions $f : \Omega(\subseteq \mathbb{R}^s) \rightarrow \mathbb{R}$ with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. A function $K : \Omega \times \Omega \rightarrow \mathbb{R}$ is called *reproducing kernel for \mathcal{H}* if

- (1) $K(\cdot, \mathbf{x}) \in \mathcal{H}$ for all $\mathbf{x} \in \Omega$,
- (2) $f(\mathbf{x}) = \langle f, K(\cdot, \mathbf{x}) \rangle_{\mathcal{H}}$ for all $f \in \mathcal{H}$ and all $\mathbf{x} \in \Omega$.

The name *reproducing kernel* is inspired by the reproducing property (2) in Definition 13.1. It is known that the reproducing kernel of a Hilbert space is unique, and that existence of a reproducing kernel is equivalent to the fact that the point evaluation functionals $\delta_{\mathbf{x}}$ are bounded linear functionals on Ω , *i.e.*, there exists a positive constant $M = M_{\mathbf{x}}$ such that

$$|\delta_{\mathbf{x}} f| = |f(\mathbf{x})| \leq M \|f\|_{\mathcal{H}}$$

for all $f \in \mathcal{H}$ and all $\mathbf{x} \in \Omega$. This latter fact is due to the Riesz representation theorem.

Other properties of reproducing kernels are given by

Theorem 13.1. Suppose \mathcal{H} is a Hilbert space of functions $f : \Omega \rightarrow \mathbb{R}$ with reproducing kernel K . Then we have

- (1) $K(\mathbf{x}, \mathbf{y}) = \langle K(\cdot, \mathbf{y}), K(\cdot, \mathbf{x}) \rangle_{\mathcal{H}}$ for $\mathbf{x}, \mathbf{y} \in \Omega$.
- (2) $K(\mathbf{x}, \mathbf{y}) = K(\mathbf{y}, \mathbf{x})$ for $\mathbf{x}, \mathbf{y} \in \Omega$.
- (3) Convergence in Hilbert space norm implies pointwise convergence, i.e., if we have $\|f - f_n\|_{\mathcal{H}} \rightarrow 0$ for $n \rightarrow \infty$ then $|f(\mathbf{x}) - f_n(\mathbf{x})| \rightarrow 0$ for all $\mathbf{x} \in \Omega$.

Proof. By Property (1) of Definition 13.1 $K(\cdot, \mathbf{y}) \in \mathcal{H}$ for every $\mathbf{y} \in \Omega$. Then the reproducing property (2) of the definition gives us

$$K(\mathbf{x}, \mathbf{y}) = \langle K(\cdot, \mathbf{y}), K(\cdot, \mathbf{x}) \rangle_{\mathcal{H}}$$

for all $\mathbf{x}, \mathbf{y} \in \Omega$. This establishes (1). Property (2) follows from (1) by the symmetry of the Hilbert space inner product. For (3) we use the reproducing property of K along with the Cauchy-Schwarz inequality:

$$|f(\mathbf{x}) - f_n(\mathbf{x})| = |\langle f - f_n, K(\cdot, \mathbf{x}) \rangle_{\mathcal{H}}| \leq \|f - f_n\|_{\mathcal{H}} \|K(\cdot, \mathbf{x})\|_{\mathcal{H}}. \quad \square$$

Now it is interesting for us that the reproducing kernel K is known to be positive definite. Here we use a slight generalization of the notion of a positive definite function to a positive definite kernel. Essentially, we replace $\Phi(\mathbf{x}_j - \mathbf{x}_k)$ in Definition 3.2 by $K(\mathbf{x}_j, \mathbf{x}_k)$. At this point we remind the reader that the space of bounded linear functionals on \mathcal{H} is known as its *dual*, and denoted by \mathcal{H}^* .

Theorem 13.2. Suppose \mathcal{H} is a reproducing kernel Hilbert function space with reproducing kernel $K : \Omega \times \Omega \rightarrow \mathbb{R}$. Then K is positive definite. Moreover, K is strictly positive definite if and only if the point evaluation functionals $\delta_{\mathbf{x}}$ are linearly independent in \mathcal{H}^* .

Proof. Since the kernel is real-valued we can restrict ourselves to a quadratic form with real coefficients. For distinct points $\mathbf{x}_1, \dots, \mathbf{x}_N$ and nonzero $\mathbf{c} \in \mathbb{R}^N$ we have

$$\begin{aligned} \sum_{j=1}^N \sum_{k=1}^N c_j c_k K(\mathbf{x}_j, \mathbf{x}_k) &= \sum_{j=1}^N \sum_{k=1}^N c_j c_k \langle K(\cdot, \mathbf{x}_j), K(\cdot, \mathbf{x}_k) \rangle_{\mathcal{H}} \\ &= \left\langle \sum_{j=1}^N c_j K(\cdot, \mathbf{x}_j), \sum_{k=1}^N c_k K(\cdot, \mathbf{x}_k) \right\rangle_{\mathcal{H}} \\ &= \left\| \sum_{j=1}^N c_j K(\cdot, \mathbf{x}_j) \right\|_{\mathcal{H}}^2 \geq 0. \end{aligned}$$

Thus K is positive definite. To establish the second claim we assume K is not strictly positive definite and show that the point evaluation functionals must be

linearly dependent. If K is not strictly positive definite then there exist distinct points $\mathbf{x}_1, \dots, \mathbf{x}_N$ and nonzero coefficients c_j such that

$$\sum_{j=1}^N \sum_{k=1}^N c_j c_k K(\mathbf{x}_j, \mathbf{x}_k) = 0.$$

The first part of the proof therefore implies

$$\sum_{j=1}^N c_j K(\cdot, \mathbf{x}_j) = 0.$$

Now we take the Hilbert space inner product with an arbitrary function $f \in \mathcal{H}$ and use the reproducing property of K to obtain

$$\begin{aligned} 0 &= \langle f, \sum_{j=1}^N c_j K(\cdot, \mathbf{x}_j) \rangle_{\mathcal{H}} \\ &= \sum_{j=1}^N c_j \langle f, K(\cdot, \mathbf{x}_j) \rangle_{\mathcal{H}} \\ &= \sum_{j=1}^N c_j f(\mathbf{x}_j) \\ &= \sum_{j=1}^N c_j \delta_{\mathbf{x}_j}(f). \end{aligned}$$

This, however, implies the linear dependence of the point evaluation functionals $\delta_{\mathbf{x}_j}(f) = f(\mathbf{x}_j)$, $j = 1, \dots, N$, since the coefficients c_j were assumed to be not all zero. An analogous argument can be used to establish the converse. \square

This theorem provides one direction of the connection between strictly positive definite functions and reproducing kernels. However, we are also interested in the other direction. Since the RBFs we have built our interpolation methods from are strictly positive definite functions, we want to know how to construct a reproducing kernel Hilbert space associated with those strictly positive definite basic functions.

13.2 Native Spaces for Strictly Positive Definite Functions

In this section we will show that every strictly positive definite radial basic function can indeed be associated with a reproducing kernel Hilbert space — its *native space*.

First, we note that Definition 13.1 tells us that \mathcal{H} contains all functions of the form

$$f = \sum_{j=1}^N c_j K(\cdot, \mathbf{x}_j)$$

provided $\mathbf{x}_j \in \Omega$. As a consequence of Theorem 13.1 we have that

$$\begin{aligned}\|f\|_{\mathcal{H}}^2 &= \langle f, f \rangle_{\mathcal{H}} = \left\langle \sum_{j=1}^N c_j K(\cdot, \mathbf{x}_j), \sum_{k=1}^N c_k K(\cdot, \mathbf{x}_k) \right\rangle_{\mathcal{H}} \\ &= \sum_{j=1}^N \sum_{k=1}^N c_j c_k \langle K(\cdot, \mathbf{x}_j), K(\cdot, \mathbf{x}_k) \rangle_{\mathcal{H}} \\ &= \sum_{j=1}^N \sum_{k=1}^N c_j c_k K(\mathbf{x}_j, \mathbf{x}_k).\end{aligned}$$

Therefore, we define the (possibly infinite-dimensional) space

$$H_K(\Omega) = \text{span}\{K(\cdot, \mathbf{y}) : \mathbf{y} \in \Omega\} \quad (13.1)$$

with an associated bilinear form $\langle \cdot, \cdot \rangle_K$ given by

$$\left\langle \sum_{j=1}^{N_K} c_j K(\cdot, \mathbf{x}_j), \sum_{k=1}^{N_K} d_k K(\cdot, \mathbf{y}_k) \right\rangle_K = \sum_{j=1}^{N_K} \sum_{k=1}^{N_K} c_j d_k K(\mathbf{x}_j, \mathbf{y}_k),$$

where $N_K = \infty$ is also allowed.

Theorem 13.3. *If $K : \Omega \times \Omega \rightarrow \mathbb{R}$ is a symmetric strictly positive definite kernel, then the bilinear form $\langle \cdot, \cdot \rangle_K$ defines an inner product on $H_K(\Omega)$. Furthermore, $H_K(\Omega)$ is a pre-Hilbert space with reproducing kernel K .*

Proof. $\langle \cdot, \cdot \rangle_K$ is obviously bilinear and symmetric. We just need to show that $\langle f, f \rangle_K > 0$ for nonzero $f \in H_K(\Omega)$. Any such f can be written in the form

$$f = \sum_{j=1}^{N_K} c_j K(\cdot, \mathbf{x}_j), \quad \mathbf{x}_j \in \Omega.$$

Then

$$\langle f, f \rangle_K = \sum_{j=1}^{N_K} \sum_{k=1}^{N_K} c_j c_k K(\mathbf{x}_j, \mathbf{x}_k) > 0$$

since K is strictly positive definite. The reproducing property follows from

$$\langle f, K(\cdot, \mathbf{x}) \rangle_K = \sum_{j=1}^{N_K} c_j K(\mathbf{x}, \mathbf{x}_j) = f(\mathbf{x}). \quad \square$$

Since we just showed that $H_K(\Omega)$ is a pre-Hilbert space, i.e., need not be complete, we now define the native space $\mathcal{N}_K(\Omega)$ of K to be the completion of $H_K(\Omega)$ with respect to the K -norm $\|\cdot\|_K$ so that $\|f\|_K = \|f\|_{\mathcal{N}_K(\Omega)}$ for all $f \in H_K(\Omega)$. The technical details concerned with this construction are discussed in [Wendland (2005a)].

In the special case when we are dealing with strictly positive definite (translation invariant) functions $\Phi(\mathbf{x} - \mathbf{y}) = K(\mathbf{x}, \mathbf{y})$ and when $\Omega = \mathbb{R}^s$ we get a characterization of native spaces in terms of Fourier transforms.

Theorem 13.4. *Suppose $\Phi \in C(\mathbb{R}^s) \cap L_1(\mathbb{R}^s)$ is a real-valued strictly positive definite function. Define*

$$\mathcal{G} = \{f \in L_2(\mathbb{R}^s) \cap C(\mathbb{R}^s) : \frac{\hat{f}}{\sqrt{\hat{\Phi}}} \in L_2(\mathbb{R}^s)\}$$

and equip this space with the bilinear form

$$\langle f, g \rangle_{\mathcal{G}} = \frac{1}{\sqrt{(2\pi)^s}} \langle \frac{\hat{f}}{\sqrt{\hat{\Phi}}}, \frac{\hat{g}}{\sqrt{\hat{\Phi}}} \rangle_{L_2(\mathbb{R}^s)} = \frac{1}{\sqrt{(2\pi)^s}} \int_{\mathbb{R}^s} \frac{\hat{f}(\omega) \overline{\hat{g}(\omega)}}{\hat{\Phi}(\omega)} d\omega.$$

Then \mathcal{G} is a real Hilbert space with inner product $\langle \cdot, \cdot \rangle_{\mathcal{G}}$ and reproducing kernel $\Phi(\cdot - \cdot)$. Hence, \mathcal{G} is the native space of Φ on \mathbb{R}^s , i.e., $\mathcal{G} = \mathcal{N}_{\Phi}(\mathbb{R}^s)$ and both inner products coincide. In particular, every $f \in \mathcal{N}_{\Phi}(\mathbb{R}^s)$ can be recovered from its Fourier transform $\hat{f} \in L_1(\mathbb{R}^s) \cap L_2(\mathbb{R}^s)$.

Another characterization of the native space is given in terms of the eigenfunctions of a linear operator associated with the reproducing kernel. This operator, $T_{\Phi} : L_2(\Omega) \rightarrow L_2(\Omega)$, is given by

$$T_{\Phi}(v)(\mathbf{x}) = \int_{\Omega} \Phi(\mathbf{x}, \mathbf{y}) v(\mathbf{y}) d\mathbf{y}, \quad v \in L_2(\Omega), \quad \mathbf{x} \in \Omega.$$

For the eigenvalues λ_k , $k = 1, 2, \dots$, and eigenfunctions ϕ_k of this operator Mercer's theorem [Riesz and Sz.-Nagy (1955)] states

Theorem 13.5 (Mercer). *Let $\Phi(\cdot, \cdot)$ be a continuous positive definite kernel that satisfies*

$$\int_{\Omega} \Phi(\mathbf{x}, \mathbf{y}) v(\mathbf{x}) v(\mathbf{y}) d\mathbf{x} d\mathbf{y} \geq 0, \quad \text{for all } v \in L_2(\Omega), \quad \mathbf{x}, \mathbf{y} \in \Omega. \quad (13.2)$$

Then Φ can be represented by

$$\Phi(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{\infty} \lambda_k \phi_k(\mathbf{x}) \phi_k(\mathbf{y}), \quad (13.3)$$

where λ_k are the (non-negative) eigenvalues and ϕ_k are the (L_2 -orthonormal) eigenfunctions of T_{Φ} . Moreover, this representation is absolutely and uniformly convergent.

We can interpret condition (13.2) as a type of integral positive definiteness. As usual, the eigenvalues and eigenfunctions satisfy $T_{\Phi} \phi_k = \lambda_k \phi_k$ or

$$\int_{\Omega} \Phi(\mathbf{x}, \mathbf{y}) \phi_k(\mathbf{y}) d\mathbf{y} = \lambda_k \phi_k(\mathbf{x}), \quad k = 1, 2, \dots$$

In general, Mercer's theorem allows us to construct a reproducing kernel Hilbert space \mathcal{H} by representing the functions in \mathcal{H} as infinite linear combinations of the eigenfunctions, i.e.,

$$\mathcal{H} = \left\{ f : f = \sum_{k=1}^{\infty} c_k \phi_k \right\}.$$

It is clear that the kernel Φ itself is in \mathcal{H} since it has the eigenfunction expansion (13.3). The inner product for \mathcal{H} is given by

$$\langle f, g \rangle_{\mathcal{H}} = \left\langle \sum_{j=1}^{\infty} c_j \phi_j, \sum_{k=1}^{\infty} d_k \phi_k \right\rangle_{\mathcal{H}} = \sum_{k=1}^{\infty} \frac{c_k d_k}{\lambda_k},$$

where we used the \mathcal{H} -orthogonality

$$\langle \phi_j, \phi_k \rangle_{\mathcal{H}} = \frac{\delta_{jk}}{\sqrt{\lambda_j} \sqrt{\lambda_k}}$$

of the eigenfunctions.

We note that Φ is indeed the reproducing kernel of \mathcal{H} since the eigenfunction expansion (13.3) of Φ and the orthogonality of the eigenfunctions imply

$$\begin{aligned} \langle f, \Phi(\cdot, \mathbf{x}) \rangle_{\mathcal{H}} &= \left\langle \sum_{j=1}^{\infty} c_j \phi_j, \sum_{k=1}^{\infty} \lambda_k \phi_k \phi_k(\mathbf{x}) \right\rangle_{\mathcal{H}} \\ &= \sum_{k=1}^{\infty} \frac{c_k \lambda_k \phi_k(\mathbf{x})}{\lambda_k} \\ &= \sum_{k=1}^{\infty} c_k \phi_k(\mathbf{x}) = f(\mathbf{x}). \end{aligned}$$

Finally, one has (*c.f.* [Wendland (2005a)]) that the native space $\mathcal{N}_{\Phi}(\Omega)$ is given by

$$\mathcal{N}_{\Phi}(\Omega) = \left\{ f \in L_2(\Omega) : \sum_{k=1}^{\infty} \frac{1}{\lambda_k} |\langle f, \phi_k \rangle_{L_2(\Omega)}|^2 < \infty \right\}$$

and the native space inner product can be written as

$$\langle f, g \rangle_{\mathcal{N}_{\Phi}} = \sum_{k=1}^{\infty} \frac{1}{\lambda_k} \langle f, \phi_k \rangle_{L_2(\Omega)} \langle g, \phi_k \rangle_{L_2(\Omega)}, \quad f, g \in \mathcal{N}_{\Phi}(\Omega).$$

Since $\mathcal{N}_{\Phi}(\Omega)$ is a subspace of $L_2(\Omega)$ this corresponds to the identification $c_k = \langle f, \phi_k \rangle_{L_2(\Omega)}$ of the generalized Fourier coefficients in the discussion above.

13.3 Examples of Native Spaces for Popular Radial Basic Functions

Theorem 13.4 shows that native spaces of translation invariant functions can be viewed as a generalization of standard *Sobolev spaces*. Indeed, for $m > s/2$ the Sobolev space W_2^m can be defined as (see, *e.g.*, [Adams (1975)])

$$W_2^m(\mathbb{R}^s) = \{f \in L_2(\mathbb{R}^s) \cap C(\mathbb{R}^s) : \hat{f}(\cdot)(1 + \|\cdot\|_2^2)^{m/2} \in L_2(\mathbb{R}^s)\}. \quad (13.4)$$

One also frequently sees the definition

$$W_2^m(\Omega) = \{f \in L_2(\Omega) \cap C(\Omega) : D^{\alpha} f \in L_2(\Omega) \text{ for all } |\alpha| \leq m, \alpha \in \mathbb{N}^s\}, \quad (13.5)$$

which applies whenever $\Omega \subset \mathbb{R}^s$ is a bounded domain. This interpretation will make clear the connection between the natives spaces of Sobolev splines and those of polyharmonic splines to be discussed below. The norm of $W_2^m(\mathbb{R}^s)$ is usually given by

$$\|f\|_{W_2^m(\mathbb{R}^s)} = \left\{ \sum_{|\alpha| \leq m} \|D^{\alpha} f\|_{L_2(\mathbb{R}^s)}^2 \right\}^{1/2}.$$

According to (13.4), any strictly positive definite function Φ whose Fourier transform decays only algebraically has a Sobolev space as its native space. In particular, the Matérn functions

$$\Phi_{\beta}(\mathbf{x}) = \frac{K_{\beta - \frac{s}{2}}(\|\mathbf{x}\|) \|\mathbf{x}\|^{\beta - \frac{s}{2}}}{2^{\beta - 1} \Gamma(\beta)}, \quad \beta > \frac{s}{2},$$

of Section 4.4 with Fourier transform

$$\hat{\Phi}_{\beta}(\omega) = (1 + \|\omega\|^2)^{-\beta}$$

can immediately be seen to have native space $\mathcal{N}_{\Phi_{\beta}}(\mathbb{R}^s) = W_2^{\beta}(\mathbb{R}^s)$ with $\beta > s/2$ (which is why some people refer to the Matérn functions as Sobolev splines).

Wendland's compactly supported functions $\Phi_{s,k} = \varphi_{s,k}(\|\cdot\|_2)$ of Chapter 11 can be shown to have native spaces $\mathcal{N}_{\Phi_{s,k}}(\mathbb{R}^s) = W_2^{s/2+k+1/2}(\mathbb{R}^s)$ (where the restriction $s \geq 3$ is required for the special case $k = 0$).

Native spaces for strictly conditionally positive definite functions can also be constructed. However, since this is more technical, we limited the discussion above to strictly positive definite functions, and refer the interested reader to the book [Wendland (2005a)] or the papers [Schaback (1999a); Schaback (2000a)]. With the extension of the theory to strictly conditionally positive definite functions the native spaces of the radial powers and thin plate (or surface) splines of Sections 8.2 and 8.3 can be shown to be the so-called *Beppo-Levi spaces* of order k

$$\text{BL}_k(\mathbb{R}^s) = \{f \in C(\mathbb{R}^s) : D^{\alpha} f \in L_2(\mathbb{R}^s) \text{ for all } |\alpha| = k, \alpha \in \mathbb{N}^s\},$$

where D^{α} denotes a *generalized derivative* of order α (defined in the same spirit as the generalized Fourier transform, see Appendix B). In fact, the intersection of all Beppo-Levi spaces $\text{BL}_k(\mathbb{R}^s)$ of order $k \leq m$ yields the Sobolev space $W_2^m(\mathbb{R}^s)$. In the literature the Beppo-Levi spaces $\text{BL}_k(\mathbb{R}^s)$ are sometimes referred to as *homogeneous Sobolev spaces of order k* . Alternatively, the Beppo-Levi spaces on \mathbb{R}^s are defined as

$$\text{BL}_k(\mathbb{R}^s) = \{f \in C(\mathbb{R}^s) : \hat{f}(\cdot) \|\cdot\|_2^m \in L_2(\mathbb{R}^s)\},$$

and the formulas given in Chapter 8 for the Fourier transforms of radial powers and thin plate splines show immediately that their native spaces are Beppo-Levi spaces. The semi-norm on BL_k is given by

$$\|f\|_{\text{BL}_k} = \left\{ \sum_{|\alpha|=k} \frac{k!}{\alpha_1! \dots \alpha_d!} \|D^{\alpha} f\|_{L_2(\mathbb{R}^s)}^2 \right\}^{1/2}, \quad (13.6)$$

and its kernel is the polynomial space Π_{k-1}^s . For more details see [Wendland (2005a)]. Beppo-Levi spaces were already studied in the early papers [Duchon (1976); Duchon (1977); Duchon (1978); Duchon (1980)].

The native spaces for Gaussians and (inverse) multiquadratics are rather small. For example, according to Theorem 13.4, for Gaussians the Fourier transform of $f \in \mathcal{N}_\Phi(\Omega)$ must decay faster than the Fourier transform of the Gaussian (which is itself a Gaussian). It is known that, even though the native space of Gaussians is small, it does contain the important class of so-called *band-limited functions*, *i.e.*, functions whose Fourier transform is compactly supported. These functions play an important role in *sampling theory* where Shannon's famous sampling theorem [Shannon (1949)] states that any band-limited function can be completely recovered from its discrete samples provided the function is sampled at a sampling rate at least twice its bandwidth. The content of this theorem was already known much earlier (see [Whittaker (1915)]).

Theorem 13.6 (Shannon Sampling). *Suppose $f \in C(\mathbb{R}^s) \cap L_1(\mathbb{R}^s)$ such that its Fourier transform vanishes outside the cube $Q = [-\frac{1}{2}, \frac{1}{2}]^s$. Then f can be uniquely reconstructed from its values on \mathbb{Z}^s , *i.e.*,*

$$f(\mathbf{x}) = \sum_{\boldsymbol{\xi} \in \mathbb{Z}^s} f(\boldsymbol{\xi}) \text{sinc}(\mathbf{x} - \boldsymbol{\xi}), \quad \mathbf{x} \in \mathbb{R}^s.$$

Here the sinc function is defined for any $\mathbf{x} = (x_1, \dots, x_s) \in \mathbb{R}^s$ as $\text{sinc } \mathbf{x} = \prod_{d=1}^s \frac{\sin(\pi x_d)}{\pi x_d}$. For more details on Shannon's sampling theorem see, *e.g.*, Chapter 29 in the book [Cheney and Light (1999)] or the paper [Unser (2000)].

Chapter 14

The Power Function and Native Space Error Estimates

14.1 Fill Distance and Approximation Orders

Our goal in this section is to provide error estimates for scattered data interpolation with strictly (conditionally) positive definite functions. As in the previous chapter we will provide most of the details for the strictly positive definite case, and only mention the extension to the conditionally positive definite case in the end. In their final form we will want our estimates to depend on some kind of measure of the data distribution. The measure that is usually used in approximation theory is the so-called *fill distance*

$$h = h_{\mathcal{X}, \Omega} = \sup_{\mathbf{x} \in \Omega} \min_{\mathbf{x}_j \in \mathcal{X}} \|\mathbf{x} - \mathbf{x}_j\|_2$$

already introduced in (2.3) in Chapter 2. The fill distance indicates how well the data fill out the domain Ω , and it therefore denotes the radius of the largest empty ball that can be placed among the data locations. We will be interested in whether the error

$$\|f - \mathcal{P}_f^{(h)}\|_\infty$$

tends to zero as $h \rightarrow 0$, and if so, how fast. Here $\{\mathcal{P}_f^{(h)}\}_h$ presents a sequence of interpolation (or, more generally, projection) operators that vary with the fill distance h . For example, $\mathcal{P}_f^{(h)}$ could denote interpolation to data given at $(2^n + 1)^s$, $n = 1, 2, \dots$, equally spaced points in the unit cube in \mathbb{R}^s (with $h = 2^{-n}$) as we used in some of our earlier examples. Of course, the definition of the fill distance also covers scattered data such as sets of Halton points. In fact, since Halton points are quasi-uniformly distributed (see Appendix A) we can assume $h \approx 2^{-n}$ for a set of $(2^n + 1)^s$ Halton points in \mathbb{R}^s . This explains the specific sizes of the point sets we used in earlier examples.

Since we want to employ the machinery of reproducing kernel Hilbert spaces presented in the previous chapter we will concentrate on error estimates for functions $f \in \mathcal{N}_\Phi$. In the next chapter we will also mention some more general estimates.

The term that is often used to measure the speed of convergence to zero is *approximation order*. We say that the approximation operator $\mathcal{P}_f^{(h)}$ has L_p -approximation

order k if

$$\|f - \mathcal{P}_f^{(h)}\|_p = \mathcal{O}(h^k) \quad \text{for } h \rightarrow 0.$$

Moreover, if we can also show that $\|f - \mathcal{P}_f^{(h)}\|_p \neq o(h^k)$, then $\mathcal{P}^{(h)}$ has *exact L_p -approximation order k* . We will concentrate mostly on the case $p = \infty$ (*i.e.*, pointwise estimates), but approximation order in other norms can also be studied.

In order to keep the following discussion as transparent as possible we will restrict ourselves to strictly positive definite functions. With (considerably) more technical details the following can also be formulated for strictly conditionally positive definite functions (see [Wendland (2005a)] for details).

14.2 Lagrange Form of the Interpolant and Cardinal Basis Functions

The key idea for the following discussion is to express the interpolant in *Lagrange form*, *i.e.*, using so-called *cardinal basis functions*. For radial basis function approximation this idea is due to [Wu and Schaback (1993)]. In the previous chapters we established that, for any strictly positive definite function Φ , the linear system

$$Ac = y$$

with $A_{ij} = \Phi(\mathbf{x}_i - \mathbf{x}_j)$, $i, j = 1, \dots, N$, $c = [c_1, \dots, c_N]^T$, and $y = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_N)]^T$ has a unique solution. In the following we will consider the more general situation where Φ is a strictly positive definite kernel, *i.e.*, the entries of A are given by $A_{ij} = \Phi(\mathbf{x}_i, \mathbf{x}_j)$. The uniqueness result holds in this case also.

In order to obtain the cardinal basis functions u_j^* , $j = 1, \dots, N$, with the property $u_j^*(\mathbf{x}_i) = \delta_{ij}$, *i.e.*,

$$u_j^*(\mathbf{x}_i) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j, \end{cases}$$

we consider the linear system

$$Au^*(\mathbf{x}) = b(\mathbf{x}), \quad (14.1)$$

where the matrix A is as above (and therefore invertible), $\mathbf{u}^* = [u_1^*, \dots, u_N^*]^T$, and $\mathbf{b} = [\Phi(\cdot, \mathbf{x}_1), \dots, \Phi(\cdot, \mathbf{x}_N)]^T$. Thus,

Theorem 14.1. Suppose Φ is a strictly positive definite kernel on \mathbb{R}^s . Then, for any distinct points $\mathbf{x}_1, \dots, \mathbf{x}_N$, there exist functions $u_j^* \in \text{span}\{\Phi(\cdot, \mathbf{x}_j), j = 1, \dots, N\}$ such that $u_j^*(\mathbf{x}_i) = \delta_{ij}$.

Therefore, we can write the interpolant \mathcal{P}_f to f at $\mathbf{x}_1, \dots, \mathbf{x}_N$ in the cardinal form

$$\mathcal{P}_f(\mathbf{x}) = \sum_{j=1}^N f(\mathbf{x}_j) u_j^*(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^s.$$

It is of interest to note that the cardinal functions do not depend on the data values of the interpolation problem. Once the data sites are fixed and the basic function is chosen with an appropriate shape parameter (whose optimal value will depend on the data), then the cardinal functions are determined by the linear system (14.1). We have plotted various cardinal functions based on the Gaussian basic function with shape parameter $\varepsilon = 5$ in Figures 14.1–14.3. The dependence on the data locations is clearly apparent when comparing the different data distributions (uniformly spaced in Figure 14.1, tensor-product Chebyshev in Figure 14.2, and Halton points in Figure 14.3). The data sets can be seen in Figure 14.5 below.

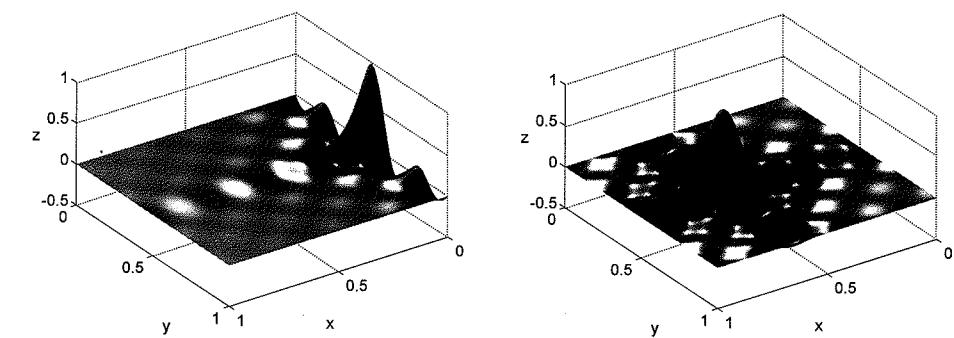


Fig. 14.1 Cardinal functions for Gaussian interpolation (with $\varepsilon = 5$) on 81 uniformly spaced points in $[0, 1]^2$. Centered at an edge point (left) and at an interior point (right).

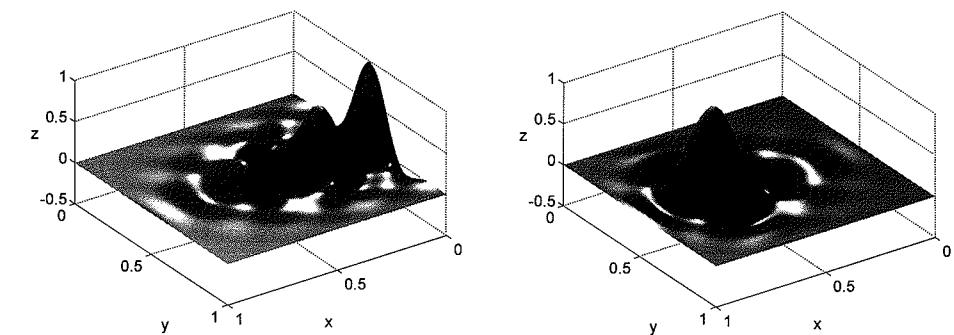


Fig. 14.2 Cardinal functions for Gaussian interpolation (with $\varepsilon = 5$) on 81 tensor-product Chebyshev points in $[0, 1]^2$. Centered at an edge point (left) and at an interior point (right).

Basic functions that grow with increasing distance from the center point (such as multiquadratics) are sometimes criticized for being “counter-intuitive” for scattered data approximation. However, as Figure 14.4 shows, the cardinal functions are just as localized as those for the Gaussian basic functions, and thus the function space

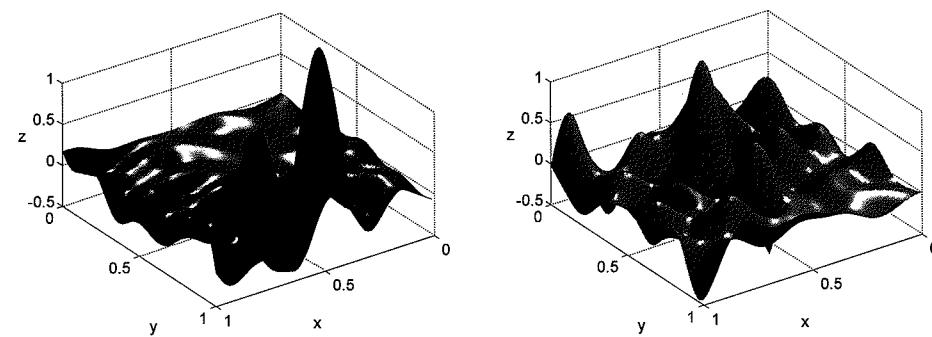


Fig. 14.3 Cardinal functions for Gaussian interpolation (with $\varepsilon = 5$) on 81 Halton points in $[0, 1]^2$. Centered at an edge point (left) and at an interior point (right).

spanned by multiquadratics is a “good” local space.

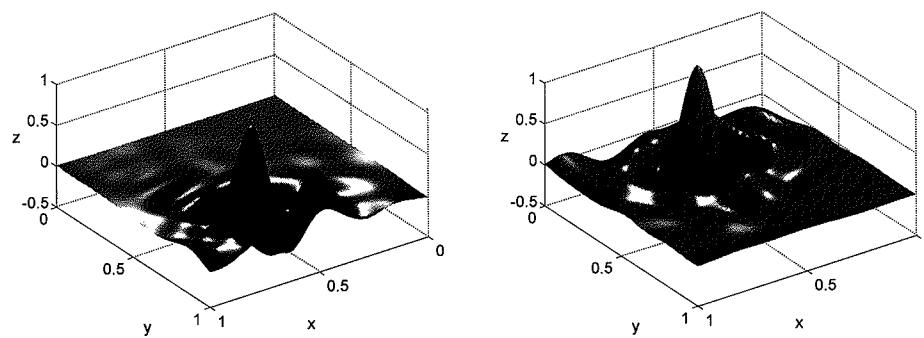


Fig. 14.4 Cardinal functions for multiquadric interpolation (with $\varepsilon = 5$) on 81 Halton points in $[0, 1]^2$. Centered at an edge point (left) and at an interior point (right).

The MATLAB program `RBFCardinalFunction.m` used to produce the plots in Figures 14.1–14.3 is provided in Program 14.1. Note that we use the pseudo-inverse (via the MATLAB command `pinv`) to stably compute the inverse of the interpolation matrix (see line 13 of Program 14.1). A specific cardinal function is then chosen in line 15.

Program 14.1. `RBFCardinalFunction.m`

```
% RBFCardinalFunction
% Computes and plots cardinal function for 2D RBF interpolation
% Calls on: DistanceMatrix
1 rbf = @(e,r) exp(-(e*r).^2); ep = 5;
2 N = 81; gridtype = 'u';
3 neval = 80; M = neval^2;
```

```
% Load data points
4 name = sprintf('Data2D_%d%s',N,gridtype); load(name)
5 ctrs = dsites; % centers coincide with data sites
6 grid = linspace(0,1,neval); [xe,ye] = meshgrid(grid);
7 epoints = [xe(:) ye(:)];
% Compute distance matrix between evaluation points and centers
8 DM_eval = DistanceMatrix(epoints,ctrs);
% Compute distance matrix between the data sites and centers
9 DM_data = DistanceMatrix(dsites,ctrs);
% Compute interpolation matrix
10 IM = rbf(ep,DM_data);
% Compute evaluation matrix
11 EM = rbf(ep,DM_eval);
% Compute cardinal functions at evaluation points
12 invIM = pinv(IM);
% centered at datasite(50)
13 for j=1:M
14     cardvec = (invIM*EM(j,:))';
15     cardfun(j) = cardvec(50);
16 end
17 figure
18 RBFplot = surf(xe,ye,reshape(cardfun,neval,neval));
19 set(RBFplot,'FaceColor','interp','EdgeColor','none')
20 colormap autumn; view([145 45]); camlight; lighting gouraud
```

14.3 The Power Function

Another important ingredient needed for our error estimates is the so-called *power function*. To this end, we consider a domain $\Omega \subseteq \mathbb{R}^s$. Then for any strictly positive definite kernel $\Phi \in C(\Omega \times \Omega)$, any set of distinct points $\mathcal{X} = \{x_1, \dots, x_N\} \subseteq \Omega$, and any vector $\mathbf{u} \in \mathbb{R}^N$, we define the quadratic form

$$Q(\mathbf{u}) = \Phi(\mathbf{x}, \mathbf{x}) - 2 \sum_{j=1}^N u_j \Phi(\mathbf{x}, \mathbf{x}_j) + \sum_{i=1}^N \sum_{j=1}^N u_i u_j \Phi(\mathbf{x}_i, \mathbf{x}_j).$$

Then

Definition 14.1. Suppose $\Omega \subseteq \mathbb{R}^s$ and $\Phi \in C(\Omega \times \Omega)$ is strictly positive definite on \mathbb{R}^s . For any distinct points $\mathcal{X} = \{x_1, \dots, x_N\} \subseteq \Omega$ the *power function* is defined by

$$[P_{\Phi, \mathcal{X}}(\mathbf{x})]^2 = Q(\mathbf{u}^*(\mathbf{x})),$$

where \mathbf{u}^* is the vector of cardinal functions from Theorem 14.1.

Using the definition of the native space norm from the previous chapter we can rewrite the quadratic form $Q(\mathbf{u})$ as

$$\begin{aligned} Q(\mathbf{u}) &= \Phi(\mathbf{x}, \mathbf{x}) - 2 \sum_{j=1}^N u_j \Phi(\mathbf{x}, \mathbf{x}_j) + \sum_{i=1}^N \sum_{j=1}^N u_i u_j \Phi(\mathbf{x}_i, \mathbf{x}_j) \\ &= \langle \Phi(\cdot, \mathbf{x}), \Phi(\cdot, \mathbf{x}) \rangle_{\mathcal{N}_\Phi(\Omega)} - 2 \sum_{j=1}^N u_j \langle \Phi(\cdot, \mathbf{x}), \Phi(\cdot, \mathbf{x}_j) \rangle_{\mathcal{N}_\Phi(\Omega)} \\ &\quad + \sum_{i=1}^N \sum_{j=1}^N u_i u_j \langle \Phi(\cdot, \mathbf{x}_i), \Phi(\cdot, \mathbf{x}_j) \rangle_{\mathcal{N}_\Phi(\Omega)} \\ &= \langle \Phi(\cdot, \mathbf{x}) - \sum_{j=1}^N u_j \Phi(\cdot, \mathbf{x}_j), \Phi(\cdot, \mathbf{x}) - \sum_{j=1}^N u_j \Phi(\cdot, \mathbf{x}_j) \rangle_{\mathcal{N}_\Phi(\Omega)} \\ &= \left\| \Phi(\cdot, \mathbf{x}) - \sum_{j=1}^N u_j \Phi(\cdot, \mathbf{x}_j) \right\|_{\mathcal{N}_\Phi(\Omega)}^2. \end{aligned} \quad (14.2)$$

The name *power function* was chosen by Schaback based on its connection to the power function of a statistical decision function (originally introduced in [Neyman and Pearson (1936)]). In the paper [Wu and Schaback (1993)] the power function was referred to as *kriging function*. This terminology comes from geostatistics (see, e.g., [Myers (1992)]).

Using the linear system notation employed earlier, i.e., $A_{ij} = \Phi(\mathbf{x}_i, \mathbf{x}_j)$, $i, j = 1, \dots, N$, $\mathbf{u} = [u_1, \dots, u_N]^T$, and $\mathbf{b} = [\Phi(\cdot, \mathbf{x}_1), \dots, \Phi(\cdot, \mathbf{x}_N)]^T$, we note that we can also rewrite the quadratic form $Q(\mathbf{u})$ as

$$Q(\mathbf{u}) = \Phi(\mathbf{x}, \mathbf{x}) - 2\mathbf{u}^T \mathbf{b}(\mathbf{x}) + \mathbf{u}^T A \mathbf{u}. \quad (14.3)$$

This suggests two alternative representations of the power function. Using the matrix-vector notation for $Q(\mathbf{u})$, the power function is given as

$$P_{\Phi, \mathcal{X}}(\mathbf{x}) = \sqrt{Q(\mathbf{u}^*(\mathbf{x}))} = \sqrt{\Phi(\mathbf{x}, \mathbf{x}) - 2(\mathbf{u}^*(\mathbf{x}))^T \mathbf{b}(\mathbf{x}) + (\mathbf{u}^*(\mathbf{x}))^T A \mathbf{u}^*(\mathbf{x})}.$$

However, by the definition of the cardinal functions $A\mathbf{u}^*(\mathbf{x}) = \mathbf{b}(\mathbf{x})$, and therefore we have the two new variants

$$\begin{aligned} P_{\Phi, \mathcal{X}}(\mathbf{x}) &= \sqrt{\Phi(\mathbf{x}, \mathbf{x}) - (\mathbf{u}^*(\mathbf{x}))^T \mathbf{b}(\mathbf{x})} \\ &= \sqrt{\Phi(\mathbf{x}, \mathbf{x}) - (\mathbf{u}^*(\mathbf{x}))^T A \mathbf{u}^*(\mathbf{x})}. \end{aligned}$$

These formulas can be used for the numerical evaluation of the power function at \mathbf{x} . To this end one has to first find the value of the cardinal functions $\mathbf{u}^*(\mathbf{x})$ by solving the system $A\mathbf{u}^*(\mathbf{x}) = \mathbf{b}(\mathbf{x})$. This results in

$$P_{\Phi, \mathcal{X}}(\mathbf{x}) = \sqrt{\Phi(\mathbf{x}, \mathbf{x}) - (\mathbf{b}(\mathbf{x}))^T A^{-1} \mathbf{b}(\mathbf{x})}. \quad (14.4)$$

Since A is a positive definite matrix whenever Φ is a strictly positive definite kernel we see that the power function satisfies the bounds

$$0 \leq P_{\Phi, \mathcal{X}}(\mathbf{x}) \leq \sqrt{\Phi(\mathbf{x}, \mathbf{x})}.$$

Plots of the power function for the Gaussian with $\varepsilon = 6$ on three different point sets with $N = 81$ in the unit square are provided in Figure 14.5. The sets of data points are displayed on the left, while the plots of the power function are displayed in the right column. Dependence of the power function on the data locations is clearly visible. In fact, this connection was used in a recent paper [De Marchi *et al.* (2005)] to iteratively obtain an optimal set of data locations that are independent of the data values.

At this point the power function is mostly a theoretical tool that helps us better understand error estimates since we can decouple the effects due to the data function f from those due to the basic function Φ and the data locations \mathcal{X} (see the following Theorem 14.2).

The power function is defined in an analogous way for strictly conditionally positive definite functions.

14.4 Generic Error Estimates for Functions in $\mathcal{N}_\Phi(\Omega)$

Now we can give a first generic error estimate.

Theorem 14.2. *Let $\Omega \subseteq \mathbb{R}^s$, $\Phi \in C(\Omega \times \Omega)$ be strictly positive definite on \mathbb{R}^s , and suppose that the points $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ are distinct. Denote the interpolant to $f \in \mathcal{N}_\Phi(\Omega)$ on \mathcal{X} by \mathcal{P}_f . Then for every $\mathbf{x} \in \Omega$*

$$|f(\mathbf{x}) - \mathcal{P}_f(\mathbf{x})| \leq P_{\Phi, \mathcal{X}}(\mathbf{x}) \|f\|_{\mathcal{N}_\Phi(\Omega)}.$$

Proof. Since f is assumed to lie in the native space of Φ the reproducing property of Φ yields

$$f(\mathbf{x}) = \langle f, \Phi(\cdot, \mathbf{x}) \rangle_{\mathcal{N}_\Phi(\Omega)}.$$

We express the interpolant in its cardinal form and apply the reproducing property of Φ . This gives us

$$\begin{aligned} \mathcal{P}_f(\mathbf{x}) &= \sum_{j=1}^N f(\mathbf{x}_j) u_j^*(\mathbf{x}) \\ &= \sum_{j=1}^N u_j^*(\mathbf{x}) \langle f, \Phi(\cdot, \mathbf{x}_j) \rangle_{\mathcal{N}_\Phi(\Omega)} \\ &= \langle f, \sum_{j=1}^N u_j^*(\mathbf{x}) \Phi(\cdot, \mathbf{x}_j) \rangle_{\mathcal{N}_\Phi(\Omega)}. \end{aligned}$$

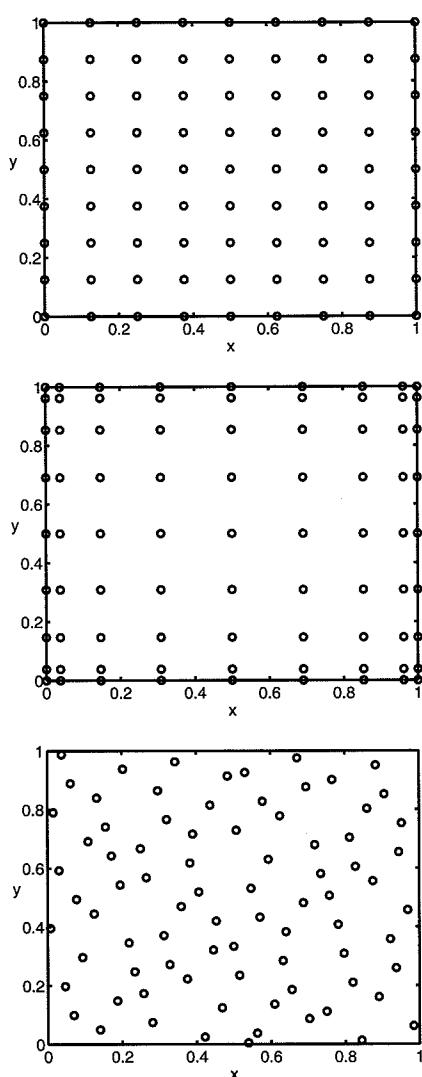


Fig. 14.5 Data sites and power function for Gaussian interpolant with $\varepsilon = 6$ based on $N = 81$ uniformly distributed points (top), tensor-product Chebyshev points (middle), and Halton points (bottom).

Now all that remains to be done is to combine the two formulas just derived and apply the Cauchy-Schwarz inequality. Thus,

$$|f(\mathbf{x}) - \mathcal{P}_f(\mathbf{x})| = \left| \langle f, \Phi(\cdot, \mathbf{x}) - \sum_{j=1}^N u_j^*(\mathbf{x}) \Phi(\cdot, \mathbf{x}_j) \rangle_{\mathcal{N}_\Phi(\Omega)} \right|$$

$$\begin{aligned} &\leq \|f\|_{\mathcal{N}_\Phi(\Omega)} \left\| \Phi(\cdot, \mathbf{x}) - \sum_{j=1}^N u_j^*(\mathbf{x}) \Phi(\cdot, \mathbf{x}_j) \right\|_{\mathcal{N}_\Phi(\Omega)} \\ &= \|f\|_{\mathcal{N}_\Phi(\Omega)} P_{\Phi, X}(\mathbf{x}), \end{aligned}$$

where we have applied (14.2) and the definition of the power function. \square

One of the main benefits of Theorem 14.2 is that we are now able to estimate the interpolation error by considering two independent phenomena:

- the smoothness of the data (measured in terms of the native space norm of f — which is independent of the data locations, but does depend on Φ),
- and the contribution due to the use of the specific kernel (*i.e.*, basic function) Φ and the distribution of the data (measured in terms of the power function — independent of the actual data values).

This is analogous to the standard error estimate for polynomial interpolation cited in most numerical analysis texts. Note, however, that, for any given basic function Φ , a change of the shape parameter ε will have an effect on both terms in the error bound in Theorem 14.2 since the native space norm of f varies with ε .

14.5 Error Estimates in Terms of the Fill Distance

The next step is to refine this error bound by expressing the influence of the data locations in terms of the fill distance. And then, of course, the bound needs to be specialized to various choices of basic functions Φ .

The most common strategy for obtaining error bounds in numerical analysis is to take advantage of the polynomial precision of a method (at least locally), and then to apply a Taylor expansion. With this in mind we observe

Theorem 14.3. Let $\Omega \subseteq \mathbb{R}^s$, and suppose $\Phi \in C(\Omega \times \Omega)$ is strictly positive definite on \mathbb{R}^s . Let $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ be a set of distinct points in Ω , and define the quadratic form $Q(\mathbf{u})$ as in (14.2). The minimum of $Q(\mathbf{u})$ is given for the vector $\mathbf{u} = \mathbf{u}^*(\mathbf{x})$ from Theorem 14.1, *i.e.*,

$$Q(\mathbf{u}^*(\mathbf{x})) \leq Q(\mathbf{u}) \quad \text{for all } \mathbf{u} \in \mathbb{R}^N.$$

Proof. We showed above (see (14.3)) that

$$Q(\mathbf{u}) = \Phi(\mathbf{x}, \mathbf{x}) - 2\mathbf{u}^T \mathbf{b}(\mathbf{x}) + \mathbf{u}^T \mathbf{A} \mathbf{u}.$$

The minimum of this quadratic form is given by the solution of the linear system

$$\mathbf{A} \mathbf{u} = \mathbf{b}(\mathbf{x}).$$

This, however, yields the cardinal functions $\mathbf{u} = \mathbf{u}^*(\mathbf{x})$. \square

In the proof below we will use a special coefficient vector $\tilde{\mathbf{u}}$ which provides the polynomial precision desired for the proof of the refined error estimate. Its existence is guaranteed by the following theorem on *local polynomial reproduction* proved in [Wendland (2005a)]. This theorem requires the notion of a domain that satisfies an interior cone condition.

Definition 14.2. A region $\Omega \subseteq \mathbb{R}^s$ satisfies an *interior cone condition* if there exists an angle $\theta \in (0, \pi/2)$ and a radius $r > 0$ such that for every $\mathbf{x} \in \Omega$ there exists a unit vector $\xi(\mathbf{x})$ such that the cone

$$C = \{\mathbf{x} + \lambda \mathbf{y} : \mathbf{y} \in \mathbb{R}^s, \|\mathbf{y}\|_2 = 1, \mathbf{y}^T \xi(\mathbf{x}) \geq \cos \theta, \lambda \in [0, r]\}$$

is contained in Ω .

A consequence of the interior cone condition is the fact that a domain that satisfies this condition contains balls of a controllable radius. In particular, this will be important when bounding the remainder of the Taylor expansions below. For more details see [Wendland (2005a)].

Existence of an approximation scheme with local polynomial precision is guaranteed by

Theorem 14.4. Suppose $\Omega \subseteq \mathbb{R}^s$ is bounded and satisfies an interior cone condition, and let ℓ be a non-negative integer. Then there exist positive constants h_0 , c_1 , and c_2 such that for all $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subseteq \Omega$ with $h_{\mathcal{X}, \Omega} \leq h_0$ and every $\mathbf{x} \in \Omega$ there exist numbers $\tilde{u}_1(\mathbf{x}), \dots, \tilde{u}_N(\mathbf{x})$ with

- (1) $\sum_{j=1}^N \tilde{u}_j(\mathbf{x}) p(\mathbf{x}_j) = p(\mathbf{x})$ for all polynomials $p \in \Pi_\ell^s$,
- (2) $\sum_{j=1}^N |\tilde{u}_j(\mathbf{x})| \leq c_1$,
- (3) $\tilde{u}_j(\mathbf{x}) = 0$ if $\|\mathbf{x} - \mathbf{x}_j\|_2 \geq c_2 h_{\mathcal{X}, \Omega}$.

Property (1) yields the polynomial precision, and property (3) shows that the scheme is local. The bound in property (2) is essential for controlling the growth of error estimates and the quantity on the left-hand side of (2) is known as the *Lebesgue constant* at \mathbf{x} .

In the following theorem and its proof we will make repeated use of multi-index notation and multivariate Taylor expansions. For $\beta = (\beta_1, \dots, \beta_s) \in \mathbb{N}_0^s$ with $|\beta| = \sum_{i=1}^s \beta_i$ we define the differential operator D^β as

$$D^\beta = \frac{\partial^{|\beta|}}{(\partial x_1)^{\beta_1} \cdots (\partial x_s)^{\beta_s}},$$

and the notation $D_2^\beta \Phi(\mathbf{w}, \cdot)$ used below indicates that the operator is applied to $\Phi(\mathbf{w}, \cdot)$ viewed as a function of the second variable.

The multivariate Taylor expansion of the function $\Phi(\mathbf{w}, \cdot)$ centered at \mathbf{w} is given by

$$\Phi(\mathbf{w}, \mathbf{z}) = \sum_{|\beta| < 2k} \frac{D_2^\beta \Phi(\mathbf{w}, \mathbf{w})}{\beta!} (\mathbf{z} - \mathbf{w})^\beta + R(\mathbf{w}, \mathbf{z})$$

with remainder

$$R(\mathbf{w}, \mathbf{z}) = \sum_{|\beta|=2k} \frac{D_2^\beta \Phi(\mathbf{w}, \xi_{\mathbf{w}, \mathbf{z}})}{\beta!} (\mathbf{z} - \mathbf{w})^\beta,$$

where $\xi_{\mathbf{w}, \mathbf{z}}$ lies somewhere on the line segment connecting \mathbf{w} and \mathbf{z} .

The generic error estimate of Theorem 14.2 can now be formulated in terms of the fill distance.

Theorem 14.5. Suppose $\Omega \subseteq \mathbb{R}^s$ is bounded and satisfies an interior cone condition. Suppose $\Phi \in C^{2k}(\Omega \times \Omega)$ is symmetric and strictly positive definite. Denote the interpolant to $f \in \mathcal{N}_\Phi(\Omega)$ on the set \mathcal{X} by \mathcal{P}_f . Then there exist positive constants h_0 and C (independent of \mathbf{x} , f and Φ) such that

$$|f(\mathbf{x}) - \mathcal{P}_f(\mathbf{x})| \leq Ch_{\mathcal{X}, \Omega}^k \sqrt{C_\Phi(\mathbf{x})} \|f\|_{\mathcal{N}_\Phi(\Omega)},$$

provided $h_{\mathcal{X}, \Omega} \leq h_0$. Here

$$C_\Phi(\mathbf{x}) = \max_{|\beta|=2k} \max_{\mathbf{w}, \mathbf{z} \in \Omega \cap B(\mathbf{x}, c_2 h_{\mathcal{X}, \Omega})} |D_2^\beta \Phi(\mathbf{w}, \mathbf{z})|$$

with $B(\mathbf{x}, c_2 h_{\mathcal{X}, \Omega})$ denoting the ball of radius $c_2 h_{\mathcal{X}, \Omega}$ centered at \mathbf{x} .

Proof. By Theorem 14.2 we know

$$|f(\mathbf{x}) - \mathcal{P}_f(\mathbf{x})| \leq P_{\Phi, \mathcal{X}}(\mathbf{x}) \|f\|_{\mathcal{N}_\Phi(\Omega)}.$$

Therefore, we now derive the bound

$$P_{\Phi, \mathcal{X}}(\mathbf{x}) \leq Ch_{\mathcal{X}, \Omega}^k \sqrt{C_\Phi(\mathbf{x})}$$

for the power function in terms of the fill distance.

We know that the power function is defined by

$$[P_{\Phi, \mathcal{X}}(\mathbf{x})]^2 = Q(\mathbf{u}^*(\mathbf{x})).$$

Moreover, we know from Theorem 14.3 that the quadratic form $Q(\mathbf{u})$ is minimized by $\mathbf{u} = \mathbf{u}^*(\mathbf{x})$. Therefore, any other coefficient vector \mathbf{u} will yield an upper bound on the power function. We take $\mathbf{u} = \tilde{\mathbf{u}}(\mathbf{x})$ from Theorem 14.4 so that we are ensured to have polynomial precision of degree $\ell \geq 2k - 1$.

For this specific choice of coefficients we have

$$[P_{\Phi, \mathcal{X}}(\mathbf{x})]^2 \leq Q(\mathbf{u}) = \Phi(\mathbf{x}, \mathbf{x}) - 2 \sum_j u_j \Phi(\mathbf{x}, \mathbf{x}_j) + \sum_i \sum_j u_i u_j \Phi(\mathbf{x}_i, \mathbf{x}_j),$$

where the sums are over those indices j with $u_j \neq 0$. Now we apply the Taylor expansion centered at \mathbf{x} to $\Phi(\mathbf{x}, \cdot)$ and centered at \mathbf{x}_i to $\Phi(\mathbf{x}_i, \cdot)$, and evaluate both functions at \mathbf{x}_j . This yields

$$\begin{aligned} Q(\mathbf{u}) &= \Phi(\mathbf{x}, \mathbf{x}) - 2 \sum_j u_j \left[\sum_{|\beta|<2k} \frac{D_2^\beta \Phi(\mathbf{x}, \mathbf{x})}{\beta!} (\mathbf{x}_j - \mathbf{x})^\beta + R(\mathbf{x}, \mathbf{x}_j) \right] \\ &\quad + \sum_i \sum_j u_i u_j \left[\sum_{|\beta|<2k} \frac{D_2^\beta \Phi(\mathbf{x}_i, \mathbf{x}_i)}{\beta!} (\mathbf{x}_j - \mathbf{x}_i)^\beta + R(\mathbf{x}_i, \mathbf{x}_j) \right]. \end{aligned}$$

Next, we identify $p(z) = (z - \mathbf{x})^\beta$ so that $p(x) = 0$ unless $\beta = \mathbf{0}$. Therefore the polynomial precision property of the coefficient vector \mathbf{u} simplifies this expression to

$$\begin{aligned} Q(\mathbf{u}) &= \Phi(\mathbf{x}, \mathbf{x}) - 2 \sum_j u_j R(\mathbf{x}, \mathbf{x}_j) \\ &\quad + \sum_i u_i \sum_{|\beta|<2k} \frac{D_2^\beta \Phi(\mathbf{x}_i, \mathbf{x}_i)}{\beta!} (\mathbf{x} - \mathbf{x}_i)^\beta + \sum_i \sum_j u_i u_j R(\mathbf{x}_i, \mathbf{x}_j). \end{aligned} \quad (14.5)$$

Now we can apply the Taylor expansion again and make the observation that

$$\sum_{|\beta|<2k} \frac{D_2^\beta \Phi(\mathbf{x}_i, \mathbf{x}_i)}{\beta!} (\mathbf{x} - \mathbf{x}_i)^\beta = \Phi(\mathbf{x}_i, \mathbf{x}) - R(\mathbf{x}_i, \mathbf{x}). \quad (14.6)$$

If we use (14.6) and rearrange the terms in (14.5) we get

$$\begin{aligned} Q(\mathbf{u}) &= -\Phi(\mathbf{x}, \mathbf{x}) - \sum_j u_j \left[2R(\mathbf{x}, \mathbf{x}_j) - \sum_i u_i R(\mathbf{x}_i, \mathbf{x}_j) \right] \\ &\quad + \sum_i u_i [\Phi(\mathbf{x}_i, \mathbf{x}) - R(\mathbf{x}_i, \mathbf{x})]. \end{aligned} \quad (14.7)$$

One final Taylor expansion we need is (using the symmetry of Φ)

$$\Phi(\mathbf{x}_i, \mathbf{x}) = \Phi(\mathbf{x}, \mathbf{x}_i) = \sum_{|\beta|<2k} \frac{D_2^\beta \Phi(\mathbf{x}, \mathbf{x})}{\beta!} (\mathbf{x}_i - \mathbf{x})^\beta + R(\mathbf{x}, \mathbf{x}_i). \quad (14.8)$$

If we insert (14.8) into (14.7) and once more take advantage of the polynomial precision property of the coefficient vector \mathbf{u} we are left with

$$Q(\mathbf{u}) = - \sum_j u_j \left[R(\mathbf{x}, \mathbf{x}_j) + R(\mathbf{x}_j, \mathbf{x}) - \sum_i u_i R(\mathbf{x}_i, \mathbf{x}_j) \right].$$

Now Theorem 14.4 allows us to bound $\sum_j |u_j| \leq c_1$. Moreover, $\|\mathbf{x} - \mathbf{x}_j\|_2 \leq c_2 h_{\mathcal{X}, \Omega}$ and $\|\mathbf{x}_i - \mathbf{x}_j\|_2 \leq 2c_2 h_{\mathcal{X}, \Omega}$. Therefore, all three remainder terms can be bounded by an expression of the form $C h_{\mathcal{X}, \Omega}^{2k} C_\Phi(\mathbf{x})$. Here we made use of the interior cone property of Ω enabling us to define the term $C_\Phi(\mathbf{x})$. Combining these bounds and taking the square root yields the stated bound for the power function. \square

Theorem 14.5 says that interpolation with a C^{2k} smooth kernel Φ has approximation order k . Thus, for infinitely smooth strictly positive definite functions such as the Gaussians, Laguerre-Gaussians, Poisson radial functions, and the generalized inverse multiquadratics we see that the approximation order k is arbitrarily high. For strictly positive definite functions with limited smoothness such as the Matérn functions, the Whittaker radial functions, as well as all of the compactly supported functions, the approximation order is limited by the smoothness of the basic function.

The estimate in Theorem 14.5 is still generic. It does not fully account for the particular basic function Φ being used for the interpolation since the factor $C_\Phi(\mathbf{x})$ still depends on Φ . Moreover, we point out that the term $C_\Phi(\mathbf{x})$ may include a hidden dependence on $h_{\mathcal{X}, \Omega}$. For most basic functions it will be possible to use $C_\Phi(\mathbf{x})$ to “squeeze out” additional powers of h . This is the reason for splitting the constant in front of the h -power into a generic C and a $C_\Phi(\mathbf{x})$.

The statement of Theorem 14.5 can be generalized for strictly conditionally positive definite functions and also to cover the error for approximating the derivatives of f by derivatives of \mathcal{P}_f . We state this general theorem without comment (c.f. [Wendland (2005a)] for details).

Theorem 14.6. Suppose $\Omega \subseteq \mathbb{R}^s$ is open and bounded and satisfies an interior cone condition. Suppose $\Phi \in C^{2k}(\Omega \times \Omega)$ is symmetric and strictly conditionally positive definite of order m on \mathbb{R}^s . Denote the interpolant to $f \in \mathcal{N}_\Phi(\Omega)$ on the $(m-1)$ -unisolvent set \mathcal{X} by \mathcal{P}_f . Fix $\alpha \in \mathbb{N}_0^s$ with $|\alpha| \leq k$. Then there exist positive constants h_0 and C (independent of \mathbf{x} , f and Φ) such that

$$|D^\alpha f(\mathbf{x}) - D^\alpha \mathcal{P}_f(\mathbf{x})| \leq Ch_{\mathcal{X}, \Omega}^{k-|\alpha|} \sqrt{C_\Phi(\mathbf{x})} |f|_{\mathcal{N}_\Phi(\Omega)},$$

provided $h_{\mathcal{X}, \Omega} \leq h_0$. Here

$$C_\Phi(\mathbf{x}) = \max_{\substack{\beta, \gamma \in \mathbb{N}_0^s \\ |\beta|+|\gamma|=2k}} \max_{w, z \in \Omega \cap B(\mathbf{x}, c_2 h_{\mathcal{X}, \Omega})} |D_1^\beta D_2^\gamma \Phi(w, z)|.$$

Note that for conditionally positive definite functions we have only a native space semi-norm instead of a norm.

Refined and Improved Error Bounds

15.1 Native Space Error Bounds for Specific Basis Functions

For the first part of this chapter we discuss the non-stationary setting. The additional refinement of the error estimate of Theorem 14.6 for specific functions Φ is rather technical (for details see, *e.g.*, the book [Wendland (2005a)]). A large body of literature exists on this topic such as, *e.g.*, [Buhmann and Dyn (1991); Light (1996); Light and Wayne (1995); Light and Wayne (1998); Madych (1992); Madych and Nelson (1992); Narcowich and Ward (2004); Narcowich *et al.* (2003); Narcowich *et al.* (2005); Schaback (1995b); Schaback (1996); Wendland (1998); Wendland (1997); Wu and Schaback (1993); Yoon (2003)]). We now list some of the results that can be obtained.

15.1.1 Infinitely Smooth Basis Functions

As mentioned before, an application of Theorem 14.6 to infinitely smooth functions such as Gaussians or generalized (inverse) multiquadratics immediately yields arbitrarily high algebraic convergence rates, *i.e.*, for every $\ell \in \mathbb{N}$ and $|\alpha| \leq \ell$ we have

$$|D^\alpha f(\mathbf{x}) - D^\alpha \mathcal{P}_f(\mathbf{x})| \leq C_\ell h^{\ell - |\alpha|} |f|_{\mathcal{N}_\Phi(\Omega)}. \quad (15.1)$$

whenever $f \in \mathcal{N}_\Phi(\Omega)$. A considerable amount of work has gone into investigating the dependence of the constant C_ℓ on ℓ (see, *e.g.*, [Wendland (2001b)]).

Using different proof techniques (see, *e.g.*, [Madych and Nelson (1988)] or [Wendland (2005a)] for details) it is possible to derive more precise error bounds for Gaussians and (inverse) multiquadratics. The resulting theorem from [Wendland (2005a)] is

Theorem 15.1. *Let Ω be a cube in \mathbb{R}^s . Suppose that $\Phi = \varphi(\|\cdot\|)$ is a strictly conditionally positive definite radial function such that $\psi = \varphi(\sqrt{\cdot})$ satisfies $|\psi^{(\ell)}(r)| \leq \ell! M^\ell$ for all integers $\ell \geq \ell_0$ and all $r \geq 0$, where M is a fixed positive constant. Then there exists a constant c such that for any $f \in \mathcal{N}_\Phi(\Omega)$*

$$\|f - \mathcal{P}_f\|_{L_\infty(\Omega)} \leq e^{\frac{-c}{h^s \varphi(\Omega)}} |f|_{\mathcal{N}_\Phi(\Omega)}, \quad (15.2)$$

for all data sites \mathcal{X} with sufficiently small fill distance $h_{\mathcal{X}, \Omega}$.

Moreover, if ψ satisfies even $|\psi^{(\ell)}(r)| \leq M^\ell$, then

$$\|f - \mathcal{P}_f\|_{L_\infty(\Omega)} \leq e^{\frac{-c|\log h_{\mathcal{X}, \Omega}|}{h_{\mathcal{X}, \Omega}}} \|f\|_{\mathcal{N}_\Phi(\Omega)} \quad (15.3)$$

provided $h_{\mathcal{X}, \Omega}$ is sufficiently small.

Example 15.1. For Gaussians $\Phi(\mathbf{x}) = e^{-\varepsilon^2 \|\mathbf{x}\|^2}$, $\varepsilon > 0$ fixed, we have $\psi(r) = e^{-\varepsilon^2 r}$, so that $\psi^{(\ell)}(r) = (-1)^\ell \varepsilon^{2\ell} e^{-\varepsilon^2 r}$ for $\ell \geq \ell_0 = 0$. Thus, $M = \varepsilon^2$, and the error bound (15.3) applies. This kind of exponential approximation order is usually referred to as *spectral* (or even super-spectral) approximation order. We emphasize that this nice property holds only in the non-stationary setting and for data functions f that are in the native space of the Gaussians such as band-limited functions.

Example 15.2. For generalized (inverse) multiquadratics $\Phi(\mathbf{x}) = (1 + \|\mathbf{x}\|^2)^\beta$, $\beta < 0$, or $0 < \beta \notin \mathbb{N}$, we have $\psi(r) = (1+r)^\beta$. In this case one can show that $|\psi^\ell(r)| \leq \ell! M^\ell$ whenever $\ell \geq \lceil \beta \rceil$. Here $M = 1 + |\beta + 1|$. Therefore, the error estimate (15.2) applies, i.e., in the non-stationary setting generalized (inverse) multiquadratics have spectral approximation order.

Example 15.3. For Laguerre-Gaussians $\Phi(\mathbf{x}) = L_n^{s/2}(\|\varepsilon \mathbf{x}\|^2) e^{-\varepsilon^2 \|\mathbf{x}\|^2}$, $\varepsilon > 0$ fixed, we have $\psi(r) = L_n^{s/2}(\varepsilon^2 r) e^{-\varepsilon^2 r}$ and the derivatives $\psi^{(\ell)}$ will be bounded by $\psi^{(\ell)}(0) = p_n(\ell) \varepsilon^{2\ell}$, where p_n is a polynomial of degree n . Thus, the approximation power of Laguerre-Gaussians falls between (15.3) and (15.2) and Laguerre-Gaussians have at least spectral approximation power.

15.1.2 Basis Functions with Finite Smoothness

For functions with finite smoothness (such as the Matérn functions, radial powers, thin plate splines, and Wendland's compactly supported functions) it is possible to bound the constant $C_\Phi(\mathbf{x})$ by some additional powers of h , and thereby to improve the order predicted by Theorem 14.6. In particular, for C^k functions the factor $C_\Phi(\mathbf{x})$ can be expressed as

$$C_\Phi(\mathbf{x}) = \max_{|\beta|=2k} \|D^\beta \Phi\|_{L_\infty(B(0, 2ch_{\mathcal{X}, \Omega}))}$$

independent of \mathbf{x} (see [Wendland (2005a)]). Therefore, this results in the following error estimates (see, e.g., [Wendland (2005a)], or the much earlier [Wu and Schaback (1993)] where other proof techniques were used).

Example 15.4. For the Matérn functions $\Phi(\mathbf{x}) = \frac{K_{\beta-\frac{s}{2}}(\|\mathbf{x}\|) \|\mathbf{x}\|^{\beta-\frac{s}{2}}}{2^{\beta-1} \Gamma(\beta)}$, $\beta > \frac{s}{2}$, we get

$$|D^\alpha f(\mathbf{x}) - D^\alpha \mathcal{P}_f(\mathbf{x})| \leq Ch_{\mathcal{X}, \Omega}^{\beta-\frac{s}{2}-|\alpha|} |f|_{\mathcal{N}_\Phi(\Omega)}. \quad (15.4)$$

provided $|\alpha| \leq \beta - \lceil \frac{s+1}{2} \rceil$, $h_{\mathcal{X}, \Omega}$ is sufficiently small, and $f \in \mathcal{N}_\Phi(\Omega)$.

Example 15.5. For the compactly supported Wendland functions $\Phi_{s,k}(\mathbf{x}) = \varphi_{s,k}(\|\mathbf{x}\|)$ this first refinement leads to

$$|D^\alpha f(\mathbf{x}) - D^\alpha \mathcal{P}_f(\mathbf{x})| \leq Ch_{\mathcal{X}, \Omega}^{k+\frac{1}{2}-|\alpha|} \|f\|_{\mathcal{N}_\Phi(\Omega)}. \quad (15.5)$$

provided $|\alpha| \leq k$, $h_{\mathcal{X}, \Omega}$ is sufficiently small, and $f \in \mathcal{N}_\Phi(\Omega)$.

Example 15.6. For the radial powers $\Phi(\mathbf{x}) = (-1)^{\lceil \beta/2 \rceil} \|\mathbf{x}\|^\beta$, $0 < \beta \notin 2\mathbb{N}$, we get

$$|D^\alpha f(\mathbf{x}) - D^\alpha \mathcal{P}_f(\mathbf{x})| \leq Ch_{\mathcal{X}, \Omega}^{\frac{\beta}{2}-|\alpha|} |f|_{\mathcal{N}_\Phi(\Omega)}. \quad (15.6)$$

provided $|\alpha| \leq \frac{\lceil \beta \rceil - 1}{2}$, $h_{\mathcal{X}, \Omega}$ is sufficiently small, and $f \in \mathcal{N}_\Phi(\Omega)$.

Example 15.7. For thin plate splines $\Phi(\mathbf{x}) = (-1)^{k+1} \|\mathbf{x}\|^{2k} \log \|\mathbf{x}\|$, we get

$$|D^\alpha f(\mathbf{x}) - D^\alpha \mathcal{P}_f(\mathbf{x})| \leq Ch_{\mathcal{X}, \Omega}^{k-|\alpha|} |f|_{\mathcal{N}_\Phi(\Omega)}. \quad (15.7)$$

provided $|\alpha| \leq k-1$, $h_{\mathcal{X}, \Omega}$ is sufficiently small, and $f \in \mathcal{N}_\Phi(\Omega)$.

15.2 Improvements for Native Space Error Bounds

Radial powers and thin plate splines can be interpreted as a generalization of univariate natural splines. Therefore, we know that the approximation order estimates obtained via the native space approach are not optimal. For example, for interpolation with univariate piecewise linear splines (i.e., $\Phi(\mathbf{x}) = \|\mathbf{x}\|$ in $\mathbf{x} \in \mathbb{R}$) we know the approximation order to be $\mathcal{O}(h)$, whereas the estimate (15.6) yields only approximation order $\mathcal{O}(h^{1/2})$. Similarly, for thin plate splines $\Phi(\mathbf{x}) = \|\mathbf{x}\|^2 \log \|\mathbf{x}\|$ one would expect order $\mathcal{O}(h^2)$ in the case of pure function approximation. However, the estimate (15.7) yields only $\mathcal{O}(h)$. These two examples suggest that it should be possible to “double” the approximation orders obtained thus far.

One can improve the estimates for functions with finite smoothness (i.e., Matérn functions, Wendland functions, radial powers, and thin plate splines) by either (or both) of the following two ideas:

- by requiring the data function f to be even smoother than what the native space prescribes, i.e., by building certain boundary conditions into the native space;
- by using weaker norms to measure the error.

The idea to localize the data by adding boundary conditions was introduced in the paper [Light and Wayne (1998)]. This “trick” allows us to “square” the approximation order, and thus reach the expected approximation orders. The second idea can already be found in the early paper [Duchon (1978)].

After applying both of these techniques the final approximation order estimate for interpolation with the compactly supported functions $\Phi_{s,k}$ is (see [Wendland (1997)])

$$\|f - \mathcal{P}_f\|_{L_2(\Omega)} \leq Ch^{2k+1+s} \|f\|_{W_2^{2k+1+s}(\mathbb{R}^s)}, \quad (15.8)$$

where f is assumed to lie in the subspace $W_2^{2k+1+s}(\mathbb{R}^s)$ of $\mathcal{N}_\Phi(\mathbb{R}^s) = W_2^{k+\frac{s+1}{2}}$. For example, for the popular basic function $\varphi_{3,1}(r) = (1-r)_+^4(4r+1)$ we have

$$\|f - \mathcal{P}_f\|_{L_2(\Omega)} \leq Ch^6 \|f\|_{W_2^6(\mathbb{R}^s)}.$$

Note that the numerical experiments in Table 12.2 produced RMS-convergence rates only as high as 4.5.

For radial powers and thin plate splines one obtains L_2 -error estimates of order $\mathcal{O}(h^{\beta+s})$ and $\mathcal{O}(h^{2k+s})$, respectively. These estimates are optimal, *i.e.*, exact approximation orders, as shown in [Bejancu (1999)].

More work on improved error bounds can be found in, *e.g.*, [Johnson (2004a)] or [Schaback (1999b)].

15.3 Error Bounds for Functions Outside the Native Space

The error bounds mentioned so far were all valid under the assumption that the function f providing the data came from (a subspace of) the native space of the RBF employed in the interpolation. We now mention a few recent results that provide error bounds for interpolation of functions f *not* in the native space of the basic function. In particular, the case when f lies in some Sobolev space is of great interest. A rather general theorem was recently given in [Narcowich *et al.* (2005)]. In this theorem Narcowich, Ward and Wendland provide Sobolev bounds for functions with many zeros. However, since the interpolation error function is just such a function, these bounds have a direct application to our situation. We point out that this theorem again applies to the non-stationary setting.

Theorem 15.2. *Let k be a positive integer, $0 < \sigma \leq 1$, $1 \leq p < \infty$, $1 \leq q \leq \infty$ and let α be a multi-index satisfying $k > |\alpha| + s/p$ or, for $p = 1$, $k \geq |\alpha| + s$. Let $\mathcal{X} \subset \Omega$ be a discrete set with fill distance $h = h_{\mathcal{X}, \Omega}$ where Ω is a compact set with Lipschitz boundary which satisfies an interior cone condition. If $u \in W_p^{k+\sigma}(\Omega)$ satisfies $u|_{\mathcal{X}} = 0$, then*

$$|u|_{W_q^{|\alpha|}(\Omega)} \leq ch^{k+\sigma-|\alpha|-s(1/p-1/q)_+} |u|_{W_p^{k+\sigma}(\Omega)},$$

where c is a constant independent of u and h , and $(x)_+$ is the cutoff function.

Suppose we have an interpolation process $\mathcal{P} : W_p^{k+\sigma}(\Omega) \rightarrow V$ that maps Sobolev functions to a finite-dimensional subspace V of $W_p^{k+\sigma}(\Omega)$ with the additional property $|\mathcal{P}_f|_{W_p^{k+\sigma}(\Omega)} \leq |f|_{W_p^{k+\sigma}(\Omega)}$, then Theorem 15.2 immediately yields the error estimate

$$|f - \mathcal{P}_f|_{W_q^{|\alpha|}(\Omega)} \leq ch^{k+\sigma-|\alpha|-s(1/p-1/q)_+} |f|_{W_p^{k+\sigma}(\Omega)}.$$

The additional property $|\mathcal{P}_f|_{W_p^{k+\sigma}(\Omega)} \leq |f|_{W_p^{k+\sigma}(\Omega)}$ is certainly satisfied provided the native space of the basic function is a Sobolev space. Thus, Theorem 15.2 provides an alternative to the power function approach discussed in the previous

chapter if we base \mathcal{P} on linear combinations of shifts of the basic function Φ . This new approach has the advantage that the term $C_\Phi(\mathbf{x})$ which may depend on both Φ and \mathcal{X} no longer needs to be dealt with.

In particular, the authors of [Narcowich *et al.* (2005)] show that if the Fourier transform of Φ satisfies

$$c_1(1 + \|\omega\|_2^2)^{-\tau} \leq \hat{\Phi}(\omega) \leq c_2(1 + \|\omega\|_2^2)^{-\tau}, \quad \|\omega\| \rightarrow \infty, \omega \in \mathbb{R}^s, \quad (15.9)$$

then the above error estimate holds with $\tau = k + \sigma$ and $p = 2$ provided the fill distance is sufficiently small. Examples of basic functions with an appropriately decaying Fourier transform are provided by the families of Wendland or Matérn functions. In addition, Narcowich, Ward and Wendland show that analogous error bounds hold for radial powers and thin plate splines (whose native spaces are Beppo-Levi spaces).

For functions f *outside* the native space of a basic function Φ whose Fourier transform satisfies (15.9) Narcowich, Ward and Wendland prove

Theorem 15.3. *Let k and n be integers with $0 \leq n < k \leq \tau$ and $k > s/2$, and let $f \in C^k(\bar{\Omega})$. Also suppose that $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \Omega$ satisfies $\text{diam}(\mathcal{X}) \leq 1$ with sufficiently small fill distance. Then for any $1 \leq q \leq \infty$ we have*

$$|f - \mathcal{P}_f|_{W_q^n(\Omega)} \leq c\rho_{\mathcal{X}}^{\tau-k} h^{k-n-s(1/2-1/q)_+} \|f\|_{C^k(\bar{\Omega})},$$

where $\rho_{\mathcal{X}} = \frac{h}{q_{\mathcal{X}}}$ is the mesh ratio for \mathcal{X} and $q_{\mathcal{X}}$ is the separation distance $\frac{1}{2} \min_{i \neq j} \|\mathbf{x}_i - \mathbf{x}_j\|_2$.

We remind the reader that the fill distance corresponds to the radius of the largest possible empty ball that can be placed between the points in \mathcal{X} . The separation distance (*c.f.* Chapter 16), on the other hand, can be interpreted as the radius of the largest ball that can be placed around every point in \mathcal{X} such that no two balls overlap. Thus, the mesh ratio is a measure of the non-uniformity of the distribution of the points in \mathcal{X} .

Similar results were obtained earlier in [Brownlee and Light (2004)] (for radial powers and thin plate splines only), and in [Yoon (2003)] (for shifted surface splines, see below).

Example 15.8. If we consider polyharmonic splines, then the decay condition (15.9) for the Fourier transform is satisfied with $\tau = 2\beta$ for thin plate splines and with $\tau = \beta$ for radial powers. If we take $k = \tau$, $n = 0$, and $q = \infty$ in Theorem 15.3 then we arrive at the bound

$$|f - \mathcal{P}_f|_{L_\infty} \leq ch^{2\beta-s/2} \|f\|_{C^{2\beta}(\bar{\Omega})}$$

for thin plate splines $\Phi(\mathbf{x}) = \|\mathbf{x}\|^{2\beta} \log(\|\mathbf{x}\|)$, and

$$|f - \mathcal{P}_f|_{L_\infty} \leq ch^{\beta-s/2} \|f\|_{C^\beta(\bar{\Omega})}$$

for radial powers $\Phi(\mathbf{x}) = \|\mathbf{x}\|^\beta$. These bounds immediately correspond to the “optimal” native space bounds obtained earlier only after the improvements discussed

in the previous subsection. For data functions f with less smoothness the approximation order is reduced accordingly.

Lower bounds on the approximation order for approximation by polyharmonic splines were recently provided in [Maiorov (2005)]. Maiorov studies for any $1 \leq p, q \leq \infty$ and $\beta/s > (1/p - 1/q)_+$ the error E of L_q -approximation of W_p^β functions by polyharmonic splines. More precisely,

$$E(W_p^\beta([0, 1]^s), \mathcal{R}_N(\varphi_\beta, \beta), L_q([0, 1]^s)) \geq cN^{-\beta/s},$$

where $\mathcal{R}_N(\varphi_\beta, \beta)$ denotes the linear space formed by all possible linear combinations of N polyharmonic (or thin-plate type) splines

$$\varphi_\beta(r) = \begin{cases} r^{2\beta-s} & \text{if } s \text{ is odd} \\ r^{2\beta-s} \log r & \text{if } s \text{ is even,} \end{cases} \quad \beta > \frac{s}{2},$$

and multivariate polynomials of degree at most $\beta - 1$. Note that these bounds are in terms of the number N of data sites instead of the usual fill distance h .

For the special cases $p = q = \infty$ and $p = 2, 1 \leq q \leq 2$ the above lower bound is shown to be asymptotically exact.

15.4 Error Bounds for Stationary Approximation

The stationary setting is a natural approach for use with local basis functions. The main motivation comes from the computational point of view. We are interested in maintaining sparse interpolation matrices as the density of the data increases. This can be achieved by scaling the basis functions proportional to the data density. In principle we can take any of our basic functions and apply a scaling of the variable, *i.e.*, we replace \mathbf{x} by $\varepsilon \mathbf{x}$, $\varepsilon > 0$. As mentioned several times earlier, this scaling results in “peaked” or “narrow” basis functions for large values of ε , and “flat” basis functions for $\varepsilon \rightarrow 0$. We will now discuss what happens if we choose ε inversely proportional to the fill distance, *i.e.*,

$$\varepsilon_h = \frac{\varepsilon_0}{h_{\mathcal{X}, \Omega}} \tag{15.10}$$

for some fixed base scale ε_0 and study the approximation error based on the RBF interpolant

$$\mathcal{P}_f(\mathbf{x}) = \sum_{j=1}^N c_j \varphi_{\varepsilon_h}(\|\mathbf{x} - \mathbf{x}_j\|),$$

where

$$\varphi_{\varepsilon_h} = \varphi(\varepsilon_h \cdot).$$

Example 15.9. A rather disappointing fact is that *Gaussians do not provide any positive approximation order*, *i.e.*, the approximation process is saturated. This was

studied by [Buhmann (1989a)] on infinite lattices. However, for quasi-interpolation the *approximate approximation* approach of Maz’ya shows that it is possible to choose ε_0 in such a way that the level at which the saturation occurs can be controlled (see, *e.g.*, [Maz’ya and Schmidt (1996)]). Therefore, Gaussians may very well be used for stationary interpolation provided an appropriate initial shape parameter is chosen. We will illustrate this behavior in the next chapter. The same kind of argument also applies to the Laguerre-Gaussians of Section 4.2.

Example 15.10. Basis functions with compact support such as the Wendland functions also do not provide any positive approximation order in the stationary case. This can be seen by looking at the power function for the scaled basic function $\Phi_{\varepsilon_h} = \Phi(\varepsilon_h \cdot)$ which is of the form $P_{\Phi_{\varepsilon_h}, \mathcal{X}}(\mathbf{x}) = P_{\Phi, \mathcal{X}_{\varepsilon_h}}(\varepsilon_h \mathbf{x})$ where $\mathcal{X}_{\varepsilon_h} = \{\varepsilon_h \mathbf{x}_1, \dots, \varepsilon_h \mathbf{x}_N\}$ denotes the scaled data set. Moreover, the fill distances of the sets $\mathcal{X}_{\varepsilon_h}$ and \mathcal{X} satisfy $h_{\mathcal{X}_{\varepsilon_h}, \Omega} = \varepsilon_h h_{\mathcal{X}, \frac{\Omega}{\varepsilon_h}}$. Therefore, the power function (which can be bounded in terms of the fill distance, *c.f.* the proof of Theorem 14.5) satisfies

$$P_{\Phi_{\varepsilon_h}, \mathcal{X}}(\mathbf{x}) \leq C \left(\varepsilon_h h_{\mathcal{X}, \frac{\Omega}{\varepsilon_h}} \right)^\alpha$$

for some $\alpha > 0$. This, however, does not go to zero if ε_h is chosen as in (15.10).

If, on the other hand, we work in the approximate approximation regime, then we can obtain good convergence in many cases (see the next chapter for some numerical experiments).

Example 15.11. Stationary interpolation with (inverse) multiquadratics, radial powers and thin plate splines presents no difficulties. In fact, [Schaback (1995c)] shows that the native space error bound for thin plate splines and radial powers is invariant under a stationary scaling. Therefore, the non-stationary bound of Theorem 15.3 applies in the stationary case also. The advantage of scaling thin plate splines or radial powers comes from the added stability one can gain by preventing the separation distance from becoming too small (see Chapter 16 and the work of Iske on local polyharmonic spline approximation, *e.g.*, [Iske (2004)]).

Yoon provides error estimates for stationary approximation of rough functions (*i.e.*, functions that are not in the native space of the basic function) by so-called *shifted surface splines*. Shifted surface splines are of the form

$$\Phi(\mathbf{x}) = \begin{cases} (-1)^{\lceil \beta - s/2 \rceil} (1 + \|\mathbf{x}\|^2)^{\beta - s/2}, & s \text{ odd,} \\ (-1)^{\beta - s/2 + 1} (1 + \|\mathbf{x}\|^2)^{\beta - s/2} \log(1 + \|\mathbf{x}\|^2)^{1/2}, & s \text{ even,} \end{cases}$$

where $s/2 < \beta \in \mathbb{N}$. These functions include all of the (inverse) multiquadratics, radial powers and thin plate splines.

Yoon has the following theorem (see [Yoon (2003)] for the L_p case, and [Yoon (2001)] for L_∞ bounds only).

Theorem 15.4. Let Φ be a shifted surface spline with shape parameter ε inversely proportional to the fill distance $h_{\mathcal{X}, \Omega}$. Then there exists a positive constant C (independent of \mathcal{X}) such that for every f in the Sobolev space $W_2^\beta(\Omega) \cap W_\infty^\beta(\Omega)$ we

have

$$\|f - \mathcal{P}_f\|_{L_p(\Omega)} \leq Ch^{\gamma_p} |f|_{W_2^\beta(\mathbb{R}^s)}, \quad 1 \leq p \leq \infty,$$

with

$$\gamma_p = \min\{\beta, \beta - s/2 + s/p\}.$$

Furthermore, if $f \in W_2^\alpha(\Omega) \cap W_\infty^\alpha(\Omega)$ with $\max\{0, s/2 - s/p\} < \alpha < \beta$, then

$$\|f - \mathcal{P}_f\|_{L_p(\Omega)} = o(h^{\gamma_p - \beta + \alpha}).$$

Yoon's estimates address the question of how well the infinitely smooth (inverse) multiquadratics approximate functions that are less smooth than those in their native space. For example, Theorem 15.4 states that L_2 -approximation to functions in $W_2^2(\Omega)$, $\Omega \subseteq \mathbb{R}^s$, by multiquadratics $\Phi_\varepsilon(\mathbf{x}) = \sqrt{1 + \|\varepsilon\mathbf{x}\|^2}$ is of the order $\mathcal{O}(h^2)$. However, we emphasize once more that this refers to stationary approximation of rough functions, *i.e.*, ε is scaled inversely proportional to the fill distance and f need not lie in the native space of Φ , whereas the spectral order given in (15.2) corresponds to approximation of functions in the native space in the non-stationary case with fixed ε .

For thin plate splines and radial powers the approximation orders in Theorem 15.4 are equivalent to those of Theorem 15.3 and the results of Brownlee and Light mentioned above. This is to be expected due to the invariance of these basic functions with respect to scaling.

The second part of Yoon's result is a step toward *exact approximation orders* as is the work of [Maiorov (2005)] and [Bejancu (1999)] mentioned above.

15.5 Convergence with Respect to the Shape Parameter

None of the error bounds discussed thus far have taken into account the possibility of varying the shape parameter ε for a fixed data set \mathcal{X} . However, in the literature the infinitely smooth basic functions such as the Gaussians and (inverse) multiquadratics are usually formulated including the shape parameter ε (or another parameter equivalent to it) and one may wonder how a change in this shape parameter affects the convergence properties of the RBF interpolant. In fact, quite a bit of work has been spent on the quest for the "optimal" shape parameter (see, *e.g.*, [Carlson and Foley (1991); Foley (1994); Hagan and Kansa (1994); Kansa and Carlson (1992); Rippa (1999); Tarwater (1985); Wertz *et al.* (2006)]).

Convergence of the infinitely smooth Gaussians and (inverse) multiquadratics with respect to the shape parameter was studied early on in [Madych (1991)]. Madych showed that for these basic functions there exists a positive constant $\lambda < 1$ such that

$$|f(\mathbf{x}) - \mathcal{P}_f(\mathbf{x})| \leq C\lambda^{1/(\varepsilon h_{\mathcal{X},\Omega})} \quad (15.11)$$

provided f is in the native space of Φ . This estimate shows that taking either the shape parameter ε or the fill distance $h_{\mathcal{X},\Omega}$ to zero results in exponential convergence.

15.6 Polynomial Interpolation as the Limit of RBF Interpolation

Recently, a number of authors (see, *e.g.*, [Driscoll and Fornberg (2002); Fornberg and Flyer (2005); Fornberg and Wright (2004); Larsson and Fornberg (2005); Schaback (2005); Schaback (2006b)]) have studied the limiting case as $\varepsilon \rightarrow 0$ of scaled radial basis function interpolation with infinitely smooth basic functions such as Gaussians and generalized (inverse) multiquadratics. It turns out that there is an interesting connection to polynomial interpolation.

In [Driscoll and Fornberg (2002)] univariate ($s = 1$) interpolation with ε -scaled infinitely smooth radial basic functions is studied. Driscoll and Fornberg show that the RBF interpolant

$$\mathcal{P}_f(x) = \sum_{j=1}^N c_j \varphi(\|\varepsilon(x - x_j)\|), \quad x \in [a, b] \subset \mathbb{R},$$

to function values at N distinct data sites tends to the Lagrange interpolating polynomial of f as $\varepsilon \rightarrow 0$.

The multivariate case is more complicated. However, the limiting RBF interpolant (provided it exists) is given by a low-degree multivariate polynomial (see [Larsson and Fornberg (2005); Schaback (2005); Schaback (2006b)]). For example, if the data sites are located such that they guarantee a unique polynomial interpolant, then the limiting RBF interpolant is given by this polynomial. If polynomial interpolation is not unique, then the RBF limit depends on the choice of basic function. However, these statements require the RBFs to satisfy an (unproven) condition on certain coefficient matrices $A_{p,J}$. In [Larsson and Fornberg (2005)] the authors also provide an explanation for the error behavior for small values of the shape parameter, and for the existence of an optimal (positive) value of ε giving rise to a global minimum of the error function. For the special case of scaled Gaussians Schaback [Schaback (2005)] shows that the RBF interpolant converges to the de Boor and Ron least polynomial interpolant (see [de Boor and Ron (1990); de Boor and Ron (1992a)] and also [de Boor (2006)]) as $\varepsilon \rightarrow 0$.

In [Fornberg and Wright (2004)] the authors describe a so-called *Contour-Padé* algorithm that makes it possible (for data sets of relatively modest size) to compute the RBF interpolant for all values of the shape parameter ε including the limiting case $\varepsilon \rightarrow 0$. We present some numerical result obtained with Grady Wright's MATLAB toolbox in Chapter 17.

We will later exploit the connection between RBF and polynomial interpolants to design numerical solvers for partial differential equations.

Chapter 16

Stability and Trade-Off Principles

16.1 Stability and Conditioning of Radial Basis Function Interpolants

A standard criterion for measuring the numerical stability of an approximation method is its condition number. In particular, for radial basis function interpolation we need to look at the condition number of the interpolation matrix A with entries $A_{ij} = \Phi(\mathbf{x}_i - \mathbf{x}_j)$. For any matrix A its ℓ_2 -condition number is given by

$$\text{cond}(A) = \|A\|_2 \|A^{-1}\|_2 = \frac{\sigma_{\max}}{\sigma_{\min}},$$

where σ_{\max} and σ_{\min} are the largest and smallest singular values of A . If we concentrate on positive definite matrices A , then the condition number of A can also take be computed as the ratio

$$\frac{\lambda_{\max}}{\lambda_{\min}}$$

of the largest and smallest eigenvalues.

What do we know about these eigenvalues? First, Gershgorin's theorem (see, e.g., [Meyer (2000)]) says that

$$|\lambda_{\max} - A_{ii}| \leq \sum_{\substack{j=1 \\ j \neq i}}^N |A_{ij}|$$

for some $i \in \{1, \dots, N\}$. Therefore,

$$\lambda_{\max} \leq N \max_{i,j=1,\dots,N} |A_{ij}| = N \max_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}} |\Phi(\mathbf{x}_i - \mathbf{x}_j)|,$$

which, since Φ is strictly positive definite, becomes

$$\lambda_{\max} \leq N \Phi(\mathbf{0})$$

by the properties of positive definite functions (Property (4) in Theorem 3.1). Now, as long as the data are not too wildly distributed, N will grow as $h_{\mathcal{X}, \Omega}^{-s}$ which is acceptable. Therefore, the main work in establishing a bound for the condition number of A lies in finding lower bounds for λ_{\min} (or correspondingly upper bounds

for the norm of the inverse $\|A^{-1}\|_2$). This is the subject of several papers by Ball, Narcowich, Sivakumar and Ward [Ball *et al.* (1992); Narcowich *et al.* (1994); Narcowich and Ward (1991a); Narcowich and Ward (1991b); Narcowich and Ward (1992)] who make use of a result from [Ball (1992)] on eigenvalues of distance matrices. Ball's result follows from the *Rayleigh quotient* (or the Courant-Fischer Theorem 9.5), which gives the smallest eigenvalue of a symmetric positive definite matrix as

$$\lambda_{\min} = \min_{c \in \mathbb{R}^N \setminus \{0\}} \frac{c^T A c}{c^T c}.$$

This can be used to prove the following bound for the norm of the inverse of A which covers also the case of conditional positive (negative) definiteness of order one.

Lemma 16.1. *Let $\mathbf{x}_1, \dots, \mathbf{x}_N$ be distinct points in \mathbb{R}^s and let $\Phi : \mathbb{R}^s \rightarrow \mathbb{R}$ be either strictly positive definite, or strictly conditionally negative definite of order one with $\Phi(\mathbf{0}) \leq 0$. Also, let A be the interpolation matrix with entries $A_{ij} = \Phi(\mathbf{x}_i - \mathbf{x}_j)$. If the inequality*

$$\sum_{i=1}^N \sum_{j=1}^N c_i c_j A_{ij} \geq \theta \|c\|_2^2$$

is satisfied whenever the components of c satisfy $\sum_{j=1}^N c_j = 0$, then

$$\|A^{-1}\|_2 \leq \theta^{-1}.$$

Note that for positive definite matrices the Rayleigh quotient implies $\theta = \lambda_{\min}$ which shows why lower bounds on the smallest eigenvalue correspond to upper bounds on the norm of the inverse of A . In order to obtain the bound for conditionally negative definite matrices the Courant-Fischer theorem 9.5 needs to be employed.

The bound in Lemma 16.1 is too generic to give us any information for specific basic functions Φ . This extension was accomplished in some of the other papers mentioned above. Narcowich and Ward establish bounds on the norm of the inverse of A in terms of the *separation distance* of the data sites

$$q_{\mathcal{X}} = \frac{1}{2} \min_{i \neq j} \|\mathbf{x}_i - \mathbf{x}_j\|_2.$$

We can picture $q_{\mathcal{X}}$ as the radius of the largest ball that can be placed around every point in \mathcal{X} such that no two balls overlap (see Figure 16.1). The separation distance is sometimes also referred to as the *packing radius*. In our MATLAB code we can compute the separation distance via

```
qX = min(min(DM_data+eye(size(DM_data))))/2
```

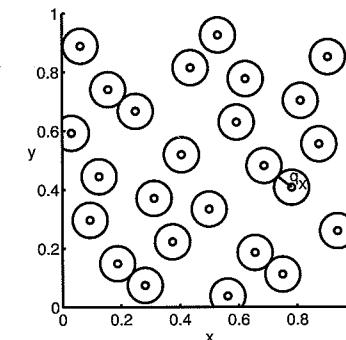


Fig. 16.1 The separation distance for $N = 25$ Halton points ($q_{\mathcal{X}} \approx 0.0597$).

where DM_data is the matrix of pairwise distances among the data sites \mathcal{X} . The identity matrix is added only to avoid counting the distance of a point \mathbf{x}_j to itself as a potential minimum.

The derivation of these bounds is rather technical, and for details we refer to either the original papers by Narcowich, Ward and co-workers listed above, the more recent paper [Schaback (2002)], or the book [Wendland (2005a)]. By providing matching lower bounds for $\|A^{-1}\|_2$ (*i.e.*, upper bounds for λ_{\min}) Schaback showed that the upper bounds on the norm of the inverse obtained by Narcowich, Ward and others are near optimal (see [Schaback (1994b)]).

We now list several (lower) bounds for λ_{\min} as derived in [Wendland (2005a)]. In the examples below the explicit (space-dependent) constants

$$M_s = 12 \left(\frac{\pi \Gamma^2(\frac{s+2}{2})}{9} \right)^{1/(s+1)} \leq 6.38s \quad \text{and} \quad C_s = \frac{1}{2\Gamma(\frac{s+2}{2})} \left(\frac{M_s}{\sqrt{8}} \right)^s$$

are used. The upper bound for M_s can be obtained using Stirling's formula (see [Wendland (2005a)]).

Since the bounds in the literature for Gaussians and multiquadratics also include the influence of the shape parameter ε we present the basic functions in their scaled version here.

Example 16.1. For Gaussians $\Phi(\mathbf{x}) = e^{-\varepsilon^2 \|\mathbf{x}\|^2}$ one obtains

$$\lambda_{\min} \geq C_s (\sqrt{2\varepsilon})^{-s} e^{-40.71s^2/(q_{\mathcal{X}}\varepsilon)^2} q_{\mathcal{X}}^{-s}.$$

We see that, for a fixed shape parameter ε , the lower bound for λ_{\min} goes exponentially to zero as the separation distance $q_{\mathcal{X}}$ decreases. Since we observed above that the condition number of the interpolation matrix A depends on the ratio of its largest and smallest eigenvalues and the growth of λ_{\max} is of order N we see that the condition number grows exponentially with decreasing separation distance. This shows that, if one adds more interpolation points in order to improve the accuracy of the interpolant (within the same domain Ω), then the problem becomes

increasingly ill-conditioned. Of course one would always expect this to happen, but here the ill-conditioning grows primarily due to the decrease in the separation distance $q_{\mathcal{X}}$, and not to the increase in the number N of data points. We will come back to this observation when we discuss a possible change of basis in Section 34.4.

On the other hand, if one keeps the number of points (or at least the separation distance) fixed and instead decreases the value of ε , then the condition number of A suffers in almost the same exponential manner. Of course, an increase in ε can be used to improve the condition number of A (however, as we saw earlier, at the expense of accuracy of the fit).

Example 16.2. For scaled generalized (inverse) multiquadratics $\Phi(\mathbf{x}) = (1 + \|\varepsilon\mathbf{x}\|^2)^{\beta}$, $\beta \in \mathbb{R} \setminus \mathbb{N}_0$ one obtains

$$\lambda_{\min} \geq C(s, \beta, \varepsilon) q_{\mathcal{X}}^{\beta - \frac{s}{2} + \frac{1}{2}} e^{-2M_s/(q_{\mathcal{X}}\varepsilon)}$$

with another explicitly known constant $C(s, \beta, \varepsilon)$.

The same comments as in the previous example apply.

Example 16.3. For thin plate splines $\Phi(\mathbf{x}) = (-1)^{\beta+1} \|\mathbf{x}\|^{2\beta} \log \|\mathbf{x}\|$, $\beta \in \mathbb{N}$, one obtains

$$\lambda_{\min} \geq C_s c_{\beta} (2M_s)^{-s-2\beta} q_{\mathcal{X}}^{2\beta}$$

with another explicitly known constant c_{β} .

In this case the lower bound also goes to zero with decreasing separation distance. However the decay is only of polynomial order.

Example 16.4. For the radial powers $\Phi(\mathbf{x}) = (-1)^{\lceil \beta/2 \rceil} \|\mathbf{x}\|^{\beta}$, $0 < \beta \notin 2\mathbb{N}$, one obtains

$$\lambda_{\min} \geq C_s c_{\beta} (2M_s)^{-s-\beta} q_{\mathcal{X}}^{\beta}$$

with another explicitly known constant c_{β} (different from c_{β} in Example 3). Again, the decay is of polynomial order.

Example 16.5. For the compactly supported functions of Wendland $\Phi_{s,k}(\mathbf{x}) = \varphi_{s,k}(\|\mathbf{x}\|)$ one obtains

$$\lambda_{\min} \geq C(s, k) q_{\mathcal{X}}^{2k+1}$$

with a constant $C(s, k)$ depending on s and k . The lower bound goes to zero with the separation distance at a polynomial rate.

16.2 Trade-Off Principle I: Accuracy vs. Stability

The observations made in Examples 16.1 and 16.2 above set up the first *trade-off principle*. This principle states that if we use the standard approach to the RBF interpolation problem (*i.e.*, solution of the linear system (6.3)) then there is

a conflict between theoretically achievable accuracy and numerical stability. For example, the error bounds for non-stationary interpolation using infinitely smooth basis functions show that the error decreases (exponentially) as the fill distance decreases. For well-distributed data a decrease in the fill distance also implies a decrease of the separation distance. But now the condition estimates of the previous subsection imply that the condition number of A grows exponentially. This leads to numerical instabilities which make it virtually impossible to obtain the highly accurate results promised by the theoretical error bounds.

Similarly, if we use the shape parameter to (exponentially) increase accuracy as guaranteed by Madych's error bound (15.11), then the condition number again grows exponentially. This is to be expected since for small values of ε the basic functions more and more resemble a constant function, and therefore the rows (as well as columns) of the matrix A become more and more alike, so that the matrix becomes almost singular — even for well separated data sites.

In the literature this phenomenon has been referred to as *trade-off* or (*uncertainty*) principle (see, *e.g.*, the papers [Schaback (1995b); Schaback (1995c)]).

Schaback looked at the power function $P_{\Phi, \mathcal{X}}$ and showed that it can always be bounded from above by a function F_{Φ} depending on the fill distance. On the other hand, he showed that the Rayleigh quotient can always be bounded from below by a function G_{Φ} depending on the separation distance. Furthermore, Schaback showed that

$$G_{\Phi}(q_{\mathcal{X}}) \leq F_{\Phi}(h_{\mathcal{X}, \Omega}),$$

and therefore, for well-distributed data (with $q_{\mathcal{X}} \approx h_{\mathcal{X}, \Omega}$), a small error bound (*i.e.*, small $F_{\Phi}(h_{\mathcal{X}, \Omega})$) will necessarily result in a small lower bound (*i.e.*, small $G_{\Phi}(q_{\mathcal{X}})$) for the Rayleigh quotient, and therefore for the smallest eigenvalue. This however implies a large condition number.

We have seen evidence of the first trade-off principle in various numerical experiments. This trade-off has led a number of people to search for an “optimal” value of the shape parameter, *i.e.*, a value that yields maximal accuracy, while still maintaining numerical stability.

In particular, multiquadratics have attracted the best part of this attention. For example, in his original work on (inverse) multiquadric interpolation in \mathbb{R}^2 Hardy [Hardy (1971)] suggested using $\varepsilon = 1/(0.815d)$, where $d = \frac{1}{N} \sum_{i=1}^N d_i$, and d_i is the distance from \mathbf{x}_i to its nearest neighbor. Later, in [Franke (1982a)], one can find the recommended value $\varepsilon = \frac{0.8\sqrt{N}}{D}$, where D is the diameter of the smallest circle containing all data points. Another strategy for finding a good value for ε is based on the observation that such a value seems to be similar for multiquadratics and inverse multiquadratics (see [Foley (1994)]). Other studies were reported in [Carlson and Foley (1992); Carlson and Natarajan (1994)]. We will consider a more recent algorithm proposed in [Rippa (1999)] in the next chapter.

16.3 Trade-Off Principle II: Accuracy and Stability vs. Problem Size

More recently, Fornberg and co-workers have investigated the dependence of the stability on the values of the shape parameter ε in a series of papers (*e.g.*, [Driscoll and Fornberg (2002); Fornberg and Wright (2004); Larsson and Fornberg (2005); Platte and Driscoll (2005)]). They suggest a way to stably compute very accurate generalized (inverse) multiquadric and Gaussian interpolants with extreme values of $\varepsilon \rightarrow 0$ by using a complex Contour-Padé integration algorithm. Thus, this approach allows us to overcome the first trade-off principle mentioned in the previous section. However, there is another kind of trade-off associated with the Contour-Padé approach. Namely it is limited to only rather small data sets (roughly $N = 20$ for $s = 1$ and $N = 80$ for $s = 2$).

In spite of these limitations the Contour-Padé algorithm has been used to gain a number of theoretical insights such as the connection between RBF interpolation and polynomial interpolation mentioned in Section 15.6. We present some numerical experiments based on the Contour-Padé approach in the next chapter.

16.4 Trade-Off Principle III: Accuracy vs. Efficiency

There is also a trade-off principle for compactly supported functions. This was explained theoretically as well as illustrated with numerical experiments in [Schaback (1997b)]. The consequences are as follows. In the case of stationary interpolation, *i.e.*, if we scale the support size of the basis functions proportional to the fill distance $h_{\mathcal{X},\Omega}$, the “bandwidth” of the interpolation matrix A is kept constant. This means we can apply numerical algorithms (*e.g.*, the conjugate gradient method) for the solution of the interpolation system that can be performed with $\mathcal{O}(N)$ computational complexity. The method is numerically stable, but there will be essentially no convergence (see our earlier numerical experiments in Table 12.1). In the non-stationary case, *i.e.*, with fixed support size, the bandwidth of A increases as $h_{\mathcal{X},\Omega}$ decreases. This results in convergence (*i.e.*, the error decreases) as we showed with our experiments in Table 12.2 and the error bounds in Section 15.1.2. However, the interpolation matrices will become more and more densely populated as well as ill-conditioned. Therefore, this approach is not very efficient.

Chapter 17

Numerical Evidence for Approximation Order Results

17.1 Interpolation for $\varepsilon \rightarrow 0$

We begin by considering the choice of the shape parameter for a fixed data set. This is probably the situation that will arise most frequently in practical situations. In other words, we assume we are given a set of data (\mathbf{x}_j, f_j) , $j = 1, \dots, N$, with data sites $\mathbf{x}_j \in \mathbb{R}^s$ (with $s = 1$ or $s = 2$ for the purpose of our experiments), and function values $f_j = f(\mathbf{x}_j) \in \mathbb{R}$. Our goal is to use an RBF interpolant

$$\mathcal{P}_f(\mathbf{x}) = \sum_{j=1}^N c_j \varphi(\|\mathbf{x} - \mathbf{x}_j\|)$$

to match these data exactly, *i.e.*, to satisfy $\mathcal{P}_f(\mathbf{x}_i) = f(\mathbf{x}_i)$, $i = 1, \dots, N$. The two most important questions now seem to be:

- Which basic function φ should we use?
- How should we scale the basis functions $\varphi_j = \varphi(\|\cdot - \mathbf{x}_j\|)$?

The error bounds we reviewed in previous chapters give us some insight into the first issue. If we know that the data come from a very smooth function, then application of one of the smoother basic functions is called for. Otherwise, there is not much to be gained from doing so. In fact, these functions may add too much smoothness to the interpolant. A first attempt at providing guidelines for the selection of appropriate basic functions (or kernels) can be found in [Schaback and Wendland (2006)]. We will not pursue this issue any further.

Instead we want to focus our attention on the second question, *i.e.*, the choice of the shape parameter ε . A number of strategies can be used to guide us in making a decision. We will assume throughout that a (fixed) basic function has been chosen, and that we will use only one value to scale all basis functions uniformly. Clearly, one can also follow other strategies such as using a shape parameter that varies with j , or even basic functions that vary with j . While some work has been done in these directions (see, *e.g.*, [Bozzini *et al.* (2002); Kansa and Carlson (1992); Schaback and Wendland (2000b); Fornberg and Zuev (2006)]), not much concrete can be said in these cases.

We now discuss four strategies for choosing a “good” value of ε .

17.1.1 Choosing a Good Shape Parameter via Trial and Error

The simplest strategy is to perform a series of interpolation experiments with varying shape parameter, and then to pick the “best” one. This strategy can be used if we know the function f that generated the data, and therefore can calculate some sort of error for the interpolant. Of course, if we already know f , then the exercise of finding an interpolant \mathcal{P}_f may be mostly pointless. However, this is the strategy we used for the “academic” examples in Chapter 2.

If we do not have any knowledge of f , then it becomes very difficult to decide what “best” means. One (non-optimal) criterion we used in Chapter 2 was based on the trade-off principle, *i.e.*, the fact that for small ε the error improves while the condition number grows. We then defined “best” to be the smallest ε for which MATLAB did not issue a near-singular warning.

In many cases selection of an optimal shape parameter via *trial and error* will end up being a rather subjective process. However, this may presently be the approach taken by most practitioners.

For comparison with the other selection methods featured below we present three one-dimensional test cases for which we know the data function f . We use

$$\begin{aligned} F_1(x) &= \text{sinc}(x), \\ F_2(x) &= \frac{3}{4} \left(e^{-(9x-2)^2/4} + e^{-(9x+1)^2/49} \right) + \frac{1}{2} e^{-(9x-7)^2/4} - \frac{1}{10} e^{-(9x-4)^2}, \\ F_3(x) &= \left(1 - \left| x - \frac{1}{2} \right| \right)^5 \left(1 + 5\left| x - \frac{1}{2} \right| - 27\left| x - \frac{1}{2} \right|^2 \right). \end{aligned}$$

The first of these functions is the classical band limited function (and thus in the native space of Gaussians). The second function is a one-dimensional variant of Franke’s function, and the third function is one of Gneiting’s C^2 oscillatory compactly supported RBFs shifted to the point $(1/2, 1/2)$ (see Table 11.4).

For these functions we list maximum errors and optimal shape parameters ε in Table 17.1. Maximum errors for a large range of ε values and the different values of N used in Table 17.1 are displayed in Figure 17.1. The optimal ε values listed in Table 17.1 corresponds to the lowest point for each of the curves in the figure. Clearly, the optimal value of the shape parameter is strongly dependent on the function that generated the test data.

17.1.2 The Power Function as Indicator for a Good Shape Parameter

Another strategy is suggested by the error analysis of Chapter 14. We showed there in Theorem 14.2 that

$$|f(\mathbf{x}) - \mathcal{P}_f(\mathbf{x})| \leq P_{\Phi, \mathcal{X}}(\mathbf{x}) \|f\|_{\mathcal{N}_{\Phi}(\Omega)},$$

Table 17.1 Optimal ε values based on Gaussian interpolation to various test functions in 1D for various choices of N uniform points.

N	F_1		F_2		F_3	
	max-error	optimal ε	max-error	optimal ε	max-error	optimal ε
3	2.1403e-003	1.12	4.9722e-001	2.20	3.7087e-002	5.68
5	2.3260e-005	0.68	6.9380e-002	5.44	2.5253e-002	5.20
9	4.8764e-009	0.64	1.7555e-002	5.20	2.5360e-003	8.84
17	1.8922e-010	1.00	2.1928e-004	5.80	1.4380e-003	9.52
33	1.5250e-010	2.04	1.6536e-007	6.08	3.4189e-004	13.24
65	4.1307e-010	2.04	3.6260e-009	7.48	8.6431e-005	21.84

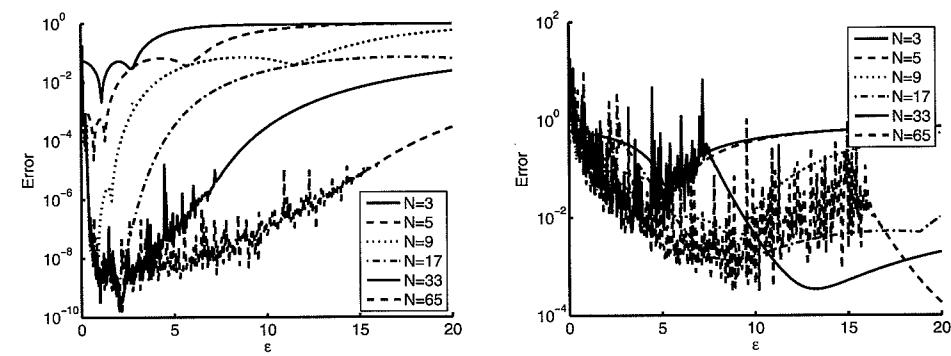


Fig. 17.1 Optimal ε curves based on Gaussian interpolation in 1D for various choices of N uniform points. Data sampled from sinc function F_1 (left) and C^2 oscillatory function F_3 (right).

where $P_{\Phi, \mathcal{X}}$ denotes the power function. This estimate decouples the interpolation error into a component independent of the data function f and one depending on f . Once we have decided on a basic function Φ and a data set \mathcal{X} we can use the power function based on scaled versions of Φ to optimize the error component that is independent of f . While this approach has the advantage over the previously mentioned trial and error approach that it is objective and does not depend on any knowledge of the data function, unfortunately, this approach will not be an optimal one since the second component of the error bound also depends on the basic function via the native space norm (which changes when Φ is scaled).

We said earlier (see (14.4)) that the power function can be computed via

$$P_{\Phi, \mathcal{X}}(\mathbf{x}) = \sqrt{\Phi(\mathbf{x}, \mathbf{x}) - (\mathbf{b}(\mathbf{x}))^T A^{-1} \mathbf{b}(\mathbf{x})},$$

where A is the interpolation matrix and $\mathbf{b} = [\Phi(\cdot, \mathbf{x}_1), \dots, \Phi(\cdot, \mathbf{x}_N)]^T$. This formula is implemented on lines 15–18 in the MATLAB program `Powerfunction2D.m`. We compute the inverse of A using the function `pinv` which is based on the singular value decomposition of A and therefore guarantees maximum stability. Also, due to roundoff some of the arguments of the `sqrt` function on line 18 come out negative.

This explains the use of the `real` command. The vectors $b(x)$ are just rows of the evaluation matrix if x is taken from the grid of evaluation points we used earlier for error computations and plotting purposes. Except for the loop over the shape parameter ϵ (lines 12–20) the rest of the program is similar to earlier code.

Program 17.1. Powerfunction2D.m

```
% Powerfunction2D
% Script that finds "optimal" shape parameter by computing the power
% function for the 2D RBF interpolation approach with varying epsilon
% Calls on: DistanceMatrix
1 rbf = @(e,r) exp(-(e*r).^2); % Define the Gaussian RBF
% Parameters for shape parameter loop below
2 mine = 0; maxe = 20;
3 ne = 500; ep = linspace(mine,maxe,ne);
% Number and type of data points
4 N = 81; gridtype = 'u';
% Resolution of grid for power function norm computation
5 neval = 20; M = neval^2;
% Load data points
6 name = sprintf('Data2D_%d%s',N,gridtype); load(name)
7 ctrs = dsites; % centers coincide with data sites
8 grid = linspace(0,1,neval); [xe,ye] = meshgrid(grid);
9 epoints = [xe(:) ye(:)];
% Compute distance matrix between evaluation points and centers
10 DM_eval = DistanceMatrix(epoints,ctrs);
% Compute distance matrix between the data sites and centers
11 DM_data = DistanceMatrix(dsites,ctrs);
12 for i=1:length(ep)
    % Compute interpolation matrix
13 IM = rbf(ep(i),DM_data);
    % Compute evaluation matrix
14 EM = rbf(ep(i),DM_eval);
    % Compute power function at evaluation points
15 invIM = inv(IM); phi0 = rbf(ep(i),0);
16 for j=1:M
        powfun(j) = real(sqrt(phi0-(invIM*EM(j,:))*EM(j,:)));
17 end
    % Compute max. norm of power function on evaluation grid
18 maxPF(i) = max(powfun);
19 end
21 fprintf('Smallest maximum norm: %e\n', min(maxPF))
22 fprintf('at epsilon = %f\n',ep(maxPF==min(maxPF)))
```

```
23 fprintf('with cond(A) = %e\n', ...
    condest(rbf(ep(find(maxPF==min(maxPF))),DM_data)))
% Plot power function norm
24 figure; semilogy(ep,maxPF,'b');
```

In Figure 17.2 we show plots of the maximum norms of the power function vs. ϵ for a one-dimensional experiment (left) and a 2D experiment (right). Each plot shows several curves corresponding to different choices of N (set on line 4 of `Powerfunction2D.m`). The optimal ϵ values along with the corresponding condition numbers of the interpolation matrix (computed using the `condest` command) are listed in Table 17.2. The graphs of the maximum norm of the power function can all be included in a single plot by adding another loop to the program which varies N . The program for the one-dimensional case is almost identical to the one printed and therefore omitted.

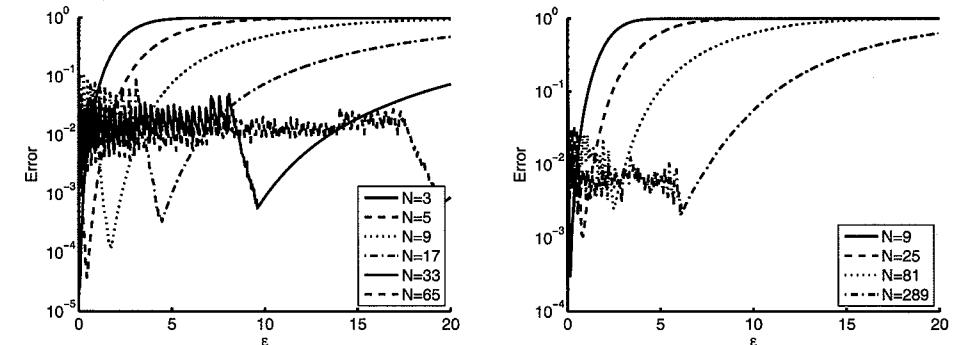


Fig. 17.2 Optimal ϵ curves based on power functions for Gaussians in 1D (left) and 2D (right) for various choices of N uniform points.

Clearly, even for the small data sets considered here, the numerical instability, *i.e.*, large condition number of the interpolation matrix A , plays a significant role.

Table 17.2 Optimal ϵ values based on power functions for Gaussians in 1D and 2D for various choices of N uniform points.

N	1D		2D		
	optimal ϵ	cond(A)	N	optimal ϵ	cond(A)
3	0.04	1.8749e+007	9	0.16	5.3534e+009
5	0.44	5.7658e+007	25	0.84	1.0211e+011
9	1.72	6.5682e+008	81	0.04	2.0734e+019
17	4.48	6.1306e+009	289	0.56	1.2194e+020
33	9.60	5.4579e+010			
65	19.52	1.2440e+011			

17.1.3 Choosing a Good Shape Parameter via Cross Validation

A third strategy for finding an “optimal” shape parameter is to use a *cross validation* approach. In [Rippa (1999)] an algorithm is described that corresponds to a variant of cross validation known as “*leave-one-out*” *cross validation*. This method is rather popular in the statistics literature where it is also known as PRESS (Predictive REsidual Sum of Squares) provided the 2-norm is used. In this algorithm an “optimal” value of ε is selected by minimizing the (least squares) error for a fit to the data based on an interpolant for which one of the centers was “left out”. A major advantage over the previous method is that now the dependence of the error on the data function is also taken into account. Therefore, the predicted “optimal” shape parameter is closer to the one we found via the trial and error approach (for which we had to assume knowledge of the exact solution).

A similar strategy was proposed earlier in [Golberg *et al.* (1996)] for the solution of elliptic partial differential equations via the dual reciprocity method based on multiquadric interpolation.

Specifically, if $\mathcal{P}_f^{[k]}$ is the radial basis function interpolant to the data $\{f_1, \dots, f_{k-1}, f_{k+1}, \dots, f_N\}$, i.e.,

$$\mathcal{P}_f^{[k]}(\mathbf{x}) = \sum_{\substack{j=1 \\ j \neq k}}^N c_j^{[k]} \varphi(\|\mathbf{x} - \mathbf{x}_j\|),$$

such that

$$\mathcal{P}_f^{[k]}(\mathbf{x}_i) = f_i, \quad i = 1, \dots, k-1, k+1, \dots, N,$$

and if E_k is the error

$$E_k = f_k - \mathcal{P}_f^{[k]}(\mathbf{x}_k)$$

at the one point \mathbf{x}_k not used to determine the interpolant, then the quality of the fit is determined by the norm of the vector of errors $E = [E_1, \dots, E_N]^T$ obtained by removing in turn one of the data points and comparing the resulting fit with the (known) value at the removed point. In [Rippa (1999)] the author presented examples based on use of the ℓ_1 and ℓ_2 norms. We will mostly use the maximum norm (see line 14 in the code below).

By adding a loop over ε we can compare the error norms for different values of the shape parameter, and choose that value of ε that yields the minimal error norm as the optimal one.

While a naive implementation of the leave-one-out algorithm is rather expensive (on the order of N^4 operations), Rippa showed that the computation of the error components can be simplified to a single formula

$$E_k = \frac{c_k}{A_{kk}^{-1}}, \quad (17.1)$$

where c_k is the k th coefficient in the interpolant \mathcal{P}_f based on the *full* data set, and A_{kk}^{-1} is the k th diagonal element of the inverse of the corresponding interpolation

matrix. Since both c_k and A^{-1} need to be computed only once for each value of ε this results in $\mathcal{O}(N^3)$ computational complexity. Note that all entries in the error vector E can be computed in a single statement in MATLAB if we vectorize the component formula (17.1) (see line 13 in Program 17.2). The sinc function used on line 5 is not a standard MATLAB function (it is part of the Signal Processing Toolbox). Therefore we provide it in Program C.2 in Appendix C.

Program 17.2. LOOCV2D.m

```
% LOOCV2D
% Script that performs leave-one-out cross-validation
% (Rippa's method) to find a good epsilon for 2D RBF interpolation
% Calls on: DistanceMatrix
1 rbf = @(e,r) exp(-(e*r).^2); % Gaussian RBF
% Parameters for shape parameter loop below
2 mine = 0; maxe = 20; ne = 500;
3 ep = linspace(mine,maxe,ne);
% Number and type of data points
4 N = 81; gridtype = 'u';
% Define test function
5 testfunction = @(x,y) sinc(x).*sinc(y);
% Load data points
6 name = sprintf('Data2D_%d%s',N,gridtype); load(name)
7 ctrs = dsites; % centers coincide with data sites
% Create right-hand side vector, i.e.,
% evaluate the test function at the data points.
8 rhs = testfunction(dsites(:,1),dsites(:,2));
% Compute distance matrix between the data sites and centers
9 DM_data = DistanceMatrix(dsites,ctrs);
10 for i=1:length(ep)
    % Compute interpolation matrix
11 IM = rbf(ep(i),DM_data);
    % Compute error function (i.e., "cost" of epsilon)
12 invIM = pinv(IM);
13 EF = (invIM*rhs)./diag(invIM);
    % Compute maximum norm of EF
14 maxEF(i) = norm(EF(:),inf);
15 end
16 fprintf('Smallest maximum norm: %e\n', min(maxEF))
17 fprintf('at epsilon = %f\n', ep(maxEF==min(maxEF)))
    % Plot cost function norm
18 figure; semilogy(ep,maxEF,'b');
```

In Figure 17.3 we show plots of the predicted maximum errors vs. ε for a one-

dimensional experiment (left) and a 2D experiment (right) based on data sampled from the sinc function F_1 . Each plot shows several curves corresponding to different choices of N (set on line 4 of `L00CV2D.m`). The optimal ε values are listed in Table 17.3.

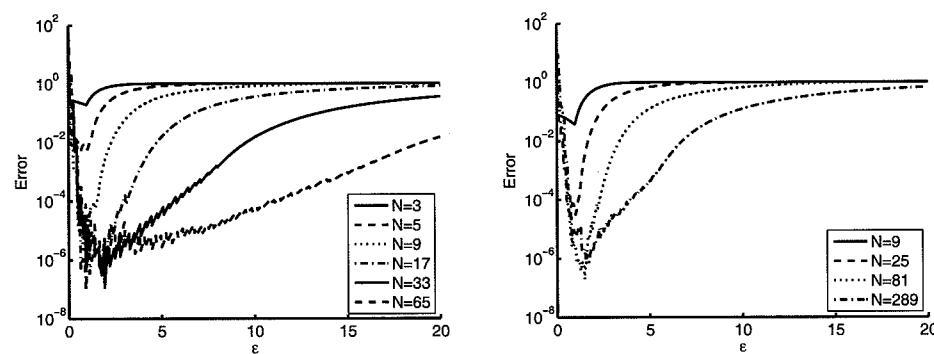


Fig. 17.3 Optimal ε curves based on leave-one-out cross validation for interpolation to the sinc function with Gaussians in 1D (left) and 2D (right) for various choices of N uniform points.

Table 17.3 Optimal ε values based on leave-one-out cross validation for interpolation to the sinc function with Gaussians in 1D and 2D for various choices of N uniform points.

1D		2D	
N	optimal ε	N	optimal ε
3	0.96	9	0.96
5	1.00	25	1.00
9	0.80	81	1.48
17	0.92	289	1.60
33	1.92		
65	1.76		

The graphs in Figure 17.4 show side-by-side the optimal ε curves for the trial and error approach and for the leave-one-out cross validation approach in the case of 1D Gaussian interpolation to data sampled from the test function F_2 . The similarity of the curves is clearly apparent. Thus, the leave-one-out cross validation approach can be recommended as a good method for selecting an “optimal” shape parameter ε since for this method no knowledge of the exact error is needed. Another pair of comparison plots is given by the Gaussian interpolants to the sinc function F_1 in Figure 17.1 (left) and Figure 17.3 (left).

Similar conclusions hold for other basic functions, other test functions, other data distributions, and other space dimensions. For example, Figure 17.5 shows the

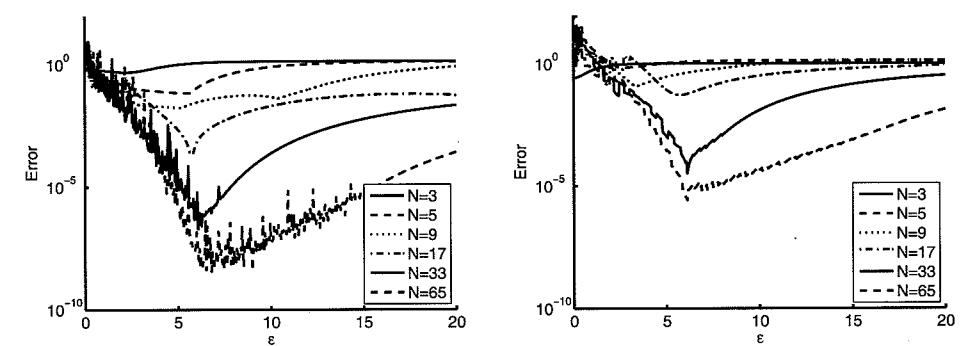


Fig. 17.4 Optimal ε curves based on interpolation to the test function F_2 with Gaussians for various choices of N uniform points. Trial and error approach (left), leave-one-out cross validation (right).

optimal ε curves for interpolation to the 1D Franke function F_2 with Wendland’s C^2 function $\varphi_{3,1}$ on uniformly spaced points, and on Chebyshev points. Note that for this configuration all computations are stable, and the optimal scale parameter is quite small, *i.e.*, the support radius of the compactly supported basic function is chosen to be very large. In other words, the best results for compactly supported functions are obtained with dense matrices.

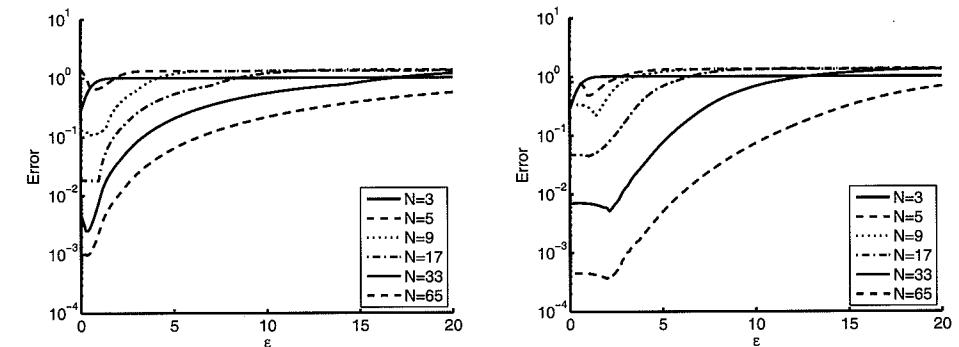


Fig. 17.5 Optimal ε curves based on leave-one-out cross validation for interpolation to 1D Franke’s function with Wendland’s function $\varphi_{3,1}$ for various choices of N uniform points (left) and Chebyshev points (right).

If we are not interested in the ε -curves displayed above, but only want to find a good value of the shape parameter as quickly as possible, then we can use the MATLAB function `fminbnd` to find the minimum of the cost function for ε . First, we implement the cost function in the subroutine `CostEpsilon.m` displayed in Program 17.3. The commands are the same as those on lines 11–14 in Program 17.2.

Program 17.3. CostEpsilon.m

```
% ceps = CostEpsilon(ep,r,rbf,rhs)
% Implements cost function for optimization of shape parameter
% via Rippa's LOOCV algorithm
% Example of usage in LOOCV2Dmin.m
1 function ceps = CostEpsilon(ep,r,rbf,rhs)
2 A = rbf(ep,r);
3 invA = pinv(A);
4 EF = (invA*rhs)./diag(invA);
5 ceps = norm(EF(:,inf);
```

In order to demonstrate the use of the `CostEpsilon` function we use a modification of Program 17.2 which we list as Program 17.4.

Program 17.4. LOOCV2Dmin.m

```
% LOOCV2Dmin
% Script that performs leave-one-out cross-validation
% (Rippa's method) to find a good epsilon for 2D RBF interpolation
% with the help of MATLAB's fminbnd
% Calls on: DistanceMatrix
% Requires: CostEpsilon
1 rbf = @(e,r) exp(-(e*r).^2); % Gaussian RBF
    % Parameters for shape parameter optimization below
2 mine = 0; maxe = 20;
    % Number and type of data points
3 N = 81; gridtype = 'u';
    % Define test function
4 testfunction = @(x,y) sinc(x).*sinc(y);
    % Load data points
5 name = sprintf('Data2D_%d%s',N,gridtype); load(name)
6 ctrs = dsites; % centers coincide with data sites
    % Create right-hand side vector, i.e.,
    % evaluate the test function at the data points.
7 rhs = testfunction(dsites(:,1),dsites(:,2));
    % Compute distance matrix between the data sites and centers
8 DM_data = DistanceMatrix(dsites,ctrs);
9a [ep,fval] = fminbnd(@(ep) CostEpsilon(ep,DM_data,rbf,rhs),...
9b                 mine,maxe);
10 fprintf('Smallest maximum norm: %e\n', fval)
11 fprintf('at epsilon = %f\n', ep)
```

17.1.4 The Contour-Padé Algorithm

The Contour-Padé algorithm was the subject of Grady Wright's Ph.D. thesis [Wright (2003)] and was reported in [Fornberg and Wright (2004)]. The aim of the Contour-Padé algorithm is to come up with a method that allows the computation and evaluation of RBF interpolants for infinitely smooth basic functions when the shape parameter ε tends to zero (including the limiting case).

The starting point is to consider evaluation of the RBF interpolant

$$\mathcal{P}_f(\mathbf{x}, \varepsilon) = \sum_{j=1}^N c_j \varphi_\varepsilon(\|\mathbf{x} - \mathbf{x}_j\|)$$

for a fixed evaluation point \mathbf{x} as an analytic function of ε .

The key idea is to represent $\mathcal{P}_f(\mathbf{x}, \varepsilon)$ by a *Laurent series* in ε , and approximate the "negative part" of the series by a Padé approximant, *i.e.*,

$$\mathcal{P}_f(\mathbf{x}, \varepsilon) \approx r(\varepsilon) + \sum_{k=0}^{\infty} d_k \varepsilon^k,$$

where $r(\varepsilon)$ is the rational Padé approximant.

We then rewrite the interpolant in cardinal form, *i.e.*, as

$$\begin{aligned} \mathcal{P}_f(\mathbf{x}, \varepsilon) &= \sum_{j=1}^N c_j \varphi_\varepsilon(\|\mathbf{x} - \mathbf{x}_j\|) \\ &= \mathbf{b}^T(\mathbf{x}, \varepsilon) \mathbf{c} \\ &= \mathbf{b}^T(\mathbf{x}, \varepsilon) \mathbf{A}^{-1}(\varepsilon) \mathbf{f} \\ &= (\mathbf{u}^*(\mathbf{x}, \varepsilon))^T \mathbf{f} \end{aligned}$$

where $\mathbf{b}(\mathbf{x}, \varepsilon)_j = \varphi_\varepsilon(\|\mathbf{x} - \mathbf{x}_j\|)$, $\mathbf{A}(\varepsilon)_{i,j} = \varphi_\varepsilon(\|\mathbf{x}_i - \mathbf{x}_j\|)$, $\mathbf{c} = [c_1, \dots, c_N]^T$, $\mathbf{f} = [f_1, \dots, f_N]^T$, and

$$\mathbf{u}^*(\mathbf{x}, \varepsilon) = \mathbf{A}^{-1}(\varepsilon) \mathbf{b}(\mathbf{x}, \varepsilon)$$

denotes the vector of values of the cardinal functions at \mathbf{x} (*c.f.* Chapter 14).

It is now the goal to stably compute the vector $\mathbf{u}^*(\varepsilon)$ for all values of $\varepsilon \geq 0$ without explicitly forming the inverse $\mathbf{A}(\varepsilon)^{-1}$ and without computing the matrix vector product $\mathbf{A}(\varepsilon)^{-1} \mathbf{b}(\varepsilon)$. Here the vectors $\mathbf{u}^*(\varepsilon)$ and $\mathbf{b}(\varepsilon)$ are obtained by evaluating the vector functions $\mathbf{u}^*(\cdot, \varepsilon)$ and $\mathbf{b}(\cdot, \varepsilon)$ on an appropriate evaluation grid.

The solution proposed by Wright and Fornberg is to use Cauchy's integral theorem to integrate around a circle in the complex ε -plane. The residuals (*i.e.*, coefficients in the Laurent expansion) are obtained using the (inverse) fast Fourier transform. The terms with negative powers of ε are then approximated using a rational Padé approximant. The integration contour (usually a circle) has to lie between the region of instability near $\varepsilon = 0$ and possible branch point singularities that lie somewhere in the complex plane depending on the choice of φ . Details of the method can be found in [Fornberg and Wright (2004)].

In Figure 17.6 we show optimal ε curves for interpolation to the 1D and 2D sinc function F_2 using Gaussians at equally spaced points. These curves should be compared with the optimal ε curves obtained for the same problem via trial and error (see Figure 17.1 and Table 17.1) and via leave-one-out cross validation (see Figure 17.3 and Table 17.3).

The main drawback of the Contour-Padé algorithm is the fact that if N becomes too large then the region of ill-conditioning around the origin in the complex ε -plane and the branch point singularities will overlap. This, however, implies that the method can only be used with limited success. Moreover, as the graphs in Figure 17.6 and the entries in Table 17.4 show, the value of N that has to be considered “large” is unfortunately rather small. For the one-dimensional case the results for $N = 17$ already are affected by instabilities, and in the two-dimensional experiment $N = 81$ causes problems.

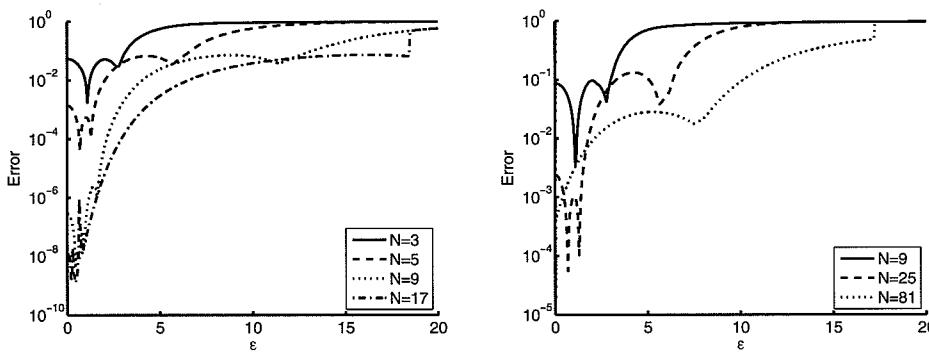


Fig. 17.6 Optimal ε curves based on Contour-Padé for interpolation to the sinc function with Gaussians in 1D (left) and 2D (right) for various choices of N uniform points.

Table 17.4 Optimal ε values based on Contour-Padé for interpolation to the sinc function with Gaussians in 1D and 2D for various choices of N uniform points.

N	1D			2D			
	max-error	ε	cond(A)	N	max-error	ε	cond(A)
3	1.7605e-003	1.10	3.3386e+001	9	3.3875e-003	1.10	1.1146e+003
5	4.0380e-005	0.70	1.3852e+006	25	5.5542e-005	0.70	1.9187e+012
9	3.9703e-009	0.45	7.7731e+016	81	3.6528e-004	0.00	∞
17	1.2726e-009	0.45	1.7327e+018				

17.1.5 Summary

All strategies pursued in this chapter have shown that even though the bound (15.11) by Madych seems to indicate that the interpolation error for functions in

the native space of the basic function goes to zero exponentially as $\varepsilon \rightarrow 0$, this does not seem to be true in practice. Especially those optimal ε curves that were computed reliably with the Contour-Padé algorithm all have a global minimum for some positive value of ε . In many cases this optimal ε value (or an ε close to the optimal value) can be found using the leave-one-out cross validation algorithm of Program 17.2. From now on we will frequently use leave-one-out cross validation to find an optimal shape parameter for our numerical experiments.

17.2 Non-stationary Interpolation

In order to illustrate the spectral convergence predicted for infinitely smooth basic functions such as Gaussians and generalized (inverse) multiquadratics we need to work in a setting for which neither the instability due to large problem size or small shape parameter have a significant effect on our experiments. Otherwise, if we simply take an “optimal” value of ε (determined via trial and error for a large $N = 4225$ problem in the “gray zone”, c.f. Chapter 2) then the spectral convergence will only be visible for a limited number of experiments (see Table 17.5).

Table 17.5 2D non-stationary interpolation ($\varepsilon = 6.3$) to Franke’s function with Gaussians on uniformly spaced and Halton points.

N	uniform		Halton	
	RMS-error	rate	RMS-error	rate
9	3.195983e-001		2.734756e-001	
25	5.008591e-002	2.6738	8.831682e-002	2.3004
81	9.029664e-003	2.4717	2.401868e-002	1.7582
289	2.263880e-004	5.3178	1.589117e-003	5.0969
1089	3.323287e-008	12.7339	1.595051e-006	10.8015
4225	1.868286e-008	0.8309	9.510404e-008	4.8203

Even for a band-limited function (see Table 17.6) the situation is not better; in fact worse, for the value of ε used.

In Figures 17.7–17.8 we are able to verify (at least to some extent) the convergence estimates for non-stationary RBF interpolants. We obtain the data for all experiments by sampling the sinc function $f(x) = \sin(\pi x)/(\pi x)$ at N uniformly spaced points in the interval $[0, 1]$ where N runs from 1 to 100. Each plot shows six maximum error curves (corresponding to shape parameters $\varepsilon = 1, 6, 11, 16, 21, 26$) versus the number N of data points on a loglog scale. The errors are evaluated on a grid of 250 equally spaced points. In order to compare these curves with the theoretical bounds from Chapter 15 we have plotted comparison curves corresponding to the theoretical bounds. For Gaussians the comparison curve is given by the graph of $h \mapsto e^{-|\log h|/h}$ corresponding to super-spectral convergence with $h = 1/(N-1)$,

Table 17.6 2D non-stationary interpolation ($\varepsilon = 6.3$) to the sinc function with Gaussians on uniformly spaced and Halton points.

<i>N</i>	uniform		Halton	
	RMS-error	rate	RMS-error	rate
9	3.302644e-001		2.823150e-001	
25	3.271035e-002	3.3358	1.282572e-001	1.6058
81	1.293184e-002	1.3388	3.407580e-002	1.7898
289	3.786113e-004	5.0941	1.990217e-003	5.3309
1089	3.476835e-008	13.4107	2.286014e-006	10.5905
4225	3.775365e-008	-0.1188	9.868530e-008	5.3724

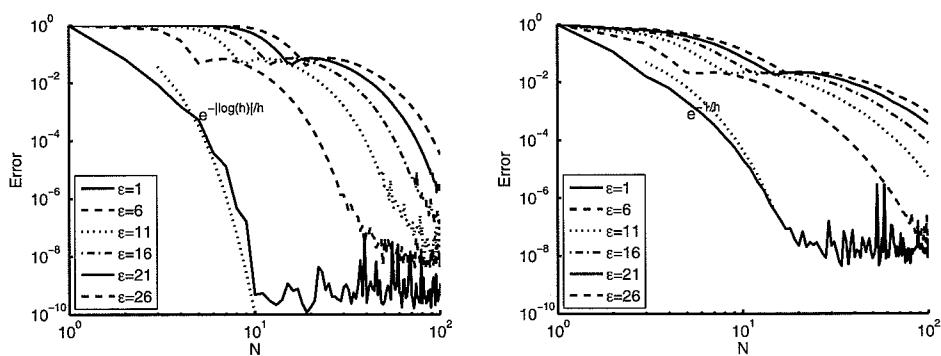


Fig. 17.7 Maximum errors for non-stationary interpolation to the sinc function with Gaussians (left) and inverse multiquadratics (right) based on N uniformly spaced points in $[0, 1]$ and $\varepsilon = 1, 6, 11, 16, 21, 26$.

and for inverse multiquadratics we have spectral convergence with $h \mapsto e^{-1/h}$. We can see that for a certain range of problems these rates are indeed obtained (see Figure 17.7).

In the case of functions with finite smoothness (such as the compactly supported functions of Wendland) we can only expect algebraic convergence rates. Figure 17.8 shows two more sets of maximum error curves. These plots are based on Wendland's C^2 function $\varphi_{3,1}(r) = (1-r)_+^4(4r+1)$ and the C^6 function $\varphi_{3,3}(r) = (1-r)_+^8(32r^3 + 25r^2 + 8r + 1)$. While the error bound (15.5) predicts only $\mathcal{O}(h^{3/2})$ and $\mathcal{O}(h^{7/2})$ approximation order, respectively. We see that an extra factor of $h^{s/2}$ is indeed possible in practice. This extra factor has also been captured in some of the theoretical work on improved error bounds (c.f. Section 15.2).

For less smooth data functions we no longer have spectral convergence for the infinitely smooth functions, while the orders remain unchanged for the basic functions with finite smoothness (as long as the data function lies in the native space of the basic function). This is illustrated in Figure 17.9 where we compare Gaussians and C^2 Wendland functions for the C^2 test function

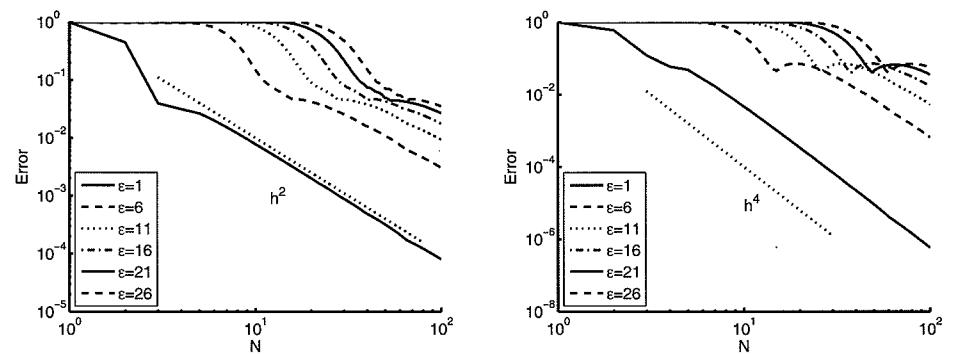


Fig. 17.8 Maximum errors for non-stationary interpolation to the sinc function with C^2 (left) and C^6 (right) Wendland function based on N uniformly spaced points in $[0, 1]$ and $\varepsilon = 1, 6, 11, 16, 21, 26$.

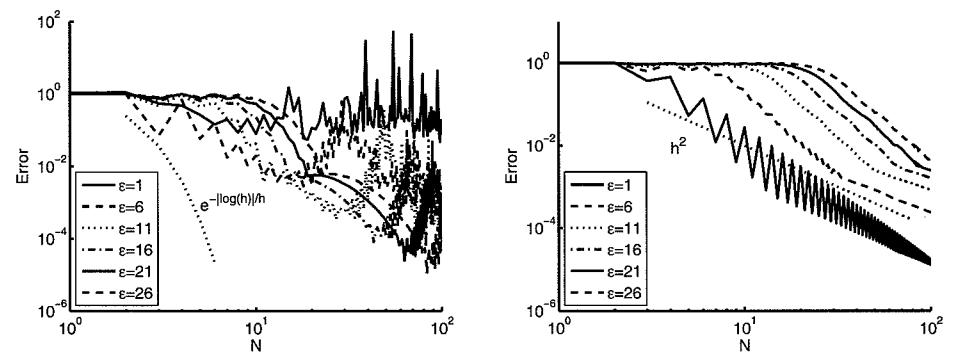


Fig. 17.9 Maximum errors for non-stationary interpolation to a C^2 function with Gaussians (left) and C^2 Wendland function (right) based on N uniformly spaced points in $[0, 1]$ and $\varepsilon = 1, 6, 11, 16, 21, 26$.

$(1 - |x - 1/2|)_+^5(1 + 5|x - 1/2| - 27(x - 1/2)^2)$ (c.f. the oscillatory functions of Table 11.4). It is interesting to note that for a certain range of N the rate of convergence for the C^2 Wendland function is even better than predicted.

17.3 Stationary Interpolation

We begin with an illustration of the fact that for radial powers and thin plate splines there is no difference in convergence behavior between the stationary and non-stationary regime. Figure 17.10 shows this phenomenon for the norm radial function $\Phi(\mathbf{x}) = \|\mathbf{x}\|$ in the case of interpolation to data sampled from the C^2 function $f(x) = |x - 1/2|^3$ at uniformly spaced points in $[0, 1]$. Moreover, the left plot in Figure 17.10 (illustrating the non-stationary setting) shows that the shape

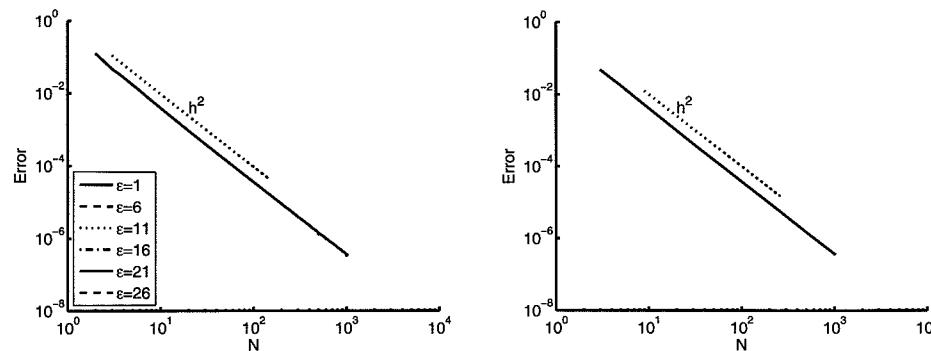


Fig. 17.10 Maximum errors for non-stationary (left) and stationary (right) interpolation to a C^2 function with the norm basic function based on N uniformly spaced points in $[0, 1]$.

parameter has no effect for the norm basic function and other polyharmonic splines.

Note that Figure 17.10 suggests that the norm basic function has $\mathcal{O}(h^2)$ approximation order, while the bound from Theorem 15.3 with $\tau = k = \beta = 1$, $n = 0$, $s = 1$ and $q = \infty$ yields only $\mathcal{O}(h^{1/2})$. Since the norm basic function is strictly conditionally positive definite of order one we can use the same RBF expansion as for strictly positive definite functions, *i.e.*, without appending a constant (*c.f.* Theorem 9.7).

The discrepancy between the theoretical bounds of Theorem 15.3 (or Theorem 15.4 as well as the native space bounds of Examples 15.6 and 15.7 of Chapter 15) and those observed in numerical experiments is similar for radial cubics and thin plate splines (which are both strictly conditionally positive definite of order two). For cubics Theorem 15.3 with $\tau = \beta = 3$, $k = 2$, $n = 0$, $s = 2$ and $q = \infty$ predicts $\mathcal{O}(h^2)$ since the mesh ratio provides another power of h for uniformly distributed data. The left plot of Figure 17.11, however, suggests $\mathcal{O}(h^3)$ or better L_∞ approximation order based on interpolation to the 2D analog of the oscillatory C^2 test function F_3 , *i.e.*, $f(\mathbf{x}) = (1 - \|\mathbf{x} - (1/2, 1/2)\|)_+^5(1 + 5\|\mathbf{x} - (1/2, 1/2)\| - 27\|\mathbf{x} - (1/2, 1/2)\|^2)$. The predicted rate for thin plate splines is $\mathcal{O}(h^{3/2})$ (since $\tau = 2\beta = 2$, $k = 2$, $n = 0$, $s = 2$ and $q = \infty$) while the plot on the right of Figure 17.11 indicates at least $\mathcal{O}(h^2)$ convergence.

For Gaussian basis functions we noted earlier that we should not expect any convergence in the stationary setting. However, if the initial shape parameter is chosen small enough (but not too small), then we can observe the approximate approximation phenomenon, *i.e.*, there is convergence up to a certain point, and then saturation occurs. This is depicted in Figure 17.12. In the left plot we used the Gaussian basic function with different initial shape parameters ($\varepsilon = 0.8, 1.0, 1.2, 1.4, 1.6, 1.8$) to interpolate data sampled from the oscillatory C^2 function used in the previous illustration at uniformly spaced points in the unit square. The plot on the right corresponds to Gaussian interpolation of data sampled from the 2D sinc function $f(x, y) = \text{sinc}(x)\text{sinc}(y)$ with initial $\varepsilon = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6$.

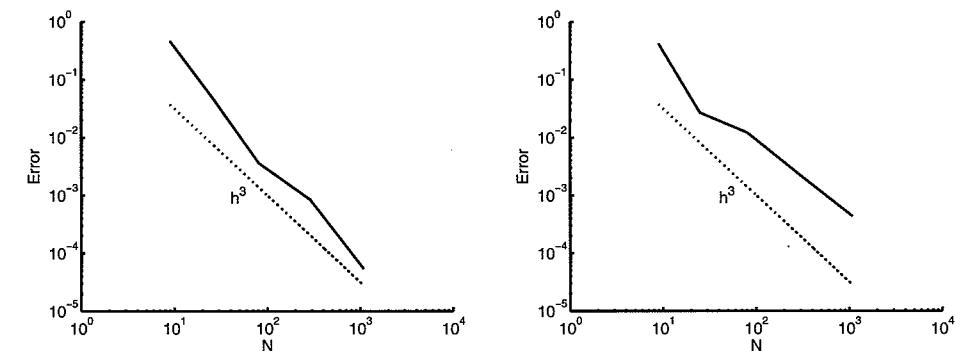


Fig. 17.11 Maximum errors for stationary interpolation to a C^2 function with the cubic radial basic function (left) and thin plate spline basic function (right) based on N uniformly spaced points in $[0, 1]^2$.

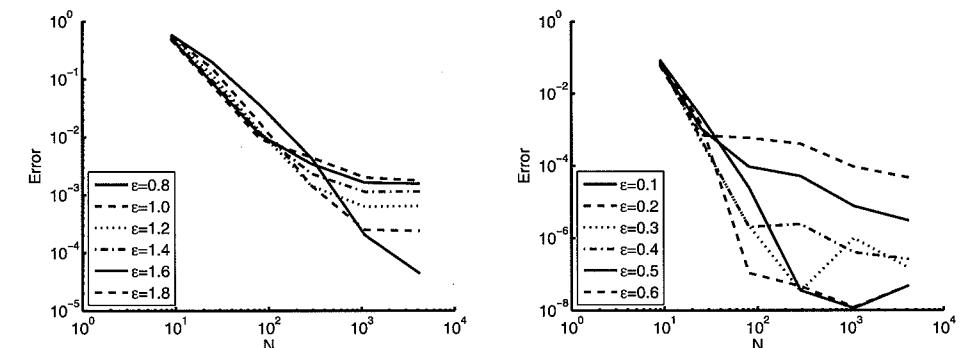


Fig. 17.12 Maximum errors for stationary interpolation to the C^2 oscillatory function (left) and to the sinc function (right) with Gaussians based on N uniformly spaced points in $[0, 1]^2$ using various initial ε values.

If we consider the range of N values used in the experiments ($N = 9, 25, 81, 289, 1089, 4225$), then we see that stationary interpolation with Gaussians does converge for the smaller values of N (at a rate better than $\mathcal{O}(h^2)$). However, the larger the value of the initial ε is taken, the sooner does the saturation occur. It is also apparent that in the case of interpolation to the sinc function small initial values of the shape parameter lead to severe ill-conditioning and subsequent instabilities especially for the tests with larger values of N . We also point out that the range of values of ε for which we can observe convergence depends on the data function f .

We will come back to the approximate approximation phenomenon in the context of quasi-interpolation and approximate moving least squares approximation in Chapters 26 and 27.

Chapter 18

The Optimality of RBF Interpolation

In this chapter we will see that within the native Hilbert spaces associated with strictly positive definite (and strictly conditionally positive definite) radial functions the radial basis function interpolant provides the *best approximation* to a given data function. This optimality of interpolants in Hilbert space is the subject of the theory of *optimal recovery* described in the late 1950s by Michael Golomb and Hans Weinberger in their paper [Golomb and Weinberger (1959)].

18.1 The Connection to Optimal Recovery

In [Golomb and Weinberger (1959)] the authors studied the following general problem:

Problem 18.1. *Given the values $f_1 = \lambda_1(f), \dots, f_N = \lambda_N(f) \in \mathbb{R}$, where $\{\lambda_1, \dots, \lambda_N\}$ is a linearly independent set of linear functionals (called information functionals yielding the information about f), how does one “best” approximate the value $\lambda(f)$ (called a feature of f) where λ is a given linear functional and f is unknown? Moreover, what is the total range of values for $\lambda(f)$?*

This is a very general problem formulation that allows not only for interpolation of function values, but also for other types of data (such as values of derivatives and integrals of f , such as averages or moments of f , etc.), as well as methods of approximation other than interpolation.

The kind of problem described above is known in the literature as an *optimal recovery problem*. Besides the seminal work by Golomb and Weinberger, optimal recovery was also studied in detail by Micchelli, Rivlin and Winograd [Micchelli et al. (1976); Micchelli and Rivlin (1977); Micchelli and Rivlin (1980); Micchelli and Rivlin (1985)].

In a Hilbert space setting the solution to this optimal recovery problem is shown to be the *minimum-norm interpolant*. More precisely, given a Hilbert space \mathcal{H} and data $f_1 = \lambda_1(f), \dots, f_N = \lambda_N(f) \in \mathbb{R}$ with $\{\lambda_1, \dots, \lambda_N\} \subseteq \mathcal{H}^*$ (the dual of \mathcal{H}), the

minimum-norm interpolant is that function $g^* \in \mathcal{H}$ that satisfies

$$\lambda_j(g^*) = f_j, \quad j = 1, \dots, N,$$

and for which

$$\|g^*\|_{\mathcal{H}} = \min_{\substack{g \in \mathcal{H} \\ \lambda_j(g) = f_j, j=1, \dots, N}} \|g\|_{\mathcal{H}}.$$

It turns out that the radial basis function interpolant with basic function Φ satisfies these criteria if \mathcal{H} is taken as the associated native space $\mathcal{N}_{\Phi}(\Omega)$.

We will present three optimality results:

- The radial basis function interpolant for any strictly conditionally positive definite function Φ is the minimum norm interpolant from $\mathcal{N}_{\Phi}(\Omega)$.
- The radial basis function interpolant provides the best approximation to f in the native space norm.
- The (cardinal form of the) radial basis function interpolant is more accurate (as measured by the pointwise error) than any other linear combination of the data.

18.2 Orthogonality in Reproducing Kernel Hilbert Spaces

The proofs of the first two “optimality theorems” require the following two lemmas. These lemmas and their corollary can also be generalized to cover the strictly conditionally positive definite case. However, to keep our discussion transparent, we present only the details of the strictly positive definite case.

Lemma 18.1. *Assume Φ is a symmetric strictly positive definite kernel on \mathbb{R}^s and let \mathcal{P}_f be the interpolant to $f \in \mathcal{N}_{\Phi}(\Omega)$ at the data sites $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subseteq \Omega$. Then*

$$\langle \mathcal{P}_f, \mathcal{P}_f - g \rangle_{\mathcal{N}_{\Phi}(\Omega)} = 0$$

for all interpolants $g \in \mathcal{N}_{\Phi}(\Omega)$, i.e., with $g(\mathbf{x}_j) = f(\mathbf{x}_j)$, $j = 1, \dots, N$.

Proof. The interpolant \mathcal{P}_f is of the form

$$\mathcal{P}_f = \sum_{j=1}^N c_j \Phi(\cdot, \mathbf{x}_j),$$

where the coefficients c_j are determined by the interpolation conditions $\mathcal{P}_f(\mathbf{x}_i) = f(\mathbf{x}_i)$, $i = 1, \dots, N$. Using this representation, the symmetry of the kernel Φ and its reproducing property we have

$$\langle \mathcal{P}_f, \mathcal{P}_f - g \rangle_{\mathcal{N}_{\Phi}(\Omega)} = \left\langle \sum_{j=1}^N c_j \Phi(\cdot, \mathbf{x}_j), \mathcal{P}_f - g \right\rangle_{\mathcal{N}_{\Phi}(\Omega)}$$

$$\begin{aligned} &= \sum_{j=1}^N c_j \langle \Phi(\cdot, \mathbf{x}_j), \mathcal{P}_f - g \rangle_{\mathcal{N}_{\Phi}(\Omega)} \\ &= \sum_{j=1}^N c_j \langle \mathcal{P}_f - g, \Phi(\cdot, \mathbf{x}_j) \rangle_{\mathcal{N}_{\Phi}(\Omega)} \\ &= \sum_{j=1}^N c_j (\mathcal{P}_f - g)(\mathbf{x}_j) \\ &= 0 \end{aligned}$$

since both \mathcal{P}_f and g interpolate f on \mathcal{X} . \square

For the next result, recall the definition of the space $H_{\Phi}(\mathcal{X})$ as the linear span

$$H_{\Phi}(\mathcal{X}) = \text{span}\{\Phi(\cdot, \mathbf{x}_j) : \mathbf{x}_j \in \mathcal{X}\}$$

(c.f. (13.1)). Clearly, $H_{\Phi}(\mathcal{X})$ is a subspace of the native space $\mathcal{N}_{\Phi}(\Omega)$.

Lemma 18.2. *Assume Φ is a strictly positive definite kernel on \mathbb{R}^s and let \mathcal{P}_f be the interpolant to $f \in \mathcal{N}_{\Phi}(\Omega)$ on $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subseteq \Omega$. Then*

$$\langle f - \mathcal{P}_f, h \rangle_{\mathcal{N}_{\Phi}(\Omega)} = 0$$

for all $h \in H_{\Phi}(\mathcal{X})$.

Proof. Any $h \in H_{\Phi}(\mathcal{X})$ can be written in the form

$$h = \sum_{j=1}^N c_j \Phi(\cdot, \mathbf{x}_j)$$

with appropriate coefficients c_j . Using this representation of h as well as the reproducing property of Φ we have

$$\begin{aligned} \langle f - \mathcal{P}_f, h \rangle_{\mathcal{N}_{\Phi}(\Omega)} &= \langle f - \mathcal{P}_f, \sum_{j=1}^N c_j \Phi(\cdot, \mathbf{x}_j) \rangle_{\mathcal{N}_{\Phi}(\Omega)} \\ &= \sum_{j=1}^N c_j \langle f - \mathcal{P}_f, \Phi(\cdot, \mathbf{x}_j) \rangle_{\mathcal{N}_{\Phi}(\Omega)} \\ &= \sum_{j=1}^N c_j (f - \mathcal{P}_f)(\mathbf{x}_j). \end{aligned}$$

This last expression, however, is zero since \mathcal{P}_f interpolates f on \mathcal{X} , i.e., $(f - \mathcal{P}_f)(\mathbf{x}_j) = 0$, $j = 1, \dots, N$. \square

The following Pythagorean theorem (or “energy splitting” theorem) is an immediate consequence of Lemma 18.2. It says that the native space “energy” of f can be split into the “energy” of the interpolant \mathcal{P}_f and that of the residual $f - \mathcal{P}_f$, which — according to Lemma 18.2 — is orthogonal to the interpolant.

Corollary 18.1. *The orthogonality property of Lemma 18.2 implies the energy split*

$$\|f\|_{\mathcal{N}_{\Phi}(\Omega)}^2 = \|f - \mathcal{P}_f\|_{\mathcal{N}_{\Phi}(\Omega)}^2 + \|\mathcal{P}_f\|_{\mathcal{N}_{\Phi}(\Omega)}^2.$$

Proof. The statement follows from

$$\begin{aligned}\|f\|_{\mathcal{N}_\Phi(\Omega)}^2 &= \|f - \mathcal{P}_f + \mathcal{P}_f\|_{\mathcal{N}_\Phi(\Omega)}^2 \\ &= \langle(f - \mathcal{P}_f) + \mathcal{P}_f, (f - \mathcal{P}_f) + \mathcal{P}_f\rangle_{\mathcal{N}_\Phi(\Omega)} \\ &= \|f - \mathcal{P}_f\|_{\mathcal{N}_\Phi(\Omega)}^2 + 2\langle f - \mathcal{P}_f, \mathcal{P}_f\rangle_{\mathcal{N}_\Phi(\Omega)} + \|\mathcal{P}_f\|_{\mathcal{N}_\Phi(\Omega)}^2\end{aligned}$$

and the fact that $\langle f - \mathcal{P}_f, \mathcal{P}_f\rangle_{\mathcal{N}_\Phi(\Omega)} = 0$ by Lemma 18.2 since $\mathcal{P}_f \in H_\Phi(\mathcal{X})$. \square

The above energy split is the fundamental idea behind a number of Krylov-type iterative algorithms for approximately solving the interpolation problem when very large data sets are involved (see, e.g., our discussion in Chapter 33 or the papers [Faul and Powell (1999); Faul and Powell (2000)] or [Schaback and Wendland (2000a)]).

18.3 Optimality Theorem I

The following theorem presents the first optimality property formulated for the general case of strictly conditionally positive definite kernels. It is taken from [Wendland (2005a)].

Theorem 18.1 (Optimality I). Suppose $\Phi \in C(\Omega \times \Omega)$ is a strictly conditionally positive definite kernel with respect to the finite-dimensional space $P \subseteq C(\Omega)$ and that \mathcal{X} is P -unisolvent. If the values f_1, \dots, f_N are given, then the interpolant \mathcal{P}_f is the minimum-(semi)norm interpolant to $\{f_j\}_{j=1}^N$, i.e.,

$$|\mathcal{P}_f|_{\mathcal{N}_\Phi(\Omega)} = \min_{\substack{g \in \mathcal{N}_\Phi(\Omega) \\ g(\mathbf{x}_j) = f_j, j=1, \dots, N}} |g|_{\mathcal{N}_\Phi(\Omega)}.$$

Proof. We consider only the strictly positive definite case. Consider an arbitrary interpolant $g \in \mathcal{N}_\Phi(\Omega)$ to f_1, \dots, f_N . Then Lemma 18.1 gives us

$$\langle \mathcal{P}_f, \mathcal{P}_f - g \rangle_{\mathcal{N}_\Phi(\Omega)} = 0.$$

This orthogonality relation gives us

$$\begin{aligned}|\mathcal{P}_f|_{\mathcal{N}_\Phi(\Omega)}^2 &= \langle \mathcal{P}_f, \mathcal{P}_f - g + g \rangle_{\mathcal{N}_\Phi(\Omega)} \\ &= \langle \mathcal{P}_f, \mathcal{P}_f - g \rangle_{\mathcal{N}_\Phi(\Omega)} + \langle \mathcal{P}_f, g \rangle_{\mathcal{N}_\Phi(\Omega)} \\ &= \langle \mathcal{P}_f, g \rangle_{\mathcal{N}_\Phi(\Omega)},\end{aligned}$$

and the Cauchy-Schwarz inequality yields

$$|\mathcal{P}_f|_{\mathcal{N}_\Phi(\Omega)}^2 \leq |\mathcal{P}_f|_{\mathcal{N}_\Phi(\Omega)} |g|_{\mathcal{N}_\Phi(\Omega)},$$

so that the statement follows. \square

As in our earlier use of conditionally positive definite functions, the space P mentioned in Theorem 18.1 is usually taken as the space Π_{m-1}^s of multivariate polynomials. Also, if Φ is strictly positive definite then the semi-norms in Theorem 18.1 become norms.

Example 18.1. We said earlier that the native space of thin plate splines $\phi(r) = r^2 \log r$, $r = \|\mathbf{x}\|_2$ with $\mathbf{x} = (x, y) \in \mathbb{R}^2$ is given by the Beppo-Levi space $BL_2(\mathbb{R}^2)$. Now, the corresponding semi-norm in the Beppo-Levi space $BL_2(\mathbb{R}^2)$ is (c.f. (13.6))

$$|f|_{BL_2(\mathbb{R}^2)}^2 = \int_{\mathbb{R}^2} \left(\left| \frac{\partial^2 f}{\partial x^2}(\mathbf{x}) \right|^2 + 2 \left| \frac{\partial^2 f}{\partial x \partial y}(\mathbf{x}) \right|^2 + \left| \frac{\partial^2 f}{\partial y^2}(\mathbf{x}) \right|^2 \right) d\mathbf{x},$$

which is the bending energy of a thin plate. By Theorem 18.1 the thin plate spline interpolant minimizes this energy. This explains the name of these functions.

18.4 Optimality Theorem II

Another nice property of the radial basis function interpolant is the fact that it is at the same time the best Hilbert-space approximation to the given data, and thus not just any projection of f but the *orthogonal projection*. Again, we formulate the theorem for the strictly conditionally positive definite case and provide details only for the strictly positive definite case.

Theorem 18.2 (Optimality II). Let

$$\begin{aligned}H_\Phi(\mathcal{X}) &= \{h = \sum_{j=1}^N c_j \Phi(\cdot, \mathbf{x}_j) + p : p \in P \\ &\quad \text{and } \sum_{j=1}^N c_j q(\mathbf{x}_j) = 0 \text{ for all } q \in P \text{ and } \mathbf{x}_j \in \mathcal{X}\},\end{aligned}$$

where $\Phi \in C(\Omega \times \Omega)$ is a strictly conditionally positive definite kernel with respect to the finite-dimensional space $P \subseteq C(\Omega)$ and \mathcal{X} is P -unisolvent. If only the values $f_1 = f(\mathbf{x}_1), \dots, f_N = f(\mathbf{x}_N)$ are given, then the interpolant \mathcal{P}_f is the best approximation to f from $H_\Phi(\mathcal{X})$ in $\mathcal{N}_\Phi(\Omega)$, i.e.,

$$|f - \mathcal{P}_f|_{\mathcal{N}_\Phi(\Omega)} \leq |f - h|_{\mathcal{N}_\Phi(\Omega)}$$

for all $h \in H_\Phi(\mathcal{X})$.

Proof. We consider only the strictly positive definite case. As explained in Section 13.2, the native space $\mathcal{N}_\Phi(\Omega)$ is the completion of $H_\Phi(\Omega)$ with respect to the $\|\cdot\|_\Phi$ -norm so that $\|f\|_\Phi = \|f\|_{\mathcal{N}_\Phi(\Omega)}$ for all $f \in H_\Phi(\Omega)$. Also, $\mathcal{X} \subseteq \Omega$. Therefore, we can characterize the best approximation g^* to f from $H_\Phi(\mathcal{X})$ by

$$\langle f - g^*, h \rangle_{\mathcal{N}_\Phi(\Omega)} = 0 \quad \text{for all } h \in H_\Phi(\mathcal{X}).$$

However, Lemma 18.2 shows that $g^* = \mathcal{P}_f$ satisfies this relation. \square

These optimality properties of radial basis function interpolants play an important role in applications such as in the design of support vector machines in statistical learning theory or the numerical solution of partial differential equations.

The optimality results above imply that one could also start with some Hilbert space \mathcal{H} with norm $\|\cdot\|_{\mathcal{H}}$ and ask to find the minimum norm interpolant (*i.e.*, Hilbert space best approximation) to some given data. In this way any given space defines a set of *optimal basis functions*, generated by the reproducing kernel of \mathcal{H} . This is how Duchon approached the subject in his papers [Duchon (1976); Duchon (1977); Duchon (1978); Duchon (1980)]. More recently, Kybic, Blu and Unser [Kybík *et al.* (2002a); Kybic *et al.* (2002b)] take this point of view and explain from a sampling theory point of view how the thin plate splines can be interpreted as fundamental solutions of the differential operator defining the semi-norm in the Beppo-Levi space $BL_2(\mathbb{R}^2)$, and thus radial basis functions can be viewed as *Green's functions*.

18.5 Optimality Theorem III

The third optimality result is in the context of quasi-interpolation, *i.e.*,

Theorem 18.3 (Optimality III). *Suppose $\Phi \in C(\Omega \times \Omega)$ is a strictly conditionally positive definite kernel with respect to the finite-dimensional space $P \subseteq C(\Omega)$. Suppose \mathcal{X} is P -unisolvent and $\mathbf{x} \in \Omega$ is fixed. Let $u_j^*(\mathbf{x})$, $j = 1, \dots, N$, be the values at \mathbf{x} of the cardinal basis functions for interpolation with Φ . Then*

$$\left| f(\mathbf{x}) - \sum_{j=1}^N f(\mathbf{x}_j) u_j^*(\mathbf{x}) \right| \leq \left| f(\mathbf{x}) - \sum_{j=1}^N f(\mathbf{x}_j) u_j \right|$$

for all choices of $u_1, \dots, u_N \in \mathbb{R}$ with $\sum_{j=1}^N u_j p(\mathbf{x}_j) = p(\mathbf{x})$ for any $p \in P$.

Theorem 18.3 is proved in [Wendland (2005a)]. It says in particular that the minimum norm interpolant \mathcal{P}_f is also more accurate (in the pointwise sense) than any linear combination of the given data values that reproduce P .

Chapter 19

Least Squares RBF Approximation with MATLAB

Up to now we have looked only at interpolation. However, many times it makes more sense to approximate the given data by a least squares fit. This is especially true if the data are contaminated with noise, or if there are so many data points that efficiency considerations force us to approximate from a space spanned by fewer basis functions than data points.

19.1 Optimal Recovery Revisited

As we saw in Chapter 18 we can interpret radial basis function interpolation as a constrained optimization problem, *i.e.*, the RBF interpolant automatically minimizes the native space norm among all interpolants in the native space. We now take this point of view again, but start with a more general formulation. Let us assume we are seeking a function \mathcal{P}_f of the form

$$\mathcal{P}_f(\mathbf{x}) = \sum_{j=1}^M c_j \Phi(\mathbf{x}, \mathbf{x}_j), \quad \mathbf{x} \in \mathbb{R}^s,$$

where the number M of basis functions is in general less than or equal the number N of data sites. We then want to determine the coefficients $\mathbf{c} = [c_1, \dots, c_M]^T$ so that we minimize the quadratic form

$$\frac{1}{2} \mathbf{c}^T Q \mathbf{c} \tag{19.1}$$

with some symmetric positive definite matrix Q subject to the linear constraints

$$A\mathbf{c} = \mathbf{f} \tag{19.2}$$

where A is an $N \times M$ matrix with full rank, and the right-hand side $\mathbf{f} = [f_1, \dots, f_N]^T$ is given. Such a constrained quadratic minimization problem can be converted to a system of linear equations by introducing *Lagrange multipliers* $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_N]^T$, *i.e.*, we consider finding the minimum of

$$\frac{1}{2} \mathbf{c}^T Q \mathbf{c} - \boldsymbol{\lambda}^T [A\mathbf{c} - \mathbf{f}] \tag{19.3}$$

with respect to \mathbf{c} and $\boldsymbol{\lambda}$. Since Q is assumed to be a positive definite matrix, it is well known that the functional to be minimized is convex, and thus has a unique minimum. Therefore, the standard necessary condition for such a minimum (which is obtained by differentiating with respect to \mathbf{c} and $\boldsymbol{\lambda}$ and finding the zeros of those derivatives) is also sufficient. This leads to

$$\begin{aligned} Q\mathbf{c} - A^T \boldsymbol{\lambda} &= \mathbf{0} \\ A\mathbf{c} - \mathbf{f} &= \mathbf{0} \end{aligned}$$

or, in matrix form,

$$\begin{bmatrix} Q - A^T \\ A \quad O \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{f} \end{bmatrix}.$$

By applying (block) Gaussian elimination to this block matrix (Q is invertible since it is assumed to be positive definite) we get

$$\boldsymbol{\lambda} = (AQ^{-1}A^T)^{-1}\mathbf{f} \quad (19.4)$$

$$\mathbf{c} = Q^{-1}A^T(AQ^{-1}A^T)^{-1}\mathbf{f}. \quad (19.5)$$

In particular, if the quadratic form represents the native space norm of the interpolant $\mathcal{P}_f = \sum_{j=1}^M c_j \Phi(\cdot, \mathbf{x}_j)$, i.e.,

$$\|\mathcal{P}_f\|_{\mathcal{N}_\Phi(\Omega)}^2 = \sum_{i=1}^M \sum_{j=1}^M c_i c_j \Phi(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{c}^T Q \mathbf{c}$$

with $Q_{ij} = \Phi(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{c} = [c_1, \dots, c_M]^T$, and the linear side conditions are the interpolation conditions

$$A\mathbf{c} = \mathbf{f} \iff \mathcal{P}_f(\mathbf{x}_i) = f_i, \quad i = 1, \dots, M,$$

with $A = A^T = Q$ (symmetric), the same \mathbf{c} as above and data vector $\mathbf{f} = [f_1, \dots, f_M]^T$, then we see that the Lagrange multipliers (19.4) become

$$\boldsymbol{\lambda} = A^{-1}\mathbf{f}$$

and the coefficients are given by

$$\mathbf{c} = \boldsymbol{\lambda}$$

via (19.5). Therefore, as we saw earlier, the minimum norm interpolant is obtained by solving the interpolation equations alone.

19.2 Regularized Least Squares Approximation

Since we took the more general point of view that \mathcal{P}_f is generated by M basis functions, and N linear constraints are specified, the above formulation also covers both over- and under-determined least squares fitting where the quadratic form

$\mathbf{c}^T Q \mathbf{c}$ represents an added *smoothing* (or *regularization*) term. This term is not required to obtain a unique solution of the system $A\mathbf{c} = \mathbf{f}$ in the over-determined case ($N \geq M$), but in the under-determined case such a constraint is needed (c.f. the solution of under-determined linear systems via singular value decomposition in the numerical linear algebra literature (e.g., [Trefethen and Bau (1997)])).

Usually the regularized least squares approximation problem is formulated as minimization of

$$\begin{aligned} &\frac{1}{2} \mathbf{c}^T Q \mathbf{c} + \omega \sum_{j=1}^N (\mathcal{P}_f(\mathbf{x}_j) - f_j)^2 \\ &\iff \frac{1}{2} \mathbf{c}^T Q \mathbf{c} + \omega (\mathbf{c}^T A - \mathbf{f})^T (\mathbf{c}^T A - \mathbf{f}). \end{aligned} \quad (19.6)$$

The quadratic form $\mathbf{c}^T Q \mathbf{c}$ controls the smoothness of the fitting function and the least squares term measures the closeness to the data. The parameter ω controls the tradeoff between these two terms with a large value of ω shifting the balance toward increased pointwise accuracy.

The formulation (19.6) is used in *regularization theory* (see, e.g., [Evgeniou et al. (2000); Girosi (1998)]). The same formulation is also used in *penalized least squares* fitting (see, e.g., [von Golitschek and Schumaker (1990)]), the literature on *smoothing splines* [Reinsch (1967); Schoenberg (1964)], and in papers by Wahba on thin plate splines (e.g., [Kimeldorf and Wahba (1971); Wahba (1979); Wahba (1990b); Wahba and Luo (1997); Wahba and Wendelberger (1980)]). In fact, the idea of smoothing a data fitting process by this kind of formulation seems to go back to at least [Whittaker (1923)]. In practice a penalized least squares formulation is especially useful if the data f_i cannot be completely trusted, i.e., they are contaminated by noise. The problem of minimizing (19.6) is also known as *ridge regression* in the statistics literature. The regularization parameter ω is usually chosen using generalized cross validation.

If we restrict ourselves to working with square symmetric systems, i.e., $A = A^T$, and assume the smoothness functional is given by the native space norm, i.e., $Q = A$, then we obtain the minimizer of the unconstrained quadratic functional (19.6) by solving the linear system

$$\left(A + \frac{1}{2\omega} I \right) \mathbf{c} = \mathbf{f} \quad (19.7)$$

which is the result of setting the derivative of (19.6) with respect to \mathbf{c} equal to zero. Thus, ridge regression corresponds to a diagonal stabilization/regularization of the usual interpolation system $A\mathbf{c} = \mathbf{f}$. This approach is especially useful for smoothing of noisy data. We present an implementation of this method and some numerical examples below in Section 19.4.

19.3 Least Squares Approximation When RBF Centers Differ from Data Sites

We are now interested in the more general setting where we still sample the given function f on the set $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ of data sites, but now introduce a second set $\Xi = \{\xi_i\}_{i=1}^M$ at which we center the basis functions. Usually we will have $M \leq N$, and the case $M = N$ with $\Xi = \mathcal{X}$ recovers the traditional interpolation setting discussed in earlier chapters. Therefore, we can let the RBF approximant be of the form

$$Q_f(\mathbf{x}) = \sum_{j=1}^M c_j \Phi(\mathbf{x}, \xi_j), \quad \mathbf{x} \in \mathbb{R}^s. \quad (19.8)$$

The coefficients c_j can be found as the least squares solution of $A\mathbf{c} = \mathbf{f}$, i.e., by minimizing $\|Q_f - f\|_2^2$, where the ℓ_2 -norm

$$\|f\|_2^2 = \sum_{i=1}^N [f(\mathbf{x}_i)]^2, \quad \mathbf{x}_i \in \mathcal{X},$$

is induced by the discrete inner product

$$\langle f, g \rangle = \sum_{i=1}^N f(\mathbf{x}_i)g(\mathbf{x}_i), \quad \mathbf{x}_i \in \mathcal{X}. \quad (19.9)$$

This approximation problem has a unique solution if the (rectangular) *collocation matrix* A with entries

$$A_{jk} = \Phi(\mathbf{x}_j, \xi_k), \quad j = 1, \dots, N, \quad k = 1, \dots, M,$$

has full rank.

If the centers in Ξ are chosen to form a subset of the data locations \mathcal{X} , then A does have full rank provided the radial basis functions are selected according to our previous chapters on interpolation. This is true, since in this case A will have an $M \times M$ square submatrix which is non-singular (by virtue of being an *interpolation matrix*).

The over-determined linear system $A\mathbf{c} = \mathbf{f}$ which arises in the solution of the least squares problem can be solved using standard algorithms from numerical linear algebra such as QR or singular value decomposition. Therefore the MATLAB code for RBF least squares approximation is almost identical to that for interpolation.

Program 19.1 presents an example for least squares approximation in 2D. Now we define two sets of points: the data points (defined in lines 3 and 8), and the centers (defined in lines 4, 6 and 7). Note that we first load the centers since our data files `Data2D_1089h` and `Data2D_81u` contain a variable `dsites` which we want to use for our data sites. Loading the data sites first, and then the centers would lead to unwanted overwriting of the values in `dsites`. The solution of the least squares problem is computed on line 16 using backslash matrix left division (`\` or `mldivide`) which automatically produces a least squares solution. The subroutines `PlotSurf` and `PlotError2D` are provided in Appendix C.

Program 19.1. RBFApproximation2D.m

```
% RBFApproximation2D
% Script that performs basic 2D RBF least squares approximation
% Calls on: DistanceMatrix, PlotSurf, PlotError2D
1 rbf = @(e,r) exp(-(e*r).^2); ep = 1;
2 testfunction = @(x,y) sinc(x).*sinc(y);
3 N = 1089; gridtype = 'h';
4 M = 81; grid2type = 'u';
5 neval = 40;
% Load centers
6 name = sprintf('Data2D_%d%s',M,grid2type); load(name)
7 ctrs = dsites;
% Load data points
8 name = sprintf('Data2D_%d%s',N,gridtype); load(name)
% Compute distance matrix between data sites and centers
9 DM_data = DistanceMatrix(dsites,ctrs);
% Build collocation matrix
10 CM = rbf(ep,DM_data);
% Create right-hand side vector, i.e.,
% evaluate the test function at the data points.
11 rhs = testfunction(dsites(:,1),dsites(:,2));
% Create neval-by-neval equally spaced evaluation
% locations in the unit square
12 grid = linspace(0,1,neval); [xe,ye] = meshgrid(grid);
13 epoints = [xe(:) ye(:)];
% Compute distance matrix between evaluation points and centers
14 DM_eval = DistanceMatrix(epoints,ctrs);
15 EM = rbf(ep,DM_eval);
% Compute RBF least squares approximation
16 Pf = EM * (CM\rhs);
% Compute exact solution, i.e., evaluate test
% function on evaluation points
17 exact = testfunction(epoints(:,1),epoints(:,2));
% Compute maximum error on evaluation grid
18 maxerr = norm(Pf-exact,inf);
% Plots
19 figure; fview = [100,30]; % viewing angles for plot
20 caption = sprintf('%d data sites and %d centers',N,M);
21 title(caption);
22 plot(dsites(:,1),dsites(:,2),'bo',ctrs(:,1),ctrs(:,2),'r+');
23 PlotSurf(xe,ye,Pf,neval,exact,maxerr,fview);
24 PlotError2D(xe,ye,Pf,exact,maxerr,neval,fview);
```

Output from RBFApproximation2D.m is presented in Figure 19.1 and the top part of Figure 19.2.

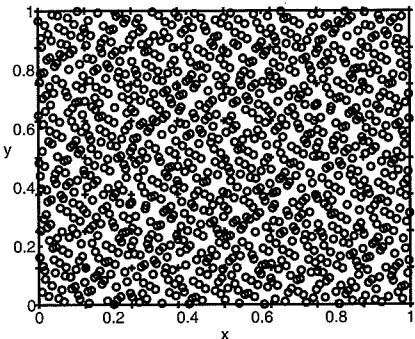


Fig. 19.1 1089 Halton data sites (o) and 81 uniform centers (+).

If $\varepsilon = 1$, then the collocation matrix is rank deficient with MATLAB reporting a numerical rank of 58. In order to have a full numerical rank for this problem ε needs to be at least 2.2 (in which case the maximum error deteriorates to 5.255591e-004 instead of 2.173460e-007 for $\varepsilon = 1$, *c.f.* the top part of Figure 19.2). There is not much theory available for the case of differing centers and data sites. We present what is known in the next chapter. Some care needs to be taken when computing least squares solutions based on sets of differing centers and data sites.

19.4 Least Squares Smoothing of Noisy Data

We present two strategies for dealing with noisy data, *i.e.*, data that we consider to be not reliable due to, *e.g.*, measurement or transmission errors. This situation arises frequently in practice. We simulate a set of noisy data by sampling Franke's test function at a set \mathcal{X} of data sites, and then adding uniformly distributed random noise of various strengths. For this experiment we use thin plate splines since their native space norm corresponds to the bending energy of a thin plate and thus they have a tendency to produce "visually pleasing" smooth and tight surfaces.

Since the thin plate splines have a singularity at the origin a little extra care needs to be taken with their implementation. The MATLAB script `tps.m` we use for our implementation of this basic function is included in Appendix C as Program C.4.

Our first strategy is to compute a straightforward least squares approximation to the (large) set of data using a (small) set of basis functions as we did in the previous section. In the statistics literature this approach is known as *regression splines*. We will not address the question of how to choose the centers for the basis functions at this point.

We use a modification of program RBFApproximation2D.m that allows us to use

thin plate splines with the added linear polynomial term. These changes can be found on lines 1, 15, 16, 19 and 24 of Program 19.2. Also, we now replace the sinc test function by Franke's function (2.2). The noise is added to the right-hand side of the linear system on line 18. This modification adds 3% noise to the data.

Program 19.2. RBFApproximation2Dlinear.m

```
% RBFApproximation2Dlinear
% Script that performs 2D RBF least squares approximation with
% linear reproduction for noisy data
% Calls on: tps, DistanceMatrix
1 rbf = @tps; ep = 1; % defined in tps.m (see Appendix C)
% Define Franke's function as testfunction
2 f1 = @(x,y) 0.75*exp(-((9*x-2).^2+(9*y-2).^2)/4);
3 f2 = @(x,y) 0.75*exp(-((9*x+1).^2/49+(9*y+1).^2/10));
4 f3 = @(x,y) 0.5*exp(-((9*x-7).^2+(9*y-3).^2)/4);
5 f4 = @(x,y) 0.2*exp(-((9*x-4).^2+(9*y-7).^2)/2);
6 testfunction = @(x,y) f1(x,y)+f2(x,y)+f3(x,y)-f4(x,y);
7 N = 1089; gridtype = 'h';
8 M = 81; grid2type = 'u';
9 neval = 40;
% Load centers
10 name = sprintf('Data2D_%d%s',M,grid2type); load(name)
11 ctrs = dsites;
% Load data points
12 name = sprintf('Data2D_%d%s',N,gridtype); load(name)
% Compute distance matrix between data sites and centers
13 DM_data = DistanceMatrix(dsites,ctrs);
14 CM = rbf(ep,DM_data); % Collocation matrix
% Add extra columns and rows for linear reproduction
15 PM = [ones(N,1) dsites]; PtM = [ones(M,1) ctrs]';
16 CM = [CM PM; [PtM zeros(3,3)]]; % Create right-hand side vector and add noise
17 rhs = testfunction(dsites(:,1),dsites(:,2));
18 rhs = rhs + 0.03*randn(size(rhs)); % Add zeros for linear (2D) reproduction
19 rhs = [rhs; zeros(3,1)]; % Create neval-by-neval equally spaced evaluation locations
% in the unit square
20 grid = linspace(0,1,neval); [xe,ye] = meshgrid(grid);
21 epoints = [xe(:) ye(:)]; % Compute distance matrix between evaluation points and centers
22 DM_eval = DistanceMatrix(epoints,ctrs);
```

```

23 EM = rbf(ep,DM_eval); % Evaluation matrix
% Add columns for linear reproduction
24 PM = [ones(neval^2,1) epoints]; EM = [EM PM];
% Compute RBF least squares approximation
25 Pf = EM * (CM\rhs);
% Compute exact solution, i.e.,
% evaluate test function on evaluation points
26 exact = testfunction(epoints(:,1),epoints(:,2));
% Compute maximum error on evaluation grid
27 maxerr = norm(Pf-exact,inf);
% Plots
28 figure; fview = [160,20]; % viewing angles for plot
29 caption = sprintf('%d data sites and %d centers',N,M);
30 title(caption);
31 plot(dsites(:,1),dsites(:,2),'bo',ctrs(:,1),ctrs(:,2),'r+');
32 PlotSurf(xe,ye,Pf,neval,exact,maxerr,fview);
33 PlotError2D(xe,ye,Pf,exact,maxerr,neval,fview);

```

Program `RBFApproxiation2Dlinear.m` was used to produce the right plot in the bottom part of Figure 19.2 and the entries in line 2 of Table 19.1. Clearly, this simple least squares approach performs much better than straightforward interpolation to the noisy data (see the left plot the bottom part of Figure 19.2 and line 1 of Table 19.1). Moreover, this least squares approximation is also much cheaper to compute. However, as we pointed out earlier, it is not clear how to choose the smaller set of RBF centers, and what is even more unsettling, there is not much mathematical theory (see the next section) to guarantee if (or when) this approach is well-posed, *i.e.*, the collocation matrix has full rank.

Table 19.1 Errors and condition numbers for various least squares approximants to noisy data.

method	ω	RMS-error	max-error	cond(A)
Interpolation	∞	2.482624e-002	9.914837e-002	1.502900e+007
LSQ approximation	NA	9.665743e-003	5.490050e-002	NA
Ridge regression	1000	1.713843e-002	7.580288e-002	2.537652e+006
Ridge regression	100	1.078358e-002	4.215865e-002	3.839384e+005
Ridge regression	10	9.173961e-003	3.349371e-002	4.571167e+004
Ridge regression	1	2.764272e-002	1.041350e-001	9.317936e+003

Another strategy for smoothing of noisy data is the ridge regression method explained earlier (see (19.7)). This method is popular in the statistics and neural network community.

The nature of the MATLAB program for ridge regression is very similar to that for RBF interpolation. We present a version for ridge regression with thin plate splines

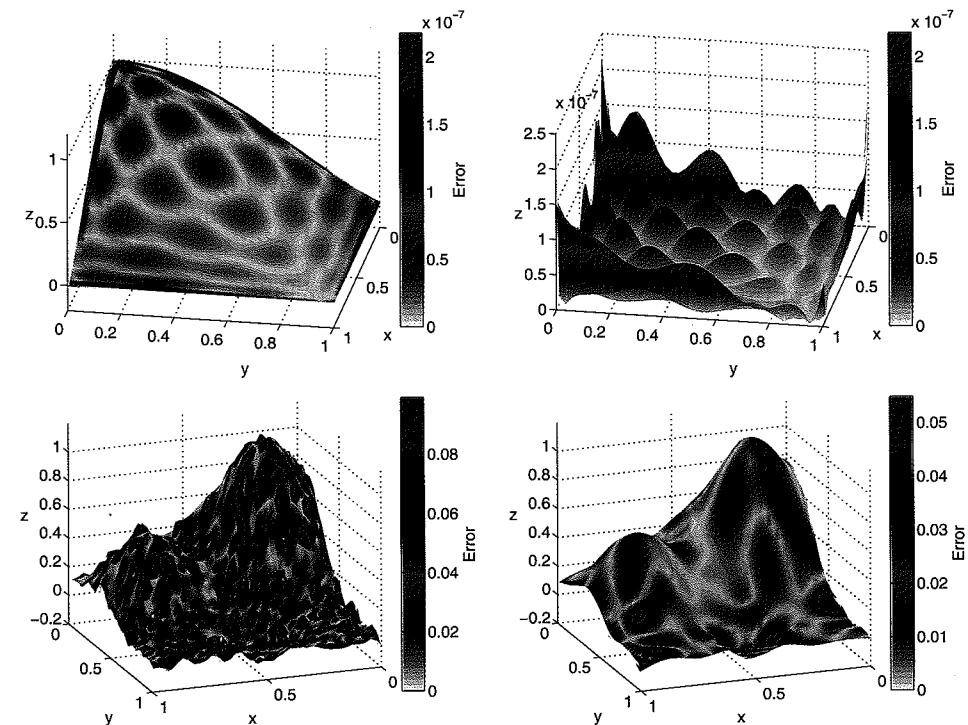


Fig. 19.2 Top: Least squares approximation (left) to 1089 data points sampled from 2D sinc function with 81 Gaussian basis functions with $\varepsilon = 1$ and maximum error (right) false-colored by magnitude of error. Bottom: Thin plate spline interpolant (left) to 1089 noisy data points sampled from Franke's function, and least squares approximation with 81 uniformly spaced thin plate spline basis functions (right) false-colored by magnitude of error.

(including the linear term in the basis expansion) for smoothing of noisy data. The smoothing parameter ω of (19.7) is defined on line 7 of Program 19.3, and the diagonal stabilization of the (interpolation) matrix A is performed on line 17. Note that the stabilization only affects the A part of the matrix, and not the extra rows and columns added for polynomial precision.

Program 19.3. TPS_RidgeRegression2D.m

```

% TPS_RidgeRegression2D
% Script that performs 2D TPS-RBF approximation with reproduction of
% linear functions and smoothing via ridge regression
% Calls on: tps, DistanceMatrix
% Use TPS (defined in tps.m, see Appendix C)
1 rbf = @tps; ep = 1;
% Define Franke's function as testfunction
2 f1 = @(x,y) 0.75*exp(-((9*x-2).^2+(9*y-2).^2)/4);

```

```

3 f2 = @(x,y) 0.75*exp(-((9*x+1).^2/49+(9*y+1).^2/10));
4 f3 = @(x,y) 0.5*exp(-((9*x-7).^2+(9*y-3).^2)/4);
5 f4 = @(x,y) 0.2*exp(-((9*x-4).^2+(9*y-7).^2));
6 testfunction = @(x,y) f1(x,y)+f2(x,y)+f3(x,y)-f4(x,y);
7 omega = 100; % Smoothing parameter
8 N = 1089; gridtype = 'h';
9 neval = 40;
% Load data points
10 name = sprintf('Data2D_%d%s',N,gridtype); load(name)
11 ctrs = dsites;
% Compute distance matrix between data sites and centers
12 DM_data = DistanceMatrix(dsites,ctrs);
% Create noisy right-hand side vector
13 rhs = testfunction(dsites(:,1),dsites(:,2));
14 rhs = rhs + 0.03*randn(size(rhs));
% Add zeros for 2D linear reproduction
15 rhs = [rhs; zeros(3,1)];
% Compute interpolation matrix and add diagonal regularization
16 IM = rbf(ep,DM_data);
17 IM = IM + eye(size(IM))/(2*omega);
% Add extra columns and rows for linear reproduction
18 PM = [ones(N,1) dsites]; IM = [IM PM; [PM' zeros(3,3)]]; 
19 fprintf('Condition number estimate: %e\n',condest(IM))
% Create neval-by-neval equally spaced evaluation locations
% in the unit square
20 grid = linspace(0,1,neval); [xe,ye] = meshgrid(grid);
21 epoints = [xe(:) ye(:)];
% Compute distance matrix between evaluation points and centers
22 DM_eval = DistanceMatrix(epoints,ctrs);
% Compute evaluation matrix and add columns for linear precision
23 EM = rbf(ep,DM_eval);
24 PM = [ones(neval^2,1) epoints]; EM = [EM PM];
% Compute RBF interpolant
25 Pf = EM * (IM\rhs);
% Compute exact solution, i.e.,
% evaluate test function on evaluation points
26 exact = testfunction(epoints(:,1),epoints(:,2));
% Compute maximum error on evaluation grid
27 maxerr = norm(Pf-exact,inf);
% Plots
28 fview = [160,20]; % viewing angles for plot
29 PlotSurf(xe,ye,Pf,neval,exact,maxerr,fview);

```

```
30 PlotError2D(xe,ye,Pf,exact,maxerr,neval,fview);
```

The results for our examples computed with Program 19.3 are shown in Figure 19.3 as well as in lines 3–6 of Table 19.1. These results illustrate very nicely the smoothing effect obtained by varying ω . A very large value of ω emphasizes the fitting component of the functional to be minimized in (19.6) resulting in a rather rough surface, while a small value of ω gives preference to the smoothing term. The “optimal” value of ω lies somewhere in the middle. In practice one would usually use cross validation to obtain the optimal value of ω .

Besides the visual smoothing of the approximating surface, a small value of ω also has a stabilizing effect on the collocation matrix. The diagonal of the matrix becomes more and more dominant. The condition estimates in Table 19.1 also verify this behavior.

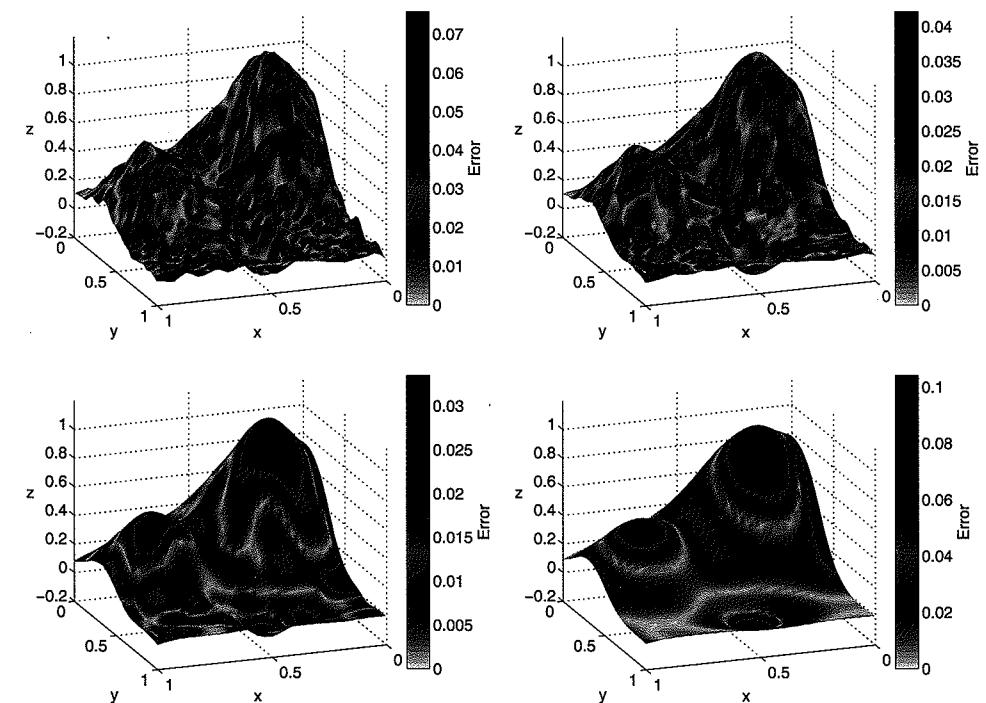


Fig. 19.3 Thin plate spline ridge regression to 1089 noisy data points sampled from Franke's function with $\omega = 1000$ (top left), $\omega = 100$ (top right), $\omega = 10$ (bottom left), and $\omega = 1$ (bottom right).

Chapter 20

Theory for Least Squares Approximation

In this chapter we give a brief account of the theoretical results known for least squares approximation with radial basis functions. These results include extensions of the RBF interpolation theory to cover well-posedness for the situation in which centers Ξ and data sites \mathcal{X} differ. We also present some recent error estimates for least squares approximation.

20.1 Well-Posedness of RBF Least Squares Approximation

The results mentioned here are due to Quak, Sivakumar and Ward [Sivakumar and Ward (1993); Quak *et al.* (1993)]. The first paper deals with discrete, the second with continuous least squares approximation. In both papers the authors do not discuss the collocation matrix A we used in the previous chapter, but rather base their results on the non-singularity of the coefficient matrix obtained from a system of normal equations.

In the discrete setting they use the inner product (19.9) which induces the ℓ_2 norm, and then discuss non-singularity of the *Gram matrix* G that occurs in the following system of *normal equations*

$$G\mathbf{c} = \mathbf{w}, \quad (20.1)$$

where the entries of G are the ℓ_2 inner products of the radial basis functions, *i.e.*,

$$G_{jk} = \langle \Phi(\cdot, \xi_j), \Phi(\cdot, \xi_k) \rangle = \sum_{i=1}^N \Phi(\mathbf{x}_i, \xi_j) \Phi(\mathbf{x}_i, \xi_k), \quad j, k = 1, \dots, M,$$

and the right-hand side vector \mathbf{w} in (20.1) is given by

$$\mathbf{w}_j = \langle \Phi(\cdot, \xi_j), \mathbf{f} \rangle = \sum_{i=1}^N \Phi(\mathbf{x}_i, \xi_j) f(\mathbf{x}_i), \quad j = 1, \dots, M.$$

Note that in the interpolation case with $M = N$ and $\Xi = \mathcal{X}$ (*i.e.*, coinciding centers and data sites) we have

$$\langle \Phi(\cdot, \mathbf{x}_j), \Phi(\cdot, \mathbf{x}_k) \rangle = \langle \Phi(\cdot, \mathbf{x}_j), \Phi(\cdot, \mathbf{x}_k) \rangle_{\mathcal{N}_\Phi(\Omega)} = \Phi(\mathbf{x}_j, \mathbf{x}_k)$$

so that G is just the interpolation matrix A . This provides yet another way of saying that the interpolation matrix A is also the system matrix for the normal equations in the case of best approximation with respect to the native space norm — a fact already mentioned earlier in Chapter 18 on optimal recovery.

In both papers, [Sivakumar and Ward (1993)] as well as [Quak *et al.* (1993)], even the formulation of the main theorems is very technical. We therefore just try to give a feel for their results.

Essentially, the authors show that the Gram matrix for certain radial basis functions (norm, (inverse) multiquadratics, and Gaussians) is non-singular if the centers $\Xi = \{\xi_k, k = 1, \dots, M\}$ are sufficiently well distributed and the data points $\mathcal{X} = \{x_j, j = 1, \dots, N\}$ are fairly evenly clustered about the centers with the diameter of the clusters being relatively small compared to the separation distance of the data points. Figure 20.1 illustrates this clustering idea.

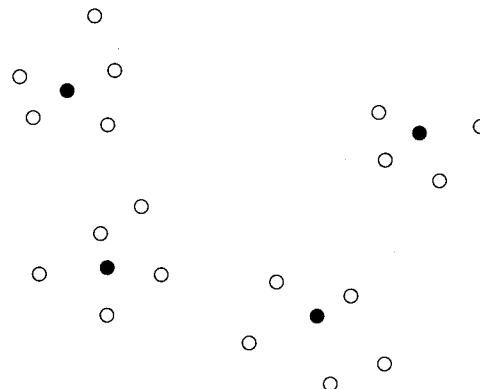


Fig. 20.1 Clusters of data points \circ around well separated centers \bullet .

One of the key ingredients in the proof of the non-singularity of G is to set up an interpolation matrix B for which the basis functions are centered at certain representatives of the clusters of knots about the data sites. One then splits the matrix B (which is non-symmetric in general) into a part that is symmetric and one that is anti-symmetric, a standard strategy in linear algebra, *i.e.*, $B = B_1 + B_2$ where B_1 and B_2 are defined by

$$B_1 = \frac{B + B^T}{2} \quad (\text{symmetric}),$$

$$B_2 = \frac{B - B^T}{2} \quad (\text{anti-symmetric}).$$

Then, lower estimates for the norm of these two parts are found and used to conclude that, under certain restrictions, G is non-singular.

As a by-product of this argumentation the authors obtain a proof for the non-singularity of *interpolation* matrices for the case in which the centers of the basis

functions are chosen different from the data sites, namely as small perturbations thereof.

20.2 Error Bounds for Least Squares Approximation

In the case of basis functions centered at the points of an infinite lattice de Boor, DeVore and Ron [de Boor *et al.* (1994b); Ron (1992)] discussed L_2 -approximation orders for radial basis functions.

More recently, [Ward (2004)] provided error bounds for least squares approximation at scattered centers and in finite domains. His work is closely linked to the results in [Narcowich *et al.* (2005)] discussed earlier in Chapter 15.

A typical result is that the continuous least squares error for approximation based on the compactly supported Wendland functions is

$$\min_{Q_f \in H_\Phi(\mathcal{X})} \|f - Q_f\|_{L_2(\Omega)} \leq C_{k,s,\Omega,\Phi} h^\tau \|f\|_{W_2^s(\Omega)}.$$

Here $\Omega \subset \mathbb{R}^s$ is a bounded Lipschitz domain which satisfies an interior cone condition, $\mathcal{X} \subset \Omega$ is the set of centers, $H_\Phi(\mathcal{X}) = \text{span}\{\Phi(\cdot - x_j), x_j \in \mathcal{X}\}$, k is such that $\tau = k + \sigma$ with $0 < \sigma \leq 1$, and $\tau > s/2$ measures the decay of the Fourier transform of Φ , *i.e.*,

$$c_1(1 + \|\omega\|_2^2)^{-\tau} \leq \hat{\Phi}(\omega) \leq c_2(1 + \|\omega\|_2^2)^{-\tau}, \quad \|\omega\| \rightarrow \infty, \omega \in \mathbb{R}^s,$$

with positive constants c_1 and c_2 so that the native space of Φ is given by the Sobolev space $W_2^s(\mathbb{R}^s)$. This error bound is of the same order as the one for interpolation (*c.f.* Theorem 15.3). We refer the reader to [Ward (2004)] for more details.