ACCURACY AND STABILITY OF GLOBAL RADIAL BASIS FUNCTION METHODS FOR THE NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

by

Rodrigo B. Platte

A dissertation submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Applied Mathematics

Summer 2005

© 2005 Rodrigo B. Platte
All Rights Reserved
ACCURACY AND STABILITY OF GLOBAL RADIAL BASIS FUNCTION METHODS FOR THE NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

by

Rodrigo B. Platte

Approved:  
Philip Broadbridge, Ph.D.  
Chairman of the Department of Mathematical Sciences

Approved:  
Thomas M. Apple, Ph.D.  
Dean of the College of Arts and Sciences

Approved:  
Conrado M. Gempesaw II, Ph.D.  
Vice Provost for Academic and International Programs
I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: ___________________________________
Tobin A. Driscoll, Ph.D.
Professor in charge of dissertation

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: ___________________________________
Bengt Fornberg, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: ___________________________________
Louis F. Rossi, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: ___________________________________
Gilberto Schleiniger, Ph.D.
Member of dissertation committee
"Hi Dr. Driscoll,

From all research topics that I heard last week, I think RBFs is the most interesting ... Could we meet again to talk about that? (today after 3:30 works for me) ..."

Email sent at 8:44 AM, October 21, 2002.

"That’s great. Let’s do Wed. after your class, say 3:30 or 3:45."

Email received at 1:17 PM.
This thesis is the result of about two and a half years of research on radial basis functions (RBFs). My first investigation of RBF methods was in connection with eigenvalue problems. There was a possibility of using RBFs in the reconstruction of ocean currents from measurements of surface velocities. One approach used for this reconstruction requires the computation of Dirichlet and Neumann modes of the Laplacian over a 2-D irregular domain; such as a bay. As a warm-up problem, we solved eigenvalue problems with corner singularities using RBFs expansions augmented by singular terms. The difficulty to obtain the type of accuracy we expected from a spectral method motivated us to explore least-squares techniques. Least squares improved the convergence in some cases, but the conditioning issue kept us from obtaining highly accurate approximations for more complex problems, such as the computation of the eigenmodes of the isospectral drums as in Driscoll [17]. This drove us to address more fundamental questions such as the source of ill-conditioning and how to avoid it. The simplest problem that could give us some insight on the behavior of RBFs was the interpolation of smooth functions in 1-D. Our goal at this point was to find a well conditioned basis for the subspace spanned by RBFs, which led us to an Arnoldi-like algorithm for Gaussians. We soon realized that Gaussians and polynomials were related. This simple connection was the starting point for what is presented in this thesis.
ACKNOWLEDGMENTS

I want to thank my wife Carina and my parents, Ronald and Niza. Their unconditional love and support were fundamental through this journey.

This dissertation would not have been possible without the guidance and encouragement of my advisor, Prof. Tobin Driscoll. He has had a profound impact in my career. Thank you for your advise and for believing in and supporting me throughout this endeavor.

I am deeply thankful to Prof. Gilberto Schleiniger for his support, advise and friendship. I also want to thank his family: Lilia, Daniel, Andre, and Laura, for their hospitality and friendship during our time in Delaware.

I am honored to have Prof. Bengt Fornberg in my dissertation committee. I have learned a lot from his book and articles on RBFs. I would also like to thank Prof. Fornberg and Dr. Grady Wright for fruitful discussions on some aspects of this work.

I am grateful to Prof. Louis Rossi, for being part of my dissertation committee, and to the faculty of the University of Delaware for their inspiring teaching and enthusiasm for mathematics.

I am thankful to colleagues and staff of the University of Delaware. Special thanks goes to Deb See for helping me in the administrative aspects of obtaining a Ph.D. and for her friendship.
I dedicate this dissertation to my wife Carina.
# TABLE OF CONTENTS

LIST OF FIGURES ................................................. xi  
LIST OF TABLES .................................................. xvii  
ABSTRACT ............................................................ xviii  

Chapter  
1 INTRODUCTION .................................................. 1  
   1.1 Spectral methods ............................................. 2  
   1.2 Radial basis functions ...................................... 4  
      1.2.1 Conditionally positive definite functions .............. 5  
      1.2.2 Interpolation by RBFs .................................. 6  
      1.2.3 RBF methods for PDEs .................................. 9  
   1.3 Thesis outline .............................................. 13  

2 POLYNOMIALS AND POTENTIAL THEORY FOR  
   GAUSSIAN RBF INTERPOLATION ............................. 17  
   2.1 Polynomial interpolation ................................... 17  
   2.2 An error estimate for Gaussian interpolation on native spaces .... 21  
   2.3 Gaussian RBFs as polynomials ............................ 23  
      2.3.1 The Runge phenomenon .................................. 27  
      2.3.2 Lebesgue constants ..................................... 28  
      2.3.3 Stable interpolation nodes ............................. 32  
      2.3.4 Location of Centers ................................... 37  
   2.4 Algorithmic implications ................................... 41  
   2.5 Differentiation Matrices ................................... 44
5.4.2 Eigenvalues as local maxima of the pseudoresolvent

5.4.2.1 Formulation
5.4.2.2 Numerical experiments
5.4.2.3 Polynomial and singular basis

6 CONCLUSION

6.1 Summary and discussion
6.2 Future work

Appendix

A REDUCING THE SIZE OF A RECTANGULAR PENCIL
B THE MINIMIZATION OF THE MINIMUM SINGULAR VALUE TO MACHINE PRECISION

BIBLIOGRAPHY
LIST OF FIGURES

1.1 The angle between two Gaussians decreases if: (a) the distance between centers is decreased; and (b) the shape parameter $c$ is increased. ......................................................... 7

1.2 Error and condition number of the interpolation matrix for the approximation of $f(x) = \sin(\pi x)$ for several values of $N$. Gaussian RBFs with equally spaced nodes and centers and $c = 1$ were used. ... 8

1.3 Numerical solution of a Poisson problem with homogeneous Dirichlet boundary condition on a peanut-like region. Multiquadrics with $c = 1$ and 200 nodes and centers were used to approximate the solution. .............................................................. 11

1.4 Error and nodes used in the computation of the solution in Figure 1.3. Two hundred nodes were used: 160 interior nodes ($\bullet$) and 40 boundary nodes ($\ast$). ..................................................... 11

2.1 Polynomial interpolation of $f(x) = 1/(1 + 16x^2)$ with $N = 16$: (a) on equally spaced nodes; (b) on Chebyshev nodes. .......................... 18

2.2 Equipotentials for (a) $\mu = 1/2$ and (b) $\mu(x) = 1/(\pi \sqrt{1 - x^2})$. Straight line represents the interval $[-1, 1]$ and dots mark the singularity locations of $f(z) = 1/(1 + 16z^2)$. ............................................. 20

2.3 Level curves of the logarithmic potential for $\beta = 1$ and $\mu(t) = 1/2$ (equispaced nodes). The straight line represents the interval $[-1, 1]$. ... 25

2.4 Contour maps of the logarithmic potential. Plots on the left were obtained with uniform node distribution. Plots on the right were obtained with Chebyshev distribution. ............................... 29
2.5 Interpolation of \( f \) with 25 equispaced nodes and \( \beta = 0.8 \). Closed curves are level curves of the logarithmic potential, dots mark the singularities of \( f \), and straight line represents the interval \([-1, 1]\). 30

2.6 Interpolation of \( f \) with 41 Chebyshev nodes and \( \beta = 2 \). Closed curves are level curves of the logarithmic potential, dots mark the singularities of \( f \), and straight line represents the interval \([-1, 1]\). 30

2.7 Lebesgue constant for different values of \( \beta \). Dashed lines mark the Lebesgue constant values for polynomial interpolation. 31

2.8 Numerical approximations of the optimal density functions for several values of \( \beta \). Dashed line shows the Chebyshev density function. 33

2.9 Node locations obtained using density function computed by solving the integral equation (2.14) for \( N = 20 \) and several values of \( \beta \). 34

2.10 Contour maps of the logarithmic potential obtained with numerically approximated optimal density function. 35

2.11 Lebesgue constant for different values of \( \beta \) and optimal node distribution. 36

2.12 Maximum error of the interpolation process using optimal nodes. Left: \( f(x) = 1/(1 + 25x^2) \) (●) and \( f(x) = 1/(4 + 25x^2) \) (*). Right: \( f(x) = 1/(x^2 - 1.8x + 0.82) \) (●) and \( f(x) = 1/(x^2 - 1.8x + 0.85) \) (*). Dashed lines mark convergence rates predicted by (2.8). 36

2.13 Equipotentials for \( \beta = 2 \) (compare with Fig. 2.4). Uniformly distributed centers on interval specified above. Interpolation points are uniformly distributed on \([-1, 1]\). 39

2.14 GRBF interpolation using Chebyshev points for centers and interpolation nodes, \( \beta = 2 \). 40

2.15 Maximum error of the interpolation of \( f(x) = 1/(1 + 25x^2) \) using barycentric interpolation (●) and the standard RBF algorithm (*). Left: \( \beta \) fixed. Right: \( c \) fixed. 43
2.16 Error in the approximation of $df/dx$, where $f(x) = 1/(1 + 25x^2)$, using the barycentric formulation for $D$ (●) and the standard RBF algorithm (*). .............................................. 45

3.1 Error in the least-squares approximation of $f(x) = 1/(1 + 16x^2)$ using monomials (*) and Chebyshev polynomials (○). ...................... 48

3.2 Number of least-squares nodes needed to approximate $f(x) = 1/(1 + 49x^2)$ to about the same accuracy obtained with interpolation on Chebyshev nodes: (*) equally spaced least-squares nodes; and (○) imperfectly clustered least-squares nodes. .............. 50

3.3 Number of Chebyshev least-squares nodes needed to approximate $f(x) = 1/(x^2 + 1.8x + 0.82)$ to about the same accuracy obtained with optimal interpolation nodes. Gaussians with equispaced centers and $\beta = 2$ were used. ................. 51

3.4 Error in the approximation of $f(x) = \tanh(10x)$ using polynomials (+) and using multiquadric RBFs with equally spaced centers (*) and imperfectly adapted centers (○). ................................. 52

3.5 Error of the least-squares approximation of $f(x) = \tanh(10x)$ for GRBFs with equally spaced centers and $\beta = 2$, using an orthogonal basis generated with an Arnoldi iteration (●) and the standard Gaussian radial basis (*). Polynomial approximation is also shown (+) ................................................................. 57

4.1 Eigenvalues of $D$ for GRBFs with equally spaced nodes in $[-1, 1]$: (a) $c = 1$; (b) $N = 9$. ................................................. 63

4.2 (a) Largest value of $c$ for which the spectrum of $D$ has non-positive real part — data obtained with multiquadrics with equally spaced centers and nodes. (b) The corresponding error in the interpolation of $\sin(4\pi x)$. (c) The corresponding condition number of the interpolation matrix. ........................................ 65

4.3 Numerical solution of the KDV equation — cf. [68, Output 27]. ... 69

4.4 Eigenvalues of $\tilde{D}$ on the unit circle with equally spaced centers and nodes: set of centers and nodes coincide (○); nodes are shifted 0.01 units from centers (+). ................................. 70

xiii
4.5 Numerical solution of a rigid-body rotation on the unit sphere computed with Gaussians. Nodes used in the approximations are shown at $t = 0$. The $L_\infty$ norm of the error at $t = 2\pi$ is 0.0075.

4.6 Eigenvalues of the GRBF differentiation matrix for $\beta = 2$ and stable nodes: N=30 (•); N=50 (○); and N=70 (*).

4.7 Spectral radius of the GRBF differentiation matrix as a function of $N$. For large values of $N$, the spectral radius grows as $O(N^2)$.

4.8 A 2-D test problem: (a) stable nodes; (b) Runge-Kutta stability region, scaled eigenvalues for stable nodes (•) and for equispaced nodes (×); (c) initial condition; (d) numerical solution at $t = 0.7$.

4.9 Spectra of differentiation matrices generated with multiquadrics and three sets of interpolation nodes: equally spaced (○); Chebyshev (+); and GRBF stable nodes (*).

4.10 Density functions for several values of $\xi$ for the Kosloff-Tal-Ezer modified Chebyshev method.

4.11 Eigenvalues of $D$ for multiquadric RBFs with equally spaced centers: (a) $c = 1$; (b) $N = 13$.

4.12 Error in the approximation $df/dx$, where $f(x) = \sin(\pi x)$, using an orthogonal basis generated with an Arnoldi iteration (•) and the standard Gaussian radial basis (*).

4.13 Eigenvalues of the least-square GRBF differentiation matrix for $\beta = 2$: N =30 (●); N =50 (○); and N =70 (*).

4.14 (a) Solution of the transport problem at $t = 1$ using $N = 20$ centers: exact solution (dotted line); least-squares method (solid line); and collocation method on GRBF nodes (dashed line). (b) maximum error for several values of $N$ obtained with the collocation method on GRBF nodes (○) and the least-squares method (*).

4.15 Eigenvalues of the Laplacian operator on the peanut region discretized with multiquadrics: (a) collocation method on uniformly spaced nodes; (b) the least-squares method.
4.16 Numerical solution of a vibrating peanut-shaped membrane using multiquadrics. Estimated $L_\infty$ norm of the error at $t = 1$ is 0.05.  

5.1 2-D regions considered for the numerical experiments. 

5.2 Numerically computed eigenfunction (left) and 177 nodes used for collocation (right). 

5.3 Relative error in selected eigenvalues of the disk. 

5.4 Boundary clustering. 

5.5 Example of distribution of centers (circles) and collocation nodes (dots) in a disk for collocation of the PDE on the boundary. 

5.6 Maximum relative error over the first 12 eigenvalues of the disk. 

5.7 Maximum relative error over the first 12 eigenvalues versus CPU time for the disk with Dirichlet boundary condition. 

5.8 Schematic picture of polar coordinates originating from a corner. 

5.9 Maximum relative error over the first 12 eigenvalues of the L-shaped region with and without corner terms augmentation. 

5.10 Contour plot of the 20th eigenfunction of the L-shaped region with Neumann boundary condition (20 contour lines). 

5.11 Error of the computed first eigenfunction of the rhombus with Dirichlet boundary condition, $N = 225$. 

5.12 Relative error of the first eigenvalue of the rhombus with and without corner terms augmentation (Dirichlet boundary condition). 

5.13 Eigenvalues of the disk computed using multiquadrics and $c = 3$. 

5.14 Eigenvalues of the disk computed using multiquadrics and $c = 1$. 

5.15 Surface of the pseudoresolvent norm for a rectangular pencil obtained from a discretization of the Laplacian in the disk. Dashed lines mark the location of true eigenvalues.
5.16 Plot of $\sigma_{\min}(zQ - A^T_iBR^{-1})$ for several real values of $z$ for the eigenvalues of the disk. Dashed lines represent the values of true eigenvalues.

5.17 Relative error in the 20 first eigenvalues of the disk computed using multiquadrics and $c = 3$ (cf. Figure 5.13).

5.18 Relative error in the 11 first eigenvalues of the disk computed using tensor product of polynomials.

5.19 Relative error in the first 3 eigenvalues of the L-shaped region. The basis functions are of the form $\{r^{n\varphi+2k}\sin(n\varphi\theta)\}$. xvi
LIST OF TABLES

1.1 Some commonly used RBFs. ............................................. 5

1.2 Error and condition number for several values of $N$ and $c$ in the solution of a boundary value problem. ..................... 12

4.1 Relative error ($\|u_{\text{approx}} - u_{\text{exact}}\|_{\infty} / \|u_{\text{exact}}\|_{\infty}$) in the computation of the solution of the KDV equation at $t = 6 \times 10^{-3}$. ................. 69
ABSTRACT

Radial basis function (RBF) approximations have been successfully used to solve partial differential equations in multidimensional complex domains. RBF methods are often called meshfree numerical schemes since they can be implemented without an underlying mesh or grid. We are particularly interested in the class of RBFs that allow exponential convergence for smooth problems. In the presence of rounding errors, stable and highly accurate approximations are often difficult even for simple geometries. In this thesis we explore this difficulty and possible remedies at theoretical and practical levels.

Theoretically, we explore a connection between Gaussian RBFs and polynomials. Using standard tools of potential theory, we find that these radial functions are susceptible to the Runge phenomenon, not only in the limit of increasingly flat functions, but also in the finite shape parameter case. We show that there exist interpolation node distributions that prevent such phenomena and allow stable approximations. This is believed to be the first proof that the stability of RBF interpolation depends on the location of the nodes. Using polynomials also provides an explicit interpolation formula that avoids the difficulties of inverting interpolation matrices, without imposing restrictions on the shape parameter or number of points.

We also show that node location plays an important role in the stability of RBF-based methods for time-dependent problems. Differentiation matrices obtained with infinitely smooth radial basis function (RBF) collocation methods have, under many conditions, eigenvalues with positive real part, preventing the use of
such methods for time-dependent problems. We prove that differentiation matrices for conditionally positive definite RBFs are stable for periodic domains. We also show that for Gaussian RBFs, special node distributions can achieve stability in 1-D and tensor-product non-periodic domains.

As a more practical approach, we investigate RBF least-squares approximations. We consider differentiation matrices based on least-squares discretizations and show that such schemes can lead to stable methods on less regular nodes. By separating centers and nodes, least-squares techniques open the possibility of the separation of accuracy and stability characteristics. An Arnoldi-like iterative algorithm is also presented to generate an orthogonal basis for the subspace spanned by Gaussian RBFs.
Chapter 1

INTRODUCTION

Numerical solutions of partial differential equations (PDEs) are routinely computed by researchers in many different areas and presently there are many numerical methods to choose from. Common choices are finite differences, finite elements, and spectral methods. These methods have been explored for decades and most of their strengths and weaknesses are known.

Radial basis functions (RBFs) are increasingly being used as an alternative to these traditional methods, especially for the solution of elliptic problems. RBF-based methods have several attractive features, most notably fast convergence (exponential for some cases) and the flexibility in the choice of node location; in fact, some implementations do not require an underlying grid or mesh. For this reason they are called meshfree numerical schemes.

In this thesis, we investigate approximations by global functions at theoretical and practical levels. Our main focus is on RBFs, but we also study other bases, including multivariate polynomials. Theoretically, we explore connections between a class of RBFs and polynomials and study to what extent node locations affect the accuracy and stability of computations. As a more practical approach, we study discrete least-squares approximations and methods for time-dependent and eigenvalue problems with corner singularities.

Our starting point is a global approximation of the form

$$F(x) = \sum_{k=0}^{N} \lambda_k \Phi_k(x). \quad (1.1)$$
The challenge is to choose the coefficients $\lambda_k$ and a family of functions $\Phi_k$ that make the error $\|F(x) - f(x)\|$ “small”, where $f$ is the target function. In particular, we are interested in methods for which the error decays exponentially with $N$, i.e., spectrally convergent methods.

### 1.1 Spectral methods

Spectral methods are based on polynomial or trigonometric expansions and are known for their exponential convergence – the computation of solutions accurate to 10 or 12 digits for smooth problems often requires little more computational effort than a solution accurate to 2 digits. The coefficients in (1.1) are usually obtained using tau, Galerkin or collocation methods [24]. In the latter scheme, the coefficients $\lambda_k$ are computed by requiring the residual to be zero at as many spatial points as possible. The collocation method is also referred to as the pseudospectral method [24].

The advent of the fast Fourier transform made these methods popular in the study of turbulent flows and meteorological simulations [54]. These numerical schemes, however, have limitations which have prevented them from being used in many applications where finite differences and finite elements are predominant. One of the main obstacles that spectral methods face is that the geometry of the problem must be simple, i.e., a cube, sphere, cylinder or the like. The reason for this restriction is the difficulty of computing an appropriate orthogonal basis in complex geometries. One approach to circumvent this limitation was presented by A.T. Patera in [53] where spectral elements were used. In this discretization, the domain is broken into a series of elements, and the solution is represented as a high-order Lagrangian interpolant through Chebyshev collocation points. These methods can also be seen as $hp$ finite elements since the accuracy of the method depends on the size of the elements ($h$) and on the degree of the polynomial basis ($p$) [1, 2, 43]. For sufficiently smooth solutions $p$-refinement usually leads to an exponential decay of
the numerical error, and from the spectral standpoint, the spectral element method can be viewed as a domain decomposition method.

At the heart of high-order elements based on polynomial interpolation, are special nodal sets. For rectangular regions, almost optimal nodes are obtained through tensor-product of the Jacobi-Gauss quadrature nodes. For other regions, the nodes are often computed by conforming, or almost conforming, structured discretization of the element [37]. For triangular and tetrahedral elements finding almost optimal nodes can also be related to solutions of minimum energy to a problem of electrostatics [35, 37]. This idea may be extended to other regions, but it is computationally expensive, as it requires a nonlinear minimization.

We also point out the work of Carpenter, Gottlieb, and Hesthaven [11, 33, 36], among others, who consider other ways of imposing boundary conditions. They observed that it is sufficient to impose the boundary conditions to the order of the scheme, i.e., weakly, such that the boundary condition is enforced exactly only asymptotically. In particular, they explore an approach known as the penalty method where a parameter is used to penalize errors at the boundary. This extra degree of freedom can be used to stabilize spectral methods on more complex regions, even if arbitrary nodes are used [11]. As mentioned in [36], however, in order to obtain spectral accuracy some restriction on the distribution of grid points must still be imposed.

Recently Bueno-Orovio et al. [9] presented a novel algorithm for problems with Neumann boundary conditions on arbitrary domains that does not require domain decomposition nor finite element discretizations. The region in consideration is embedded in a tensor product region and Fourier spectral methods are used to solve the problem in the larger region and a smooth cutoff function is used to restrict the solution to the desired domain. Notice that this scheme takes full advantage of the fast Fourier transform even though the function being approximated may not
be periodic. A negative aspect, however, is that accuracy is often compromised by the use of the cutoff function.

1.2 Radial basis functions

Is there a method that allows fast convergence rates, as the ones obtained with spectral methods, for problems with complex geometries without resorting to domain decomposition or finite elements? One method that has been proved successful in this regard is the method of particular solutions for the Laplacian eigenvalue problem [5]. For a more general class of problems, RBF methods have often been portrayed as multidimensional meshfree methods with exponential convergence rates [10, 40, 49, 50, 74, 78]. The literature on approximations by RBFs dates back to the early 70s with the work of R.L. Hardy [34] who used multiquadrics to approximate topographical data of scattered measurements. E.J. Kansa first used RBFs to solve PDEs in the early 90’s [40]. Much effort has been made to understand and apply RBFs to the solution of several types of PDEs since then [10, 12, 22, 26].

Given a set of centers $\xi_0, \ldots, \xi_N$ in $\mathbb{R}^d$, an RBF approximation takes the form

$$F(x) = \sum_{k=0}^{N} \lambda_k \phi(\|x - \xi_k\|),$$

(1.2)

where $\| \cdot \|$ denotes the Euclidean distance between two points and $\phi(r)$ is a function defined for $r \geq 0$. The coefficients $\lambda_0, \ldots, \lambda_N$ may be chosen by interpolation or other conditions at a set of nodes that typically coincide with the centers. In this work, however, we give special attention to the case that the location of centers and nodes differ, including the case that the number of nodes is larger than the number of centers.

Common choices for $\phi$ are presented in Table 1.1. They usually fall into two main categories: Infinitely smooth and containing a free parameter, and piecewise smooth and parameter-free. The shape parameter $c$ plays a key role on the convergence rate of the approximations and condition number of relevant matrices –
see, e.g., [12, 18, 60]. The fact that none of the functions presented in this table have finite support implies that the underlying matrices of RBF-based methods are usually full.

We point out that some properties of methods based on smooth RBFs differ significantly from those based on piecewise smooth functions; for instance, smooth RBF approximations converge exponentially for a certain class of functions, while piecewise smooth RBF approximations will converge at polynomial rates. The difficulties related to ill-conditioned matrices is also more predominant for smooth RBF-based algorithms than for piecewise smooth RBF-based schemes. Our focus in this work is on smooth RBFs. Other types of RBFs not investigated here include the compactly supported (piecewise smooth) RBFs [73] and a new class of oscillatory RBFs recently introduced in [27].

### 1.2.1 Conditionally positive definite functions

The concept of conditionally positive definite functions has been used to prove invertibility of RBF interpolation matrices. We have also found that this class of functions have special properties for certain time-dependent problems, as shall be discussed in more detail in Chapter 4.

A radial function \( \phi : \mathbb{R} \to \mathbb{C} \) is called conditionally positive definite of order \( m \) if for any set of distinct nodes \( x_0, x_1, ..., x_N \), and for all \( \lambda \in \mathbb{C}^{N+1} \setminus \{0\} \) satisfying

\[
\sum_{j=0}^{N} \lambda_j \phi(x_j) = 0,
\]  

\[ (1.3) \]
for all polynomials $p$ of degree less than $m$, the quadratic form
\[
\sum_{i=0}^{N} \sum_{j=0}^{N} \lambda_i \lambda_j \phi(\|x_i - x_j\|)
\] (1.4)
is positive [58, 64]. In this case, it is common to augment expansion (1.2) with a polynomial of degree at most $m - 1$ in order to impose (1.3).

Conditionally positive definite RBFs of order zero are called positive definite RBFs. Gaussians and inverse quadratics, for instance, are positive definite RBFs and multiquadrics are conditionally positive definite of order 1 [64].

1.2.2 Interpolation by RBFs

Assume that interpolation nodes and centers coincide, i.e., $x_j = \xi_j$. The augmented RBF expansion at these nodes takes the form
\[
F(x_i) = \sum_{j=0}^{N} \lambda_j \phi(\|x_i - x_j\|) + \sum_{k=0}^{N_p} \alpha_k p_k(x_i),
\] (1.5)
where \(\{p_0, p_1, ..., p_{N_p}\}\) is a basis for the space of $d$-variate polynomials of degree at most $m - 1$.

Given a function $f$, therefore, we can write (1.3) and (1.5) in matrix form,
\[
\begin{bmatrix}
A & P \\
P^T & 0
\end{bmatrix}
\begin{bmatrix}
\lambda \\
\alpha
\end{bmatrix}
= \begin{bmatrix}
F \\
0
\end{bmatrix},
\] (1.6)
where the elements of $A$ are $A_{i,j} = \phi(\|x_i - x_j\|)$, the elements of $P$ are $P_{i,j} = p_j(x_i)$, and $F_i = f(x_i)$.

For conditionally positive definite functions, the linear system above is guaranteed to have a unique solution as long as the only polynomial of degree at most $m - 1$ that vanishes at all nodes is the zero polynomial [64]. In practice, however, this matrix becomes ill-conditioned as the approximations get more accurate, to the extent that global interpolants are rarely computed for more than a couple of
Figure 1.1: The angle between two Gaussians decreases if: (a) the distance between centers is decreased; and (b) the shape parameter \( c \) is increased.

hundred nodes. This is one of the main drawbacks experienced by RBF implementations. Based on numerical and theoretical observations, in [62] Schaback states that for RBFs, “Either one goes for a small error and gets a bad sensitivity, or one wants a stable algorithm and has to take a comparably larger error”. Notice that for large values of the shape parameter \( c \), the smooth functions in Table 1.1 become flat, aggravating the ill-conditioning problem. Another source of ill-conditioning is the proximity of the centers, i.e., the closer the centers are to one another the smaller is the angle\(^1\) between the basis functions. This is illustrated in Figure 1.1 for Gaussians in 1-D. Notice that the angle between two Gaussians is given by

\[
\cos^{-1} \left( e^{-h^2/(2c^2)} \right),
\]

where \( h \) is the distance between the centers. In fact, if \( N \) is fixed, the condition number of the interpolation matrix is a function of one parameter, \( h/c \) [12]. Moreover, according to the uncertainty principle derived in [62], the condition number must grow exponentially with \( N \) for spectrally accurate RBF methods [10].

\(^1\) Here the angle between two functions \( f \) and \( g \) is defined by \( \cos^{-1} \left( \frac{(f,g)}{\|f\| \|g\|} \right) \).
Figure 1.2: Error and condition number of the interpolation matrix for the approximation of $f(x) = \sin(\pi x)$ for several values of $N$. Gaussian RBFs with equally spaced nodes and centers and $c = 1$ were used.

Figure 1.2 presents a typical convergence plot of an RBF approximation. The target function here is $f(x) = \sin(\pi x)$. We use Gaussian RBFs with equally spaced nodes and centers and $c = 1$. Notice that convergence is spectral for $N < 15$, but for larger values of $N$ rounding errors contaminate the approximations and accuracy becomes compromised. The dashed line illustrates how fast the interpolation matrix becomes ill-conditioned. The point where the error curve levels off can be changed by suitable choices of the shape parameter and researchers have suggested different strategies for choosing $c$ [12, 22, 41, 60].

We point out that the difficulty of working with ill-conditioned matrices can, in some cases, be overcome. Fornberg and Wright [29] recently presented a contour-integral approach that allows numerically stable computations of RBF interpolants for all values of the free parameter $c$. The main idea of this method is to consider the RBF interpolant (1.2) at a fixed $x$ as a function of $\varepsilon = 1/c$, not only for real values of $c$ but also complex values. In a neighborhood of $\varepsilon = 0$, (1.2) becomes a meromorphic
function of $\varepsilon$ and can therefore be written as a sum of a rational function of $\varepsilon$ and a power series in $\varepsilon$. The goal is then to determine the coefficients to the rational function and the power series using a contour-integral. The authors show that this is an effective tool to investigate properties of RBF approximations; they also point out in [29] that, at the present time, the technique is only recommended for small to moderate data sets. This approach demonstrates that the ill-conditioning is not intrinsic to RBF methods, but only associated to some algorithms.

In the limit $c \to \infty$ (increasingly flat RBFs) Driscoll and Fornberg [18] showed that in 1-D RBF interpolants usually converge to the Lagrange minimal-degree interpolating polynomial. They also pointed out that the limit may not exist in higher dimensions. This work has been extended in [30, 48]. In [30] Fornberg et al. conclude that if multivariate RBF interpolants converge for $c \to \infty$, they take the form of a multivariate polynomial. They also note that the existence of the limit, for most RBF, depends on the data point distribution. In [48] Larsson and Fornberg give explicit asymptotic expressions for the form of the multivariate interpolants in the limit of nearly flat RBFs. They also present a criterion in order for the limits to have a certain form, which seems to hold for all standard RBFs in use.

1.2.3 RBF methods for PDEs

One of the simplest ways of using RBFs to solve PDEs is through straight collocation [40, 47]. Consider, for instance, the following Poisson problem:

$$\Delta u(x) = f(x) \text{ in } \Omega \quad (1.7)$$

$$u(x) = g(x) \text{ on } \partial\Omega \quad (1.8)$$

In order to enforce the boundary condition, nodes along the boundary are needed. Assume that $x_0, ..., x_{N_I}$ are interior nodes, $x_{N_I+1}, ..., x_N$ are boundary nodes, and the
solution $u$ takes the form (1.2) with $\xi_j = x_j$. Enforcing (1.7) and (1.8) at interior and boundary nodes, respectively, gives the following algebraic system:

$$\sum_{k=0}^{N} \lambda_k \Delta \phi(\|x - x_k\|) \big|_{x=x_j} = f(x_j), \quad j = 0, ..., N_I$$

(1.9)

$$\sum_{k=0}^{N} \lambda_k \phi(\|x_j - x_k\|) = g(x_j), \quad j = N_I + 1, ..., N.$$  (1.10)

The coefficients $\lambda_k$ can then be found (in most cases) by solving this $(N+1) \times (N+1)$ system.

Figure 1.3 presents a numerical solution of $\Delta u(x) = -2$ with homogeneous Dirichlet boundary condition. Multiquadrics with $c = 1$ and 200 nodes were used to compute this solution. A finer grid was used to generate the graph. The nodes used in the approximation are shown in Figure 1.4. The maximum error in this approximation is about 0.012. The estimated error in the approximation is also plotted in this figure. In order to compute the error a solution on a larger set of nodes was computed and subtracted from the numerical approximation obtained in Figure 1.3. Notice that the error is larger near the boundary. The inaccuracy of RBF approximations close to the edges has been observed and investigated by Fornberg et al. [25] and Fedoseyev et al. [23], among others. We shall give special attention to this issue in this thesis.

Table 1.2 presents the error for the same boundary value problem for other values of $N$ and shape parameter $c$. The error was computed using the maximum norm on 3000 nodes uniformly distributed. The condition number of the matrix used to solve the system (1.9)-(1.10) is also shown. Notice that the error decays and the condition number grows with $N$ as expected. For $N = 200$, among the values of $c$ used, $c = 1$ gives the most accurate solution. The optimal value of $c$ depends on the function being approximated [29, 60]. The data also shows that this algorithm is not suitable for computations with large $c$ as the system becomes severely ill-conditioned for $c > 2$.  

10
Figure 1.3: Numerical solution of a Poisson problem with homogeneous Dirichlet boundary condition on a peanut-like region. Multiquadrics with $c = 1$ and 200 nodes and centers were used to approximate the solution.

Figure 1.4: Error and nodes used in the computation of the solution in Figure 1.3. Two hundred nodes were used: 160 interior nodes (●) and 40 boundary nodes (⋆).
Fasshauer [21], among others, pointed out that the matrix resulting from the linear system (1.9)-(1.10) is not symmetric and that there are no guarantees that it is nonsingular even if the radial function is positive definite. In order to guarantee a nonsingular system, a variation of this method has been introduced [21, 77]. The idea is to augment the expansion (1.2) with terms of the form $\Delta \phi (\|x-x_k\|)$, for $x_k \in \partial \Omega$, which restores the symmetry of the system when boundary conditions are imposed (see [21] for details). This method, however, has not become popular, as practical experience shows that straight collocation works well and that the real difficulty lies on the bad conditioning of the system, which is not significantly improved by symmetric collocation.

In this work, we shall also use the idea of differentiation matrices, which are commonly used in pseudospectral methods [24, 68]. The goal is to derive a finite dimensional representation of a linear differential operator $\mathcal{L}$, like the Laplacian or a convection operator. Let $A$ and $A^L$ be matrices with elements

$$A_{i,j} = \phi (\|x_j - \xi_i\|) \quad \text{and} \quad A^L_{i,j} = \mathcal{L} \phi (\|x - \xi_i\|) \big|_{x=x_j}.$$

We can then write $u = A \lambda$ and $u_\mathcal{L} = A^L \lambda$, where $u$ and $u_\mathcal{L}$ are vectors containing the RBF approximations of the values of $u(x)$ and $\mathcal{L}u(x)$ at the collocation nodes $x = x_j$. This gives $u_\mathcal{L} = A^L A^{-1} u$ and $L = A^L A^{-1}$ is the finite-dimensional representation of

<table>
<thead>
<tr>
<th>$N$</th>
<th>Estimated Error ($L_\infty$)</th>
<th>Condition Number</th>
<th>$c$</th>
<th>Estimated Error ($L_\infty$)</th>
<th>Condition Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td>0.106</td>
<td>$2.9 \times 10^4$</td>
<td>0.1</td>
<td>0.070</td>
<td>$1.1 \times 10^4$</td>
</tr>
<tr>
<td>117</td>
<td>0.043</td>
<td>$2.2 \times 10^8$</td>
<td>0.5</td>
<td>0.020</td>
<td>$7.6 \times 10^6$</td>
</tr>
<tr>
<td>200</td>
<td>0.012</td>
<td>$1.3 \times 10^{10}$</td>
<td>1</td>
<td>0.012</td>
<td>$1.3 \times 10^{10}$</td>
</tr>
<tr>
<td>244</td>
<td>0.005</td>
<td>$2.5 \times 10^{10}$</td>
<td>1.2</td>
<td>0.020</td>
<td>$2.2 \times 10^{11}$</td>
</tr>
</tbody>
</table>

Table 1.2: Error and condition number for several values of $N$ and $c$ in the solution of a boundary value problem.
Boundary conditions can then be imposed by suitable modifications of the matrix $L$. Notice that the matrix $A$ is guaranteed to be invertible for most RBFs, and for conditionally positive definite RBFs the formulation can be modified according to (1.6). This formulation is particularly simple to apply to time-dependent nonlinear PDEs.

1.3 Thesis outline

In this dissertation we study global methods based on radial basis functions. Understanding how well RBF interpolation works is a crucial step to comprehend how it can be used in the numerical solution of PDEs. Convergence analysis of RBF interpolation has been carried out by several researchers – see, e.g. [49, 50, 74]. For smooth $\phi$, spectral convergence has been proved for functions belonging to a certain reproducing kernel Hilbert space $\mathcal{F}_\phi$ [50]. This space, however, is rather small since the Fourier transform of functions in $\mathcal{F}_\phi$ must decay very fast or have compact support [74]. More recently, in [78] Yoon obtained spectral orders on Sobolev spaces, and in [26] error analysis was performed by considering the simplified case of equispaced periodic data. The theory of RBF interpolation, however, remains underdeveloped compared to polynomial approximations.

In Chapter 2, we use standard tools of polynomial interpolation and potential theory to study several properties of Gaussian RBF (GRBF) interpolation in 1-D, including convergence and stability. We explore the fact that Gaussians with equally spaced centers are related to polynomials through a simple change of variable. Using this connection, we demonstrate a Runge phenomenon using GRBFs on equispaced and classical Chebyshev nodes, and we compute asymptotically optimal node densities using potential theory. Numerical calculations suggest that these node densities give Lebesgue constants that grow at logarithmic rates, allowing stable approximations. We also explore the algorithmic implications of the connections we have made and derive a barycentric interpolation formula that circumvents the
difficulty of inverting a poorly conditioned matrix, so approximations can be carried out to machine precision without restrictions on the values of the shape-parameter \( c \) and number of centers \( N \). Some observations on multiquadrics and other possible extensions of the methods are also presented. Most of the results presented in Chapter 2 will also appear in [57].

The results presented in Chapter 2 suggest that accuracy of smooth RBF interpolation depends strongly on node location, to the extent the approximations may not converge due to rounding errors if the nodes are not placed appropriately. In Chapter 3 we turn to least squares approximations to avoid this limitation. Since this approach is also applicable to polynomial approximations, we include them in our study. Although least squares gives certain flexibility on node locations and opens the possibility of separation of accuracy and stability for RBFs, the issue of finding well-conditioned representations of a basis (whether for RBFs or polynomials in high dimensions) remain. We investigate Arnoldi-like iterative algorithms as methods of orthogonalization. Experimental results on the convergence of these schemes are given.

Regarding RBF methods for time-dependent problems, in Chapter 4 we study the eigenvalue stability of such discretizations. We focus on the method of lines. In periodic regions like the unit circle and the unit sphere, we prove that RBF methods are time-stable for all conditionally positive definite RBFs and node distributions. However, in non-periodic domains experience suggests that RBFs will produce discretizations that are unstable in time unless highly dissipative time stepping is used. This difficulty can be connected to results obtained in Chapter 2, where we show that arbitrary node distributions may lead to unstable interpolations. For Gaussians in 1-D or tensor product regions, one way to stabilize RBF-approximations for time-dependent problems is to use the special nodes to be presented in Chapter 2 to generate differentiation matrices. A more flexible alternative is the use
of least-squares techniques. We explore a discrete least-squares method that has
the simplicity of collocation for nonlinearities and the like, yet allows stable ex-
licit time integration. Part of our investigation on the eigenvalue stability of RBF
discretizations for time-dependent problem will appear in [56].

In Chapter 5 we investigate RBF discretizations for eigenvalue problems.
Numerical results are presented for the computation of eigenmodes of the Lapla-
cian. Since RBF approximations are relatively inaccurate near the boundaries, we
study the boundary clustering of nodes in 2-D regions and the collocation of the
PDE on the boundary. We verify that these techniques are effective for preventing
degradation of the solution near the edge of the domain, although this is a purely
experimental approach.

For eigenvalue problems, we also consider the effect of corner singularities in
2-D regions, i.e., corners with interior angle $\pi/\varphi$ where $\varphi$ is not an integer. It is
known that some eigenfunctions are not smooth at such corners, and as a result a
typical RBF approximation gives very poor results in this case. In fact, our tests
show that the numerical solution does not converge if the method is not modified.
In [17] Driscoll exploits the singular behavior of an eigenfunction at the corners and
propose an efficient algorithm to compute eigenvalues for polygonal regions. With
that as motivation, we append terms to the RBF expansion that approximate the
singular behavior of an eigenfunction.

Least-squares approximations are also used to compute eigenmodes of the
Laplacian. We use matrices that are discretizations of the Laplacian. We investigate
how accurately the spectra of these matrices approximate the eigenvalues of the
continuous operator. We also draw a connection between least-squares solutions of
eigenvalue problems and the computation of pseudospectra of a rectangular pencil
[76]. Although using pseudospectra is computationally expensive if a large number
of eigenmodes is desired, for the approximation of a small number of eigenvalues
the technique is robust and requires less additional computational resources than other approaches. Part of our results on the computation of eigenmodes of elliptic operators using RBFs has been published in [55].
Chapter 2

POLYNOMIALS AND POTENTIAL THEORY FOR
GAUSSIAN RBF INTERPOLATION

The theory of approximation of functions by polynomials is rich and can be found in many classical books, [15, 45, 61] to mention but a few. It gives the foundation for spectral and pseudospectral methods. In the first two sections of this chapter, a brief review of classical statements about the convergence of polynomial and Gaussian interpolation is given. Associations between polynomial and GRBF interpolants have been made before, e.g. [18, 30], but the results are usually related with the asymptotic behavior of smooth RBFs as $c \to \infty$. In Section 2.3 we show that finite-parameter GRBFs are also related to polynomials. This allows us to draw specific conclusions about the convergence of GRBF approximations. We focus primarily on the dependence of accuracy and stability on node locations. We use potential theory to establish necessary conditions on the singularities of the target function in the complex plane for Gaussian interpolation in 1-D with equally spaced nodes. We also provide an explicit formula for the interpolants that avoids the difficulties of inverting ill-conditioned matrices. This formulation is also used to generate the entries of differentiation matrices.

2.1 Polynomial interpolation

A problem known as the Runge phenomenon is often used to illustrate why polynomial interpolation does not work well on equally spaced nodes [7, 24, 68].
Figure 2.1: Polynomial interpolation of \( f(x) = 1/(1 + 16x^2) \) with \( N = 16 \): (a) on equally spaced nodes; (b) on Chebyshev nodes.

Figure 2.1(a) shows the polynomial interpolant of \( f(x) = 1/(1 + 16x^2) \) on equally spaced nodes on \([-1, 1]\). The fact that the amplitude of the oscillations near the boundary grows exponentially is known as the Runge phenomenon. The remedy is to cluster the nodes more densely near the boundaries. Figure 2.1(b) presents the interpolant of the same function on Chebyshev nodes – nodes given by \( x_j = \cos(j\pi/N), j = 0, ..., N \) \([24, 68]\).

Chebyshev nodes work well because they are distributed according to the node density function \( \mu(x) = 1/(\pi\sqrt{1-x^2}) \). Node density functions describe how the density of node distribution varies over \([-1, 1]\) as \( N \to \infty \) \([24, 45]\). Assume that \( \int_{-1}^{1} \mu(x)dx = 1 \), the number of nodes in the interval \([-1, a]\), \( a \leq 1 \), is approximately

\[
N \int_{-1}^{a} \mu(x)dx.
\]

We point out that the density \( \mu(x) = 1/(\pi\sqrt{1-x^2}) \) is common to all node distributions that are roots of Jacobi polynomials \([24]\). In particular, Chebyshev nodes
satisfy the equation
\[ \frac{j}{N} = \frac{1}{\pi} \int_{-1}^{x_j} \frac{1}{\sqrt{1-x^2}} \, dx, \quad j = 0, \ldots, N. \]

For finite \( N \), \( \mu \) can be a sum of Dirac delta functions of amplitude \( N^{-1} \), but in the limit, we take it to be smooth. For equispaced nodes, the limit gives the uniform density function \( \mu(x) = 1/2 \).

The following theorem was taken from [24]; similar statements can be found in [7, 45, 68]. It gives necessary and sufficient conditions on the singularities of \( f \) for convergence in the complex plane \( z = x + iy \).

**Theorem 2.1** Given a node density function \( \mu(x) \) on \([-1,1]\), we form the potential function
\[ U(z) = -\int_{-1}^{1} \ln |z - t| \mu(t) \, dt \tag{2.1} \]
Then:

- **Polynomial interpolation converges to** \( f(z) \) **inside the largest equipotential curve for** \( U(z) \) **that does not enclose any singularity of** \( f(z) \), **and diverges outside that curve.**

- **The rate of convergence/divergence is exponential.**

- **Pseudospectral approximation to any derivative converges in the same fashion as the interpolant does to the function.**

Notice that the function \( U(z) \) can be seen as the potential at \( z \) due to charges at \( \{x_j\} \), each with potential \( N^{-1} \log \|z - x_j\| \), in the limit \( N \to \infty \) [68]. Finding optimal nodes for polynomial interpolation, therefore, can be associated with the problem of finding the position of the charges that minimizes the electrostatic energy. According to [35], this remarkable connection between Jacobi polynomials and electrostatics was first revealed by Stieltjes [65, 66] in 1885.
Figure 2.2: Equipotentials for (a) $\mu = 1/2$ and (b) $\mu(x) = 1/(\pi \sqrt{1-x^2})$. Straight line represents the interval $[-1, 1]$ and dots mark the singularity locations of $f(z) = 1/(1 + 16z^2)$.

Figure 2.2 shows equipotentials for $\mu = 1/2$ (uniform distribution) and $\mu(x) = 1/(\pi \sqrt{1-x^2})$. Notice that the latter gives optimal equipotentials, as the interval $[-1, 1]$ is itself an equipotential [24]. This figure also shows the singularities of $f(z) = 1/(1 + 16z^2)$. For uniformly distributed nodes, these locations fall inside a region enclosed by an equipotential that intersects the interval $[-1, 1]$, completely explaining the left half of Figure 2.2. For $\mu(x) = 1/(\pi \sqrt{1-x^2})$, on the other hand, it is enough that the singularities are located outside the interval $[-1, 1]$ itself, leading to well-behaved interpolants.

Smooth RBFs contain a free parameter $c$ that defines the shape of the basis function – nearly flat for large $c$ and sharply peaked for small $c$. Driscoll and Fornberg proved in [18] that in the limit $c \to \infty$, RBF interpolation is equivalent to polynomial interpolation on the same nodes. This result has been extended to higher dimensions in [30, 48, 63]. In this limit, therefore, the classical Runge phenomenon, and its remedy through node spacing, apply.
For practical RBF implementations it is well appreciated that node clustering near the boundaries is helpful [25, 47], but to our knowledge there has been no clear statement about the Runge phenomenon or asymptotically stable interpolation nodes for finite-parameter RBFs. The question has perhaps been obscured somewhat by the fact that the straightforward approach to computing the \( \lambda_k \) is itself numerically ill-conditioned when the underlying approximations are accurate [62].

2.2 An error estimate for Gaussian interpolation on native spaces

In this section we present error estimates for Gaussian interpolation obtained by Madych and Nelson in [50]. The results derived in their article are also valid for other smooth RBFs; for simplicity however, we restrict our discussion to Gaussians. According to [74], this was the first paper that established spectral convergence for RBF interpolation. We start by defining the reproducing kernel Hilbert space (or native space) \( \mathcal{F}_\phi \) for a positive definite RBF \( \phi \in L_1(\mathbb{R}^d) \):

\[
\mathcal{F}_\phi = \{ f \in L_2(\mathbb{R}^d) : \hat{f}/\sqrt{\hat{\phi}} \in L_2(\mathbb{R}^d) \}.
\]

Here, \( \hat{f} \) and \( \hat{\phi} \) are the Fourier transforms\(^1\) of \( f \) and \( \phi \), respectively. We also define the native space norm

\[
\|f\|_\phi = \|\hat{f}/\sqrt{\hat{\phi}}\|_{L_2(\mathbb{R}^d)}.
\]

The following theorem was taken from [50]. It gives sufficient conditions for the spectral convergence of Gaussian interpolation. We assume that centers and nodes coincide, \( x_j = \xi_j \), and define

\[
h_\Omega = \sup_{x \in \Omega} \min_{j \in \{0,\ldots,N\}} \|x - x_j\|.
\]

Notice that \( h_\Omega \) is an estimate of how closely the nodes \( x_j \) cover \( \Omega \).

\(^1\) Here the Fourier transform of \( f \) is defined as \( \hat{f} = \int_{\mathbb{R}^d} f(x) \exp(-i\omega \cdot x) dx \).
Theorem 2.2 (Madych and Nelson) If $\phi(r) = e^{-r^2}$, $\Omega = [-1,1]^d$, then there exist $\epsilon \in (0,1)$ such that

$$|F(x) - f(x)| \leq \epsilon^{1/h}\|f\|_\phi$$

for all $x \in \Omega$ and $f \in \mathcal{F}_\phi$, as $h \rightarrow 0$.

As mentioned in [74], a close look at the proof of this theorem suggests that $\epsilon$ depends heavily on the space dimension and that it approaches 1 as $d$ increases. In [74] Wendland gives an improved version of this theorem, with constants that are independent of the space dimension, but with $\sqrt{h}$ instead of $h$ in the exponent of $\epsilon$.

Notice that for Gaussians the native space $\mathcal{F}_\phi$ is rather small, since the Fourier transform of a Gaussian is also a Gaussian, and if $f$ is in $\mathcal{F}_\phi$, $\hat{f}$ must decay faster than $\hat{\phi}$. This argument is often used to diminish the importance of native spaces [74, 78]. To illustrate this point, consider functions of the form $f(x) = 1/(1 + (ax)^2)$ defined on the real line. For $\phi(r) = e^{-r^2/c^2}$ we have that

$$\frac{\hat{f}}{\sqrt{\hat{\phi}}} = \frac{\pi^{3/4}}{a\sqrt{c}} \exp\left(\frac{\omega^2c^2}{8} - \frac{\omega}{a}\right).$$

Hence, even for small $a$ and $c$, smooth functions of the form $1/(1 + (ax)^2)$ are not in $\mathcal{F}_\phi$. In [78], Yoon extends the work of Madych and Nelson and obtain error estimates on Sobolev spaces. We point out, however, that this article does not establish spectral rates of convergence for Gaussian RBFs and that some assumptions needed to prove the theorems prohibit their application in several practical situations.

Besides these restrictive assumptions used to obtain error estimates, the limitation of the results presented in those articles is that they do not address the role of node distributions nor consider the effect of roundoff errors on the approximations. For GRBFs with equally spaced centers in 1-D, however, we can make specific statements about the convergence of the interpolants in the presence of rounding
errors for a given node distribution. In the section that follows, we shall explore this specific case. To our knowledge this is the first result that addresses the role of node locations on the stability of the interpolation process.

2.3 Gaussian RBFs as polynomials

In (1.2) we now choose \( d = 1 \), Gaussian shape functions, and centers \( \xi_k = -1 + 2k/N = -1 + kh \), \( k = 0, \ldots, N \). Hence the RBF approximation is

\[
F(x) = \sum_{k=0}^{N} \lambda_k e^{-(x+1-kh)^2/c^2} = e^{-(x+1)^2/c^2} \sum_{k=0}^{N} \lambda_k e^{(2kh-k^2h^2)/c^2} e^{2kh/c^2}.
\] (2.2)

Making the definition \( \beta = 2h/c^2 = 4/(Nc^2) \) and using the transformation

\[
s = e^{\beta x}, \quad s \in [e^{-\beta}, e^{\beta}],
\]

we find that

\[
G(s) = F(\log(s)/\beta) = e^{-N(\log s+\beta)^2} \sum_{k=0}^{N} \tilde{\lambda}_k s^k = \psi^N_{\beta}(s) \sum_{k=0}^{N} \tilde{\lambda}_k s^k,
\] (2.3)

where the \( \tilde{\lambda}_k \) are independent of \( s \). In this section we regard \( \beta \) as a fixed parameter of the GRBF method. In the literature this is sometimes called the stationary case [8].

From (2.3) it is clear that \( G/\psi^N_{\beta} \) is a polynomial of degree no greater than \( N \). If \( F \) is chosen by interpolation to a given \( f \) at \( N + 1 \) nodes, then we can apply standard potential theory to find necessary convergence conditions on the singularities of \( f \) in the complex plane \( z = x + iy \).

**Lemma 2.1** Suppose \( f \) is analytic in a closed simply connected region \( R \) that lies inside the strip \( -\pi/(2\beta) < \text{Im}(z) < \pi/(2\beta) \) and \( C \) is a simple, closed, rectifiable curve that lies in \( R \) and contains the interpolation points \( x_0, x_1, \ldots, x_N \). Then the remainder of the GRBF interpolation for \( f \) at \( x \) can be represented as the contour integral

\[
f(x) - F(x) = \frac{\beta \eta_N(x)}{2\pi i} \int_C \frac{f(z)e^{\beta z}}{\eta_N(z)(e^{\beta z} - e^{\beta x})} dz
\]
where \( \eta_N(x) = e^{-\frac{N\beta}{2}(x+1)^2} \prod_{k=0}^{N} (e^{\beta x} - e^{\beta x_k}). \)

**Proof.** Consider the conformal map \( w = e^{\beta z} \) and let \( g(s) = f(\log(s)/\beta) \). Under this transformation, the region \( R \) is mapped to a closed simply connected region that lies in the half-plane \( \text{Re}(w) > 0 \). Thus \( g/\psi_N^\beta \) is analytic in this region in the \( w \)-plane and we can use the Hermite formula for the error in polynomial interpolation [15],

\[
g(s) - G(s) = \psi_N^\beta(s) \left( \frac{g(s)}{\psi_N^\beta(s)} - \sum_{k=0}^{N} \tilde{\lambda}_k s^k \right) = \psi_N^\beta(s) \prod_{k=0}^{N} (s - s_k) \int e^{(w - s)\psi_N^\beta(w)} \prod_{k=0}^{N} (w - s_k) dw,
\]

where \( s_k = e^{\beta x_k} \) and \( C \) is the image of \( C \) in the \( w \)-plane. A change of variables completes the proof. \( \square \)

We now turn our attention to necessary conditions for uniform convergence of the interpolation process. To this end, we shall use the concept of limiting node density functions introduced in Section 2.1. Since our analysis parallels with the convergence proof for polynomial interpolation (see, e.g., [15, 45, 72]), define

\[
U_\beta(z) = \frac{\beta}{4} \text{Re} \left[ (z + 1)^2 \right] - \int_{-1}^{1} \log(|e^{\beta s} - e^{\beta t}|) \mu(t) dt. \tag{2.4}
\]

We shall refer to this function as the *logarithmic potential* and its level curves *equipotentials*.

In the theorem below we shall assume that \( \mu \) is such that there exist \( a \) and \( b, \ a < b \), with the property that if \( K \in [a, b] \) then there exists a simple, closed, rectifiable curve that satisfies \( U_\beta(z) = K \) and contains the interval \([-1, 1]\) in its interior. We denote this curve by \( C_K \) and by \( R_K \) the part of the plane which lies inside it. We also require that if \( K_1 > K_2 \) then \( R_{K_1} \subset R_{K_2} \). To illustrate this feature, consider the logarithmic potential for uniformly distributed nodes on
Figure 2.3: Level curves of the logarithmic potential for $\beta = 1$ and $\mu(t) = 1/2$ (equispaced nodes). The straight line represents the interval $[-1, 1]$.

$[-1, 1]$ and $\beta = 1$. In this case we have that $\mu(t) = 1/2$. The level curves of $u_1$ are presented in Figure 2.3. In this instance one could choose $a = 0.5$ and $b = 0.7$.

**Theorem 2.3** Suppose $\mu$ satisfies the properties above and let $B$ be the closure of $R_b$. If $f$ is an analytic function in an open region $R$ which lies inside the strip $-\pi/(2\beta) < \text{Im}(z) < \pi/(2\beta)$ and contains $B$ in its interior, then the GRBF interpolation described above converges uniformly with respect to $z \in B$.

**Proof.** Since $R$ is open and $B$ is closed, there exist $K_1$ and $K_2$, such that $K_1 < K_2 < b$ and $R_{K_1} \cup C_{K_1}$ lies inside $R$. Using Lemma 2.1 we have that for any $x$ on $C_{K_2}$,

$$|f(x) - F(x)| \leq \frac{\beta M}{2\pi \delta} \int_{C_{K_1}} \frac{|\eta_N(x)|}{|\eta_N(z)|} dz,$$

(2.5)

where $M$ is the largest value of $|f(z)e^{\beta z}|$ on $C_{K_1}$ and $\delta$ is the smallest value of $|e^{\beta z} - e^{\beta x}|$ for $z \in C_{K_1}$ and $x \in C_{K_2}$.
We also have that
\[
\left| \frac{\eta_N(x)}{\eta_N(z)} \right| = \exp \left\{ -N \left( \log |\eta_N(z)|^{\frac{1}{N}} - \log |\eta_N(x)|^{\frac{1}{N}} \right) \right\}. \tag{2.6}
\]
A bound on this exponential can be obtained using the limiting logarithmic potential. Notice that
\[
\lim_{N \to \infty} \log |\eta_N(z)|^{\frac{1}{N}} = -U_\beta(z) = -K_1 \quad \text{for } z \in C_{K_1}
\]
and
\[
\lim_{N \to \infty} \log |\eta_N(x)|^{\frac{1}{N}} = -U_\beta(x) = -K_2 \quad \text{for } x \in C_{K_2}.
\]
Hence, for any given $\epsilon$, $0 < \epsilon < (K_2 - K_1)/2$, there exists $N_\epsilon$ such that for $N > N_\epsilon$,
\[
-K_1 - \epsilon < \log |\eta_N(z)|^{\frac{1}{N}} < -K_1 + \epsilon
\]
\[
-K_2 - \epsilon < \log |\eta_N(x)|^{\frac{1}{N}} < -K_2 + \epsilon,
\]
which implies that
\[
\log |\eta_N(z)|^{\frac{1}{N}} - \log |\eta_N(x)|^{\frac{1}{N}} < m_\epsilon, \tag{2.7}
\]
where $m_\epsilon = K_2 - K_1 + 2\epsilon > 0$.

Combining (2.5), (2.6), and (2.7) gives
\[
|f(x) - F(x)| \leq \frac{\beta M_\kappa}{2\pi \delta} e^{-Nm_\epsilon}, \quad N > N_\epsilon, \quad x \in C_{K_2} \tag{2.8}
\]
where $\kappa$ is the length of $C_{K_1}$.

This last inequality implies that $|f - F| \to 0$ uniformly as $N \to \infty$ on $C_{K_2}$. Since $f - F$ is analytic in $R_{K_2}$, by the maximum modulus principle we have that $F$ converges uniformly to $f$ in $R_{K_2}$. \Box

We point out that, as happens in polynomial interpolation, the convergence in (2.8) is exponential with a rate that is governed by the equipotentials induced by the nodes.
2.3.1 The Runge phenomenon

As discussed in Section 2.1, the Runge phenomenon is well understood in polynomial interpolation in 1-D. Even if a function is smooth on the interpolation interval $[-1, 1]$, polynomial interpolants on equispaced nodes will not converge to it uniformly as $N \to \infty$ unless the function is analytic in a larger complex region. Clustering nodes more densely near the ends of the interval avoids this difficulty. Specifically, for points distributed with density $\pi^{-1}(1-x^2)^{-1/2}$, such as Chebyshev extreme points $x_j = -\cos(j\pi/N)$ and zeros of Chebyshev and Legendre polynomials, uniform convergence is guaranteed as long as the function being interpolated is analytic inside an ellipse with foci $\pm 1$ and semiminor larger than $\delta$, for some $\delta > 0$ [24].

In this section we show that for GRBFs uniform convergence may be lost, not only in the polynomial limit $c \to \infty$ but also for constant $\beta$ (which implies $c \to 0$ as $N \to \infty$), if the distribution of interpolation nodes is not chosen appropriately. Theorem 2.3 can be used to state the regularity requirements of the function being interpolated using a given node distribution and enables us to determine whether the interpolation process is convergent.

We point out that for $\beta \ll 1$,

$$U_{\beta}(z) = -\log(\beta) - \int_{-1}^{1} \log |z - t| \mu(t) dt + O(\beta).$$

(2.9)

In this case, the level curves of $U_{\beta}$ are similar to equipotentials of polynomial interpolation and the convergence of the GRBF interpolation process can be predicted from the well-known behavior of polynomial interpolation.

Equipotentials for $\beta = 0.1, 0.8, 2, 5$ are presented in Figure 2.4. On the left of this figure, we present contour maps obtained with a uniform node distribution, and on the right, contour maps obtained with the Chebyshev extreme points. Equipotentials for $\beta = 0.1$ are similar to equipotentials for polynomial interpolation (cf. Figure 2.2), as expected. By Theorem 2.3, convergence is guaranteed if the
function is analytic inside the contour line that surrounds the smallest equipotential domain that includes \([-1, 1]\), whereas any singularity inside this region leads to spurious oscillations that usually grow exponentially. Therefore, it is desirable to have the region where the function is required to be analytic as small as possible. In this sense, we note that for \(\beta = 0.1\), Chebyshev distribution is close to optimal, and for \(\beta = 5\), uniform distribution seems to be more appropriate. We also note that, for large \(\beta\), Chebyshev density over-clusters the nodes near the ends of the interval. In fact, if this clustering is used with \(\beta = 5\), even the interpolation of \(f \equiv 1\) is unstable; in this case there is no equipotential region that encloses \([-1, 1]\).

To demonstrate how the equipotentials and singularities of the interpolated function restrict the convergence of GRBF interpolation, in Figures 2.5 and 2.6 we show two pairs of interpolants. Each pair consists of one function that leads to the Runge phenomenon and one that leads to a stable interpolation process. In Figure 2.5, equispaced nodes were used. The interpolation of \(f(x) = 1/(4 + 25x^2)\) is convergent, while the interpolation of \(f(x) = 1/(1 + 25x^2)\) is not. Notice from Fig. 2.4 that the former function is singular at points inside the smallest equipotential domain and the singularities of the latter function lie outside this region. For Chebyshev nodes and \(\beta = 2\), interpolation of \(f(x) = 1/(x^2 - 1.8x + 0.82)\) generates spurious oscillation in the center of the interval. Interpolation of a slightly different function, \(f(x) = 1/(x^2 - 1.8x + 0.85)\), gives a well-behaved interpolant.

### 2.3.2 Lebesgue constants

Although Theorem 2.3 guarantees convergence for sufficiently smooth functions and properly chosen interpolation points, approximations may not converge in the presence of rounding errors due to the rapid growth of the Lebesgue constant. For GRBF interpolation, we define the Lebesgue constant by

\[
\Lambda_N^{GRBF} = \max_{x \in [-1, 1]} \sum_{k=0}^{N} |L_k(x)| ,
\]  

(2.10)
\[ \mu(t) = 1/2 \quad \mu(t) = 1/\pi \sqrt{1-t^2} \]

\[ \beta = 0.1 \]

\[ \beta = 0.8 \]

\[ \beta = 2 \]

\[ \beta = 5 \]

Figure 2.4: Contour maps of the logarithmic potential. Plots on the left were obtained with uniform node distribution. Plots on the right were obtained with Chebyshev distribution.
Figure 2.5: Interpolation of $f$ with 25 equispaced nodes and $\beta = 0.8$. Closed curves are level curves of the logarithmic potential, dots mark the singularities of $f$, and straight line represents the interval $[-1,1]$.

Figure 2.6: Interpolation of $f$ with 41 Chebyshev nodes and $\beta = 2$. Closed curves are level curves of the logarithmic potential, dots mark the singularities of $f$, and straight line represents the interval $[-1,1]$. 
Figure 2.7: Lebesgue constant for different values of $\beta$. Dashed lines mark the Lebesgue constant values for polynomial interpolation.

where

$$L_k(x) = e^{-N^2/(x+1)^2-(x_k+1)^2} \prod_{j=0}^{N} \frac{(e^{\beta x} - e^{\beta x_j})}{(e^{\beta x_k} - e^{\beta x_j})}$$  \hspace{1cm} (2.11)

is the GRBF cardinal function. Notice that $L_k(x_k) = 1$, $L_k(x_j) = 0$ ($j \neq k$), and by (2.3) $L_k(x) \in \operatorname{Span}\{e^{-(x-x_k)^2/c^2}\}$. Thus, the unique GRBF interpolant can be written as

$$F(x) = \sum_{k=0}^{N} L_k(x) f(x_k)$$  \hspace{1cm} (2.12)

and it follows that

$$\|F - f\|_\infty \leq (1 + \Lambda_N^{GRBF}) \|F_{opt} - f\|_\infty,$$  \hspace{1cm} (2.13)

where $F_{opt}$ is the best approximation to $f$ in the GRBF subspace with respect to the infinity norm.

Figure 2.7 illustrates how the GRBF Lebesgue constant grows with $N$ for equispaced nodes (left) and Chebyshev nodes (right). As expected, for small $\beta$ the GRBF Lebesgue constants approximate the polynomial Lebesgue constants, which
behave asymptotically as $O(2^N/(N \log N))$ for equispaced nodes and $O(\log N)$ for Chebyshev nodes [24, 70]. This figure shows that the Lebesgue constants grow exponentially for both node distributions, except for large values of $\beta$ for uniform nodes and small values of $\beta$ for Chebyshev nodes.

In the presence of rounding errors, (2.13) indicates that if computations are carried out with precision $\varepsilon$ then the solution will generally be contaminated by errors of size $\varepsilon \Lambda_{GRBF}^N$ [70]. For instance, if $f(x) = 1/(x^2 - 1.8x + 0.85)$ and $\beta = 2$, the convergence of the interpolation process on Chebyshev nodes in double precision stops at $N = 80$, with a minimum residue of $O(10^{-7})$ due to rounding error. Similar results have been observed on equispaced nodes if $\beta$ is small.

A connection between the Runge phenomenon and large Lebesgue constants can be made using (2.13). Notice that

$$
\|F^\text{opt} - f\|_{\infty} \leq \|e^{-(x+1)^2/x^2} - f\|_{\infty};
$$

on the other hand, for functions that cause the phenomenon, $\|F - f\|_{\infty}$ grows exponentially with $N$, implying that $\Lambda_{GRBF}^N$ must also grow exponentially.

### 2.3.3 Stable interpolation nodes

Our goal now is to find node distributions that lead to a convergent interpolation process whenever the function is analytic on $[-1,1]$. This only happens if $[-1,1]$ is itself an equipotential, as it is for Chebyshev density in polynomial interpolation. Therefore, we seek a density function $\mu$ that satisfies

$$
\frac{\beta}{4}(x+1)^2 = \int_{-1}^{1} \log(|e^{\beta x} - e^{\beta t}|) \mu(t) dt + \text{constant}, \quad x \in [-1,1].
$$

(2.14)

In order to find a numerical solution to this integral equation, we assume that the optimal $\mu$ can be approximated by

$$
\mu(t) \approx \sum_{k=0}^{N_\nu} a_k \frac{T_{2k}(t)}{\sqrt{1-t^2}},
$$

(2.15)

where $T_{2k}$ is the Chebyshev polynomial of order $2k$. We consider only even functions in our expansion because we expect the density function to be even due to symmetry.
Figure 2.8: Numerical approximations of the optimal density functions for several values of $\beta$. Dashed line shows the Chebyshev density function.

This generalizes the Chebyshev density function $\mu(t) = \pi^{-1}(1 - t^2)^{-1/2}$. We also tried more general expressions, replacing $\sqrt{1 - t^2}$ with $(1 - t^2)^{-\alpha}$, and found that $\alpha = 1/2$ was suitable.

Figure 2.8 shows density functions computed with the expression above. We computed the coefficients $a_k$ by discrete least-squares and the integral in (2.14) was approximated by Gaussian quadrature. We used $N_\mu = 9$ and 50 points to evaluate the residue in the least-squares process. With this choice of parameters the residual was less than $10^{-7}$ in all computations.

In Figure 2.9 we show 21 nodes computed using (2.14) and (2.15) for several values of $\beta$. For large values of $\beta$ the nodes are nearly equally spaced and for small values they are approximately equal to Chebyshev extreme points. The optimal equipotentials obtained for $\beta = 0.1, 0.8, 2, 5$ are presented in Figure 2.10. For all these values of $\beta$, $[-1, 1]$ seems to be a level curve of the logarithmic potential.

As mentioned in section 2.3.2, in the presence of rounding errors the Lebesgue
Equispaced nodes

Chebyshev nodes

Figure 2.9: Node locations obtained using density function computed by solving the integral equation (2.14) for $N = 20$ and several values of $\beta$.

constant also plays a crucial role. Fortunately, for the optimal nodes computed numerically in this section, experiments suggest that the Lebesgue constant grows at logarithmic rate. Figure 2.11 presents computed Lebesgue constants for different values of $\beta$ on optimal nodes.

Figure 2.12 shows the convergence of GRBF interpolation to the four functions used to illustrate the Runge phenomenon in section 2.3.1. Now all four functions can be approximated nearly to machine precision. The algorithm used to obtain these data is presented in Section 2.4. Notice that the convergence rates are determined by the singularities of the function being interpolated. Dashed lines in this figure mark the convergence rates predicted by (2.8). For instance, if $f(x) = 1/(1 + 25x^2)$ and $\beta = 0.8$ then $m_\epsilon$ is approximately the difference between the value of the potential in $[-1, 1]$ and the potential at $z = 0.2i$ (where $f$ is singular), giving $m_\epsilon \approx 0.23$.

Notice that for $\beta = 2$ the equipotentials that enclose the interval $[-1, 1]$ are
Figure 2.10: Contour maps of the logarithmic potential obtained with numerically approximated optimal density function.
Figure 2.11: Lebesgue constant for different values of $\beta$ and optimal node distribution.

Figure 2.12: Maximum error of the interpolation process using optimal nodes. Left: $f(x) = 1/(1 + 25x^2)$ (●) and $f(x) = 1/(4 + 25x^2)$ (*). Right: $f(x) = 1/(x^2 - 1.8x + 0.82)$ (●) and $f(x) = 1/(x^2 - 1.8x + 0.85)$ (*). Dashed lines mark convergence rates predicted by (2.8).
contained in a bounded region (Fig. 2.10). This indicates that the convergence rate
given by (2.8) is the same for all functions that have singularities outside this region.
In polynomial interpolation, convergence to entire functions is much faster than to
functions with finite singularities. This is not the case for GRBFs. With $\beta = 2$
we found that the rate of convergence of interpolants of $1/(1 + 4x^2)$, $1/(100 + x^2)$,
$\sin(x)$, and $|x + 2|$ were all about the same. What these functions have in common
is that they are analytic inside the smallest region that includes all equipotentials
that enclose $[-1, 1]$.

It is also worth noting that the one-parameter family $\mu_\gamma$ of node density func-
tions proportional to $(1 - t^2)^{-\gamma}$ was used in [25] to cluster nodes near boundaries in
RBF approximations. Although numerical results showed improvement in accuracy,
no clear criteria for choosing $\gamma$ was provided in those papers. By using these node
density functions and minimizing the residue in (2.14) with respect to $\gamma$, we found
that optimal values of $\gamma$ are approximately given by $\gamma \cong 0.5e^{-0.3\beta}$. We point out,
however, that interpolations using these density functions may not converge if large
values of $N$ are required.

2.3.4 Location of Centers

Up to this point we have assumed that the centers are uniformly distributed
on $[-1, 1]$. Here we briefly investigate the consequences of choosing centers $\xi_k$ that
are equispaced on the interval $[-L, L]$, where $L \neq 1$, and also discuss results where
centers are not equally spaced. Taking centers outside the interval of approximation
is of practical interest, as it was suggested in [25, 47] to improve edge accuracy.

For equispaced centers on $[-L, L]$, a straightforward modification of (2.3)
gives

$$F(x) = e^{-N\beta(x+L)^2} \sum_{k=0}^{N} \tilde{\lambda}_k e^{k\beta x},$$
where $\beta = 4L/Nc^2$. In this case the logarithmic potential becomes

$$U^L_\beta(z) = \frac{\beta}{4L} \Re [(z + L)^2] - \int_{-1}^{1} \log(|e^{\beta z} - e^{\beta t}|) \mu(t) dt.$$ 

Equipotentials for different values of $L$ are presented in Figure 2.13. We considered equispaced interpolation nodes on $[-1, 1]$. Notice that if $L = 0.5$ there is no guarantee of convergence, as no equipotential encloses $[-1, 1]$. For $L = 0.75, 1.25$, and $1.5$, there are equipotentials enclosing this interval. However, the region where $f$ is required to be smooth seems to increase with $L$. We also point out that the asymptotic behavior for small $\beta$ given in (2.9) holds independently of $L$, consistent with the fact that center location is irrelevant in the polynomial limit.

It is common practice to chose the same nodes for centers and interpolation. In Figure 2.14 we show the graphs of the GRBF interpolants, for $f(x) = 1/(x^2 - 1.8x + 0.82)$ and $f(x) = 1/(x^2 - 1.8x + 0.85)$, where both centers and interpolation nodes are Chebyshev points. These data suggest that interpolation with Chebyshev centers also suffers from the Runge phenomenon. These results are similar to the ones obtained in Figure 2.6 for equispaced centers. Notice that we cannot use the definition involving $h$ for $\beta$ if the centers are not equispaced; in this case we use the definition $\beta = 4/(Nc^2)$.

For centers arbitrarily distributed in $[-1, 1]$, an expression involving powers of $s$ can also be derived for the GRBF interpolant. Consider a nonuniform spacing between centers: $h_j = \xi_{j+1} - \xi_j$. We then have that

$$(x - \xi_{j+1})^2 = (x - \xi_j - h_j)^2 = (x - \xi_j)^2 - 2xh_j + 2\xi_jh_j + h_j^2.$$ 

New members of the GRBF basis can, thus, be constructed from old ones in a
centers on $[-0.5, 0.5]$  

centers on $[-0.75, 0.75]$  

centers on $[-1.25, 1.25]$  

centers on $[-1.5, 1.5]$  

Figure 2.13: Equipotentials for $\beta = 2$ (compare with Fig. 2.4). Uniformly distributed centers on interval specified above. Interpolation points are uniformly distributed on $[-1, 1]$
recursive fashion:

\[ e^{-(x-x_1)^2/c^2} = C_0 e^{-(x-x_0)^2/c^2} e^{2xh_0/c^2}, \]
\[ e^{-(x-x_2)^2/c^2} = C_1 e^{-(x-x_1)^2/c^2} e^{2xh_1/c^2} = C_0 C_1 e^{-(x-x_0)^2/c^2} e^{2x(h_0+h_1)/c^2}, \]
\[ e^{-(x-x_3)^2/c^2} = C_2 e^{-(x-x_0)^2/c^2} e^{2x(h_0+h_1+h_2)/c^2}, \]
\[ \vdots, \]

where the \( C_j \) are independent of \( x \). We can now write a GRBF approximation as

\[ F(x) = e^{-(x+1)^2/c^2} \left( \lambda_0 + \sum_{k=0}^{N-1} \lambda_k e^{2x(h_0+...+h_k)/c^2} \right). \]

Letting \( s_0 = \exp\{2xh_0/c^2\} \), this expansion takes the form

\[ G_2(s_0) = \exp \left\{ - \left( \frac{c^2}{2h_0 \log(s_0) + 1} \right)^2 / c^2 \right\} \left( \lambda_0 + \sum_{k=0}^{N-1} \lambda_k s_0^{\alpha_k} \right), \]

where \( \alpha_k = 1 + h_0/h_1 + ... + h_0/h_k \). The difficulty in this case is that the powers \( \alpha_k \) are not integers in general.
2.4 Algorithmic implications

As mentioned in Chapter 1, the interpolation matrix \( \phi(\|x_i - \xi_j\|) \) in most conditions becomes ill-conditioned as the approximations get more accurate, to the extent that global interpolants are rarely computed for more than a couple of hundred nodes. Although several researchers have addressed this issue introducing more stable algorithms, e.g. the Contour-Padé algorithm [29] and domain decomposition techniques [3, 41], this remains one of the main obstacles in the application of RBF-based methods to practical applications. For GRBFs with equispaced centers, however, (2.12) provides an explicit interpolation formula through the use of the cardinal functions \( L_k \), so the difficulty of inverting the interpolation matrix can be avoided. This is equivalent to Lagrange polynomial interpolation.

Notice that the exponential term \( e^{-\frac{N^2}{4}(x+1)^2 - (x_k+1)^2)} \) in (2.11) becomes very close to zero for certain values of \( x \) if \( N \) is large, affecting the accuracy of the approximations. A modification of (2.11) improves matters:

\[
L_k(x) = \prod_{j=0}^{N} \frac{e^{-\frac{1}{2}(x+1)^2 - (x_k+1)^2}(e^{\beta x} - e^{\beta x_j})}{(e^{\beta x_k} - e^{\beta x_j})},
\]  

(2.16)

The direct implementation of (2.16) together with (2.12) provides a simple algorithm for computing the GRBF interpolant for moderate values of \( N \). In our experiments, effective computations were carried out up to \( N = 300 \). We shall next derive a more stable formula to handle larger problems.

In [4] Berrut and Trefethen point out the difficulties of using the standard Lagrange formula for practical computations and argue that the barycentric form of Lagrange interpolation should be the method of choice for polynomial interpolation. For GRBFs we define the barycentric weights by

\[
w_k = \left( \prod_{j=0}^{N} e^{-\frac{1}{2}(x_k+1)^2}(e^{\beta x_k} - e^{\beta x_j}) \right)^{-1},
\]

(2.17)
and thus we have that

\[ L_k(x) = L(x) \frac{w_k}{e^{-\frac{\beta}{4} (x+1)^2} (e^{\beta x} - e^{\beta x_k})} \quad (x \neq x_k), \]

where

\[ L(x) = \prod_{j=0}^{N} e^{-\frac{\beta}{4} (x+1)^2} (e^{\beta x} - e^{\beta x_j}). \]

Therefore, the GRBF interpolant can be written as

\[ F(x) = L(x) \sum_{k=0}^{N} \frac{w_k}{e^{-\frac{\beta}{4} (x+1)^2} (e^{\beta x} - e^{\beta x_k})} f(x_k). \quad (2.18) \]

For reasons of numerical stability, it is desirable to write \( L \) as a sum involving the barycentric weights. For polynomial interpolation this is done by considering that 1 can be exactly written in terms of interpolation formulas, since it is itself a polynomial. Unfortunately, a constant function is not exactly represented in terms of GRBFs. Nevertheless, this difficulty can be circumvented if we properly choose a function that belongs to the GRBF space. In our implementation, we consider the function

\[ v(x) = \frac{1}{N} \sum_{k=0}^{N} e^{-\frac{N \beta}{4} (x - x_k)^2}. \]

Notice that in this case,

\[ L(x) = \frac{v(x)}{\sum_{k=0}^{N} \frac{w_k}{e^{-\frac{\beta}{4} (x+1)^2} (e^{\beta x} - e^{\beta x_k})} v(x_k)}. \]

Combining the last expression with (2.18) gives our GRBF barycentric formula:

\[ F(x) = v(x) \frac{\sum_{k=0}^{N} \frac{w_k}{(e^{\beta x} - e^{\beta x_k})} f(x_k)}{\sum_{k=0}^{N} \frac{w_k}{(e^{\beta x} - e^{\beta x_k})} v(x_k)}. \quad (2.19) \]

As mentioned in [4], the fact that the weights \( w_k \) appear symmetrically in the denominator and in the numerator means that any common factor in all the weights...
may be canceled without affecting the value of $F$. In some cases it is necessary to rescale terms in (2.17) to avoid overflow. In our implementation we divided each term by $\prod_{j=1}^{N} |e^{\beta x_j} - e^{-\beta}|^{1/N}$.

In [38] Higham shows that for polynomials the barycentric formula is forward stable for any set of interpolation points with a small Lebesgue constant. Our numerical experiments suggest that the GRBF barycentric formula is also stable.

Figure 2.12 was obtained using the barycentric formula. We point out that the direct inversion of the interpolation matrix becomes unstable even for moderate values of $N$. In Figure 2.15 we compare the convergence of the GRBF interpolant computed with the barycentric formula with the one found by inverting the interpolation matrix (standard RBF algorithm). We first computed approximations with $\beta$ fixed (left). Notice that for the standard implementation convergence rate changes at a level around $10^{-2}$ and the method becomes very inefficient for larger values of $N$. For the barycentric formula, on the other hand, convergence continues to machine precision. For these approximations we used nodes computed with an approximate
optimal density function as in section 2.3.3. We also compared the algorithms for fixed $c$. In this instance we used Chebyshev nodes, as $c$ constant implies that $\beta \to 0$ as $N$ becomes large and approximations become polynomial. The performance of the standard algorithm is apparently nonconvergent in this case.

2.5 Differentiation Matrices

Although the differentiation matrix for RBFs can be generated using the guidelines presented in section 1.2.3, for GRBFs in 1-D with equally spaced nodes, we can derive an explicit formula for the entries of the differentiation matrix using the barycentric formula (2.19).

We can rewrite (2.11) as

$$L_j(x) = \frac{w_j v(x)}{s_i(x)} \left( \frac{e^{\beta x} - e^{\beta x_i}}{e^{\beta x_i} - e^{\beta x_k}} \right), \quad (2.20)$$

where $s_i(x) = \sum_{k=0}^{N} w_k v(x_k) (e^{\beta x_i} - e^{\beta x_k})/(e^{\beta x} - e^{\beta x_k})$. Multiplying both sides of (2.20) by $s_i$ and differentiating the resulting equation gives

$$L'_j(x) s_i(x) + L_j(x) s'_i(x) = w_j \left( v'(x) \left( \frac{e^{\beta x} - e^{\beta x_i}}{e^{\beta x_i} - e^{\beta x_k}} \right) + v(x) \left[ \frac{e^{\beta x} - e^{\beta x_k}}{e^{\beta x} - e^{\beta x_k}} \right]' \right).$$

Since $s_i(x_i) = w_i$ and $L_j(x_i) = 0$ for $j \neq i$, we have

$$L'_j(x_i) = \frac{\beta v(x_i) e^{\beta x_i} w_j}{(e^{\beta x_i} - e^{\beta x_j}) w_i}. \quad (2.21)$$

In order to derive an expression for $L'_j(x_j)$, notice that $v(x) = \sum_{k=0}^{N} L_k(x) v(x_k)$. Differentiating $v$ and solving the resulting equation for $L'_j(x_j)$ gives

$$L'_j(x_j) = \frac{v'(x_j)}{v(x_j)} - \sum_{k=0}^{N} L'_k(x_j) \frac{v(x_k)}{v(x_j)}. \quad (2.22)$$

Therefore, the entries of the first-order differentiation matrix are $D_{i,j} = L'_j(x_i)$. This formulation for $D$ is far more robust than the one using the inverse of
the interpolation matrix, which is ill-conditioned in most cases. Moreover, it requires only $O(N^2)$ operations, instead of $O(N^3)$ usually required to invert a matrix.

In Figure 2.16 we present the error of the approximation of $df/dx$ from values of $f(x) = 1/(1 + 25x^2)$ at nodes in $[-1, 1]$. We used $\beta = 2$ and the nodes were generated with approximate optimal density functions. We observe that the use of both stable nodes and the stable formulation for the differentiation matrix permits convergence to machine precision, while the use of differentiation matrices obtained with the standard algorithm presented in section 1.2.3 gives very poor results for large $N$. 

**Figure 2.16:** Error in the approximation of $df/dx$, where $f(x) = 1/(1 + 25x^2)$, using the barycentric formulation for $D$ (●) and the standard RBF algorithm (∗).
Chapter 3

LEAST-SQUARES APPROXIMATIONS AND
ARNOLDI-LIKE ITERATIONS

In the previous chapter we showed that Gaussian RBF interpolation in 1-D with equally spaced centers requires special nodes for stability, as in polynomial interpolation. Stable nodes for other radial functions, like multiquadrics, however, are yet not known. Moreover, the task of finding stable nodes becomes more complex – perhaps impossible – in higher dimensions with complicated geometry. We propose using least-squares approximations to avoid this difficulty.

In [10] Buhmann presents several benefits of using least squares instead of interpolation. Least squares are widely used for data smoothing. Interpolation is often not suitable for noisy data, as happens in many practical applications. In such cases, the data need not to be all interpolated at once. Moreover, RBF least-squares methods open the possibility of separation of accuracy and stability characteristics. The resolving power of a certain radial basis is defined by its shape functions and the location of its centers. Since stability can usually be achieved through least squares, one can take advantage of placing centers more densely in certain regions of the domain where better resolution is desired.

In this chapter, we explore least-squares approximations. The fundamental ingredients for successful least-squares discretizations are: well conditioned bases that can accurately approximate the target function and a criteria for choosing the least-squares nodes. Unfortunately, smooth well-conditioned bases are not easy to
obtain for spectral methods in general multidimensional regions (one of the reasons that justify the use of finite elements or domain decomposition). In what follows, we shall explore Arnoldi-like iterations for the orthogonalization of polynomial and radial bases. We also discuss the trade-off between node clustering and the number of nodes needed to achieve a desired accuracy. In the final section of this chapter we derive differentiation matrices that shall be used in Chapters 4 and 5 for time-dependent and eigenvalue problems.

3.1 The ingredients for accurate approximations

Let \( \{x_i\}, i = 1, \ldots, M \), be a set of nodes in the domain of interest and \( \{\Phi_j\}, j = 0, \ldots, N \), be a basis of the approximating space, the goal is to minimize the norm of the residual \( R(\lambda, x) = f(x) - \sum_{j=0}^{N} \lambda_j \Phi_j(x) \). Since our main interest is on discrete approximations, we seek coefficients \( \lambda_i \) that minimize the residual

\[
R(\lambda) = \| f - A\lambda \|. \tag{3.1}
\]

where \( f_i = f(x_i), \lambda_i = \lambda_i, A_{i,j} = \Phi_j(x_i) \), and \( \| \cdot \| \) is the Euclidean norm in \( \mathbb{R}^M \). If the values of the function \( f \) at the least-squares nodes are known, the coefficient vector can be found by computing the pseudo-inverse of \( A \), i.e., \( \lambda = A^+ f \), where \( + \) denotes the pseudo-inverse matrix.

3.1.1 A well-conditioned basis

Whether using interpolation or least squares, one must choose a suitable approximating space. An orthonormal basis of the approximating space is ideal to find the best fit to a given function. For spectral methods, however, rounding errors often prevent straightforward orthogonalization procedures, such as the Gram-Schmidt orthogonalization, from converging. Consider for instance the polynomial approximation of \( f(x) = 1/(1 + 16x^2) \). We seek the discrete least-squares approximation on 500 Chebyshev nodes. Figure 3.1 presents the error of the approximations for
Figure 3.1: Error in the least-squares approximation of $f(x) = 1/(1 + 16x^2)$ using monomials (*) and Chebyshev polynomials (o).

For intervals in the real line, explicit expressions for orthogonal polynomials are well-known. In this case, one can used Jacobi polynomials to avoid ill-conditioning of the approximating matrices. Notice that in Figure 3.1, $f(x) = 1/(1 + 16x^2)$ was approximated to machine precision with a polynomial of degree 150. Unfortunately, explicit expressions for orthogonal polynomials are not known...
for complex domains in higher-dimensions\(^1\) and for RBFs explicit formulas for orthogonal bases are not known even in 1-D regions. We shall explore Arnoldi-like iterations as an alternative to generate well-conditioned approximation matrices later in this chapter.

### 3.1.2 Clustering and aspect ratio

Discrete least-squares approximations are more flexible in terms of node location. This flexibility however comes at a price; depending on the node distribution the number of least-squares nodes may be much larger than the needed to achieve similar accuracy of approximations obtained with optimally distributed nodes. For spectrally accurate approximations, this usually means that if nodes are not clustered near the boundary, a high aspect ratio (the number of least-square nodes divided by the dimension of the approximating space) may be needed for good approximations.

Consider for instance the polynomial approximation of the Runge function \(f(x) = 1/(1 + 49x^2)\) on \([-1, 1]\). Figure 3.2 shows the number of least-squares nodes \(M\) needed to find the \(N\)th degree polynomial that approximates \(f\) to about the same accuracy obtained with interpolation on Chebyshev nodes. If equispaced nodes are used, \(M\) must be \(O(N^2)\) to achieve that result; the dashed line that best fit this date is given by \(M = 0.089N^2\). Imperfect clustering improves matters. If nodes with density function proportional to \((1 - x^2)^{-0.3}\) are used\(^2\), experiments suggest a relationship between \(M\) and \(N\) given empirically by \(M = 0.43N^{1.31}\) (dashed line).

In the previous chapter we showed that for \(\beta = 2\), GRBF interpolation with equally spaced centers of \(f(x) = 1/(x^2 - 1.8x + 0.82)\) is divergent on Chebyshev

---

\(^1\) For certain domains, examples of orthogonal polynomials in several variables are presented in [19].

\(^2\) See [24] for details about node density functions proportional to \((1 - x^2)^{-\gamma}\).
Figure 3.2: Number of least-squares nodes needed to approximate $f(x) = 1/(1 + 49x^2)$ to about the same accuracy obtained with interpolation on Chebyshev nodes: (*) equally spaced least-squares nodes; and (○) imperfectly clustered least-squares nodes.
nodes. If least squares are used, however, adding a few extra nodes to the approximations is enough to remedy the problem. Figure 3.3 presents the number of Chebyshev nodes needed to obtain the least-squares approximation to this function with the same accuracy shown in Figure 2.12, where interpolation on optimal nodes was used. The best fit to the data in Fig. 3.3 gives $M = 1.28N$ (dashed line).

For small $\beta$, Chebyshev nodes are nearly optimal, but large values of $\beta$, equispaced nodes tend to be a better option (although asymptotically there is always need for some clustering near boundaries) and the aspect ratio $M/N$ usually increases with $\beta$. To generate this data we used an Arnoldi-like iteration for GRBFs with equally spaced centers. This algorithm shall be presented later in this chapter.

Compared to a standard polynomial basis, two of the most important features of RBFs are their flexibility in the shape parameter and center locations. The center locations can be exploited to increase resolution in specific regions. For instance,
the function $f(x) = \tanh(10x)$ varies sharply near $x = 0$ and is almost constant in other parts of the interval $[-1, 1]$. Therefore, clustering centers more densely in the middle of the interval, one would expect to get better accuracy. This is indeed the case, as shown in Figure 3.4. In this plot we compare approximations with equally spaced centers and adapted centers given by $\xi_j = (2/\pi) \sin^{-1}(-1 + 2j/N)$. As expected, the error decays faster if adapted centers are used. A plot of equivalent polynomial approximations is also shown – RBFs give significant better results in this case. For the multiquadric approximations $c = 0.5$ was used. Chebyshev least-squares nodes were used with aspect ratio of 2 and the error was measured in the maximum norm on 1000 nodes. The continuation of this plot for larger values of $N$ becomes difficult because of the ill-conditioning of RBF coefficient matrices. We shall next explore Arnoldi-like iterations as an attempt to circumvent this difficulty for Gaussian RBFs.
3.2 Arnoldi-like iterations for polynomials

The Arnoldi iteration has been widely used to construct orthogonal bases of Krylov subspaces, \( \text{Span}(b, Ab, A^2b, \ldots, A^nb) \), where \( A \) is a given matrix and \( b \) is a vector [69]. In general the basis \( \{b, Ab, A^2b, \ldots, A^nb\} \) is ill-conditioned, but the Arnoldi iteration allows well-conditioned computations, making it the starting point for many iterative algorithms for algebraic eigenvalue problems and linear systems.

The Arnoldi iteration can also be used to generate orthogonal polynomials as depicted in [69]. In fact, for the construction of univariate orthogonal polynomials the algorithm reduces to the Lanczos iteration presented below:

\[
\begin{align*}
q_0(x) &= 1, \quad q_0'(x) = 0 \\
q_0(x) &= q_0(x)/\|q_0\|, \quad q_0'(x) = q_0'(x)/\|q_0\|
\end{align*}
\]

For \( k = 1 : N \)

\[
\begin{align*}
v(x) &= xq_{k-1}(x) \\
v'(x) &= xq_{k-1}'(x) + q_{k-1}(x) \\
v(x) &= v(x) - \langle q_{j-1}, v \rangle q_{j-1}(x) - \langle q_j, v \rangle q_j(x) \\
v'(x) &= v'(x) - \langle q_{j-1}, v \rangle q_{j-1}'(x) - \langle q_j, v \rangle q_j'(x) \\
q_k(x) &= v(x)/\|v\| \\
q_k'(x) &= v'(x)/\|v\|
\end{align*}
\]

Notice that we also included the steps to generate the derivatives \( q_j' \) of \( q_j \), as we are also interested in using this algorithm with numerical schemes for PDEs. In our implementation \( \langle \cdot, \cdot \rangle \) is the usual discrete \( L_2 \) inner product.

As stated in [69], this algorithm is just the implementation of the well known three-term recurrence for constructing a sequence of orthogonal polynomials on \([-1, 1]\):

\[
xq_k(x) = a_{k-1}q_{k-1}(x) + b_kq_k(x) + a_kq_{k+1}(x).
\]

(3.2)

For Legendre polynomials and other classical Jacobi polynomials the entries \( a_k \) and \( b_k \) are known analytically [14]. The goal is to extend this algorithm to generate
orthogonal bases for cases where the analytical expressions are not known, like for polynomials in 2-D regions and Gaussian RBFs.

Bivariate polynomials in general do not satisfy a three-term recurrence. An orthogonal sequence, nonetheless, can be generated with the following Arnoldi-like iteration:

\begin{verbatim}
q_0(x, y) = 1
q_0(x, y) = q_0(x, y)/\|q_0\|
For k = 1 : N
    v(x, y) = x q_{k-1,k/2}(x, y)
    q_{k(k+1)/2} ← orthonormal component of v to \{q_0, ..., q_{k(k+1)/2-1}\}
For j = 1 : k
    v(x, y) = y q_{k-1,k/2+j}(x, y)
    q_{k(k+1)/2+j} ← orthonormal component of v to \{q_0, ..., q_{k(k+1)/2+j-1}\}
end
\end{verbatim}

In our implementation, the orthogonal component of \(v\) is computed with modified Gram-Schmidt orthogonalization. For simplicity we did not include the steps to generate the derivatives \(q'_k\); these functions can be computed in similar fashion as to what was presented in the 1-D algorithm. It can be easily verified that the basis formed with the functions \(q_k\) satisfies:

\[
\text{span}\{q_0\} = \text{span}\{1\}
\]
\[
\text{span}\{q_0, q_1\} = \text{span}\{1, x\}
\]
\[
\text{span}\{q_0, q_1, q_2\} = \text{span}\{1, x, y\}
\]
\[
\text{span}\{q_0, q_1, q_2, q_3\} = \text{span}\{1, x, y, x^2\}
\]
\[
\text{span}\{q_0, q_1, q_2, q_3, q_4\} = \text{span}\{1, x, y, x^2, xy\}
\]
\[
\text{span}\{q_0, q_1, q_2, q_3, q_4, q_5\} = \text{span}\{1, x, y, x^2, xy, y^2\}
\]
\[
\ldots
\]
3.3 Arnoldi-like iterations for Gaussian RBFs

For GRBFs with equally spaced centers in 1-D, expression (2.3) shows that that a new member of the GRBF basis can be constructed from an old one through pointwise multiplication by a function of \( x \). This is the starting point for our Arnoldi-like iteration. The Arnoldi algorithm to produce an orthogonal GRBF basis on \([-1, 1]\) is depicted below:

\[
\begin{align*}
q_0(x) &= \exp(-N\beta(x+1)^2/4) \\
q_0'(x) &= (-N\beta(x+1)/2)q_0(x) \\
q_0(x) &= q_0(x)/\|q_0\| \\
q_0'(x) &= q_0'(x)/\|q_0\|
\end{align*}
\]

For \( k = 1 : N \)

\[
\begin{align*}
v(x) &= q_{k-1}(x) \exp(\beta x) \\
v'(x) &= q_{k-1}'(x) \exp(\beta x) + \beta q_{k-1}(x) \exp(\beta x)
\end{align*}
\]

For \( j = 1 : k - 1 \)

\[
\begin{align*}
v(x) &= v(x) - \langle q_j, v \rangle q_j(x) \\
v'(x) &= v'(x) - \langle q_j, v \rangle q_j'(x)
\end{align*}
\]

end

\[
\begin{align*}
q_k(x) &= v(x)/\|v\| \\
q_k'(x) &= v'(x)/\|v\|
\end{align*}
\]

end

The innermost loop is the modified Gram-Schmidt orthogonalization. In some cases reorthogonalization may be needed due to rounding errors. In this algorithm we also included the steps to generate the derivatives \( q_j' \) of \( q_j \).

**Theorem 3.1** If roundoff errors are not present, this algorithm reduces to a Lanczos-like iteration. For the orthogonal basis \( \{q_0, \ldots, q_N\} \) generated with the algorithm described above, there exist constants \( a_k \) and \( b_k \) such that

\[
sq_k = a_{k-1}q_{k-1} + b_kq_k + a_kq_{k+1},
\]

55
where \( s = e^{\beta x} \). Furthermore, if \( \beta \ll 1 \) then \( a_k = O(\beta) \) and \( b_k = 1 + O(\beta) \).

Proof. The three term formula can be easily verified since \( sq_j \in \text{Span}\{q_0, \ldots, q_{j+1}\} \), which implies that \( \langle sq_j, q_j \rangle = \langle q_j, sq_j \rangle = 0 \) if \( j+1 < k \). Hence, \( sq_k = c_k q_{k-1} + b_k q_k + a_k q_{k+1} \) for some constants \( c_k, b_k, \) and \( a_k \). And \( c_k = \langle sq_k, q_{k-1} \rangle = \langle q_k, sq_{k-1} \rangle = \langle q_k, c_{k-1} q_{k-2} + b_{k-1} q_{k-1} + a_{k-1} q_k \rangle \) gives \( c_k = a_{k-1} \).

The bounds for \( b_k \) can be obtained from

\[
b_k = \langle sq_k, q_k \rangle = \sum_{j=0}^{M} e^{\beta x_j} q_k^2(x_j) dx,
\]

which gives \( e^{-\beta} \leq b_k \leq e^{\beta} \) and \( b_k = 1 + O(\beta) \) for \( \beta \ll 1 \). Similarly, we can show that \( e^{-\beta} \leq \|sq_k\| \leq e^{\beta} \). Now using that this basis is orthonormal, we have \( |a_{k-1}|^2 + |a_k|^2 = |b_k|^2 - \|sq_k\|^2 \), and it follows that \( a_k = O(\beta) \), for small \( \beta \). □

We point out that for \( c > 0.7 \), a simple modification of (2.3) gives a well-conditioned GRBF approximation,

\[
F(x) = e^{-(x+1)^2/c^2} \sum_{k=0}^{N} \lambda_k T_k \left( \frac{e^{\beta x} - \cosh(\beta)}{\sinh(\beta)} \right) = e^{-\frac{\beta N}{4} (x+1)^2} \sum_{k=0}^{N} \lambda_k T_k \left( \frac{e^{\beta x} - \cosh(\beta)}{\sinh(\beta)} \right),
\]

(3.3)

where \( T_k \) is the \( k \)th Chebyshev polynomial. For smaller values of the shape parameter, however, the exponential term in front of the sum becomes very close to zero for some values of \( x \), to the extent that accuracy is compromised. This is usually the case when the parameter \( \beta \) is fixed. Note that in the limit \( c \to \infty \), (3.3) becomes a sum of polynomials, in agreement with [18, 30].

Figure 3.5 presents the maximum error of the least-squares approximation of \( f(x) = \tanh(10x) \) for GRBFs with equally spaced centers and \( \beta = 2 \). Notice that if the standard Gaussian basis is used, the convergence is compromised for \( N \) larger than 30 due to rounding errors. If the basis is generated with the Arnoldi iteration, on the other hand, spectral convergence is maintained for all values of \( N \) and approximations can be computed to machine precision. For comparison, we
Figure 3.5: Error of the least-squares approximation of \( f(x) = \tanh(10x) \) for GRBFs with equally spaced centers and \( \beta = 2 \), using an orthogonal basis generated with an Arnoldi iteration (●) and the standard Gaussian radial basis (*). Polynomial approximation is also shown (+)

also show a convergence plot obtained with polynomial approximation, here we can also observe that RBFs give a better convergence rate.

3.4 Least-squares differentiation matrices

Different formulations allow the use of least squares for differential equations. Our approach is again based on differentiation matrices. Boundary conditions can be enforced weakly or strongly. In the first approach, the coefficients \( \lambda_k \) minimize the residual in the interior and boundary; i.e., boundary conditions may not be satisfied exactly at boundary nodes. In this case, a weighted norm may be used to penalize errors at boundary nodes more heavily than at interior nodes [20]. We found that this technique may require very large weights at the boundary to stabilize explicit RBF-based methods for time-dependent problems, which in turn usually worsens the condition number of the matrices used in the approximations. We believe that
a more efficient way to use discrete least-squares is to enforce boundary conditions strongly.

In order to derive a least-squares differentiation matrix, assume a general region in \( \mathbb{R}^d \) and Dirichlet boundary conditions. Given \( M_I \) nodes in the interior of the domain, we can write the norm of the residual at these points in matrix form,

\[
R(\lambda) = \| A_I \lambda - u_I \|.
\]  

(3.4)

Here \( \| \cdot \| \) is the discrete 2-norm in \( \mathbb{R}^{M_I} \) and \( A_I \) is the RBF evaluation matrix at interior nodes. If in addition we have \( M_B \) nodes on the boundary, we require

\[
A_B \lambda = u_B,
\]

(3.5)

where \( A_B \) is the RBF evaluation matrix at boundary nodes and \( u = [u_I, u_B]^T \) is a vector containing the values of the target function at the least-squares nodes.

To solve this constrained least-squares problem, we use the method of direct elimination. The method consists of reducing the number of unknowns in \( \lambda \) in order to satisfy the constraint equation (3.5), and solving the resulting unconstrained reduced system through a QR factorization. The derivation follows [6, section 5.1.2].

For simplicity, assume homogeneous boundary condition, i.e. \( u_B = 0 \). The constraint thus becomes

\[
A_B \lambda = 0.
\]

(3.6)

Since \( A_B \) may be rank-deficient or ill-conditioned in some cases, we shall not assume that \( A_B \) is full-rank. We start with the pivoted QR decomposition of \( A_B \),

\[
A_B \Pi_B = Q_B \begin{bmatrix} R_{B1} & R_{B2} \\ 0 & 0 \end{bmatrix},
\]

where \( \Pi_B \) is a permutation matrix, \( Q_B \in \mathbb{R}^{M_B \times M_B} \) is orthogonal, and \( R_{B1} \in \mathbb{R}^{r_B \times r_B} \) is upper triangular and nonsingular. Now (3.6) gives,

\[
R_{B1} \lambda_1 + R_{B2} \lambda_2 = 0
\]
where $\Pi_B^T\lambda = \begin{bmatrix} \hat{\lambda}_1 \\ \hat{\lambda}_2 \end{bmatrix}$. This gives

$$
\begin{bmatrix}
\hat{\lambda}_1 \\
\hat{\lambda}_2
\end{bmatrix} = 
\begin{bmatrix}
-R_{B1}^{-1}R_{B2} \\
I
\end{bmatrix}
\hat{\lambda}_2.
$$

Here $I$ an identity matrix which the order is determined by the size of the vector $\hat{\lambda}_2$.

Let

$$
\hat{A}_I = A_I \Pi_B^T \begin{bmatrix}
-R_{B1}^{-1}R_{B2} \\
I
\end{bmatrix}.
$$

We can then write the residual (3.4) in terms as a function of $\hat{\lambda}_2$,

$$
\hat{R}(\hat{\lambda}_2) = \left\| \hat{A}_I \hat{\lambda}_2 - u_I \right\|.
$$

Hence, the vector $\hat{\lambda}_2$ that minimizes the residual is

$$
\hat{\lambda}_2 = \hat{A}_I^+ u_I,
$$

where $\hat{A}_I^+$ is the pseudoinverse of $\hat{A}_I$. Finally, the solution of the constrained least-squares problem is

$$
\lambda = \Pi_B \begin{bmatrix}
-R_{B1}^{-1}R_{B2} \\
I
\end{bmatrix} \hat{A}_I^+ u_I.
$$

If we now let $A_I^L \in \mathbb{R}^{M_I \times N+1}$ be a matrix with elements $A_{i,j}^L = L \phi(\|x - \xi_i\|) |_{x=x_j}$, where $x_j$ are interior nodes, we obtain the discretization of $L$,

$$
L = A_I^L \Pi_B \begin{bmatrix}
-R_{B1}^{-1}R_{B2} \\
I
\end{bmatrix} \hat{A}_I^+.
$$

(3.7)

Notice that $L$ is now an $M_I \times M_I$ matrix such that $u_{IL} = Lu_I$.

We point out that this method can also be used with other boundary conditions, like Neumann boundary conditions, by modifying the constraint equation.
accordingly. In this case, \( A_B \) becomes the evaluation matrix of the normal components of the RBF derivatives, while \( u_B \) becomes the vector containing the normal derivatives of \( u \) on the boundary.

It is important to mention that since \( L \), in general, is the product of lower-rank matrices, it has several zero eigenvalues. Moreover the multiplicity of the zero eigenvalue is given by the number of columns minus the number of rows of the matrix \( \hat{A}_T^+ \). We therefore, do not recommend this approach for boundary value problems, where the inversion of \( L \) would be required. But in the next two chapters, we will show that this approach is viable for time-dependent and eigenvalue problems.

For problems in multi-dimensions, special care must be taken when the number of nodes on the boundary is relatively large. Notice that the degrees of freedom of a basis is reduced by the number of nodes on the boundary when the boundary constraint is imposed. In some of our experiments in 2-D, we observed some loss of accuracy when large aspect ratios were used. One can circumvent this difficulty by choosing a number of boundary nodes that is compatible with the dimension of the approximating space.
Chapter 4

TIME-DEPENDENT PROBLEMS

Although several authors have investigated RBF-based methods for timedependent problems [39, 40, 42], these methods remain underdeveloped compared to those for elliptic problems. Experience suggests that RBFs will produce discretizations that are unstable in time unless highly dissipative time stepping is used. In [25] Fornberg et al. suggested that one source of instability might be the relative inaccuracy of RBF approximations near the boundary. They also introduce some techniques to improve the accuracy near the edges, including clustering the nodes more densely near the boundary. Our findings in Chapter 2, also indicate the importance of node locations for RBF approximations. Specifically, we found that GRBFs are susceptible to a Runge phenomenon. Moreover, we verified that the use of GRBFs with arbitrary nodes usually lead to very large Lebesgue constants, making it difficult to obtain very accurate approximations. Using potential theory, however, one can obtain nodes that allow stable interpolation. In this chapter, we show that time stability can indeed be achieved by proper choices of nodes.

We are particularly interested in using the method of lines. Our approach is to explore the spectra of differentiation matrices. In the absence of boundary conditions, we prove that RBF methods are time-stable for all conditionally positive definite RBFs and node distributions. Therefore, for problems in periodic regions, like the unit circle and the unit sphere, RBFs are naturally stable. When boundary conditions need to be imposed, however, differentiation matrices may have eigenvalues with positive real part if arbitrary nodes are used.
We also explore the use of least-squares differentiation matrices derived in Section 3.4 for time-dependent problems since stable nodes are not known for general regions in high dimensions. The viability of least squares for time-dependent problem is empirically verified through numerical experiments. We also make use of the Arnoldi iteration for GRBFs introduced in the previous chapter, to avoid ill-conditioning in our computations.

4.1 RBFs and the method of lines

The method of lines refers to the idea of semidiscretizing in space and using standard methods for the resulting system of ordinary differential equations in time. A rule of thumb is that the method of lines is stable if the eigenvalues of the spatial discretized operator, scaled by the time-step $\Delta t$, lie in the stability region of the time-discretization operator, although in some cases the details of stability are more technical and restrictive [59].

4.1.1 Unstable eigenvalues: A case study

Consider as a test problem the transport equation,

$$u_t = u_x, \quad -1 \leq x \leq 1, \quad t > 0$$

$$u(t, 1) = 0, \quad u(0, x) = u_0(x).$$

A differentiation matrix for this problem can be easily obtained by noting that $\mathbf{u} = A\mathbf{\lambda}$ and $\mathbf{u}_x = B\mathbf{\lambda}$, where $A$ and $B$ are matrices with elements $A_{i,j} = \phi(\|x_i - \xi_j\|)$ and $B_{i,j} = \left. \frac{d}{dx} \phi(\|x - \xi_j\|) \right|_{x=x_i}$, $x_j$ are $N+1$ collocation nodes, $\mathbf{u}$ and $\mathbf{u}_x$ are vectors containing the RBF approximations of the function $u$ and $u_x$ at the collocation nodes, and $\mathbf{\lambda}$ is the vector of the coefficients $\lambda_j$. The differentiation matrix is then given by $\tilde{D} = BA^{-1}$. In order to enforce the boundary condition, assuming that
Figure 4.1: Eigenvalues of D for GRBFs with equally spaced nodes in $[-1, 1]$: (a) $c = 1$; (b) $N = 9$.

$x_N = 1$, we delete the last row and column of $\tilde{D}$ to produce a matrix we now call $D$. This leads to the coupled system of ordinary differential equations

$$u_t = Du. \quad (4.3)$$

The difficulty of using the method of lines with RBFs for (4.1)-(4.2) with arbitrary nodes is that some eigenvalues of the differentiation matrix may have positive real parts. This is illustrated in Figure 4.1. This figure was obtained with GRBFs using coincident, equally spaced centers and nodes in $[-1, 1]$. In Figure 4.1(a) the shape parameter is fixed, $c = 1$. Notice that for $N = 5$ all eigenvalues have negative real part, but as $N$ is increased eigenvalues move to the right half-plane making it difficult to use explicit methods for time integration. Similarly, in Figure 4.1(b) we observe that, for fixed $N$, eigenvalues move to the right half-plane as $c$ is increased. It is well known that the limit $c \to \infty$ is equivalent to polynomial interpolation so instability in this limit should be expected.

While experiments indicate that for a given $N$, the shape parameter $c$ can
be chosen small enough that all eigenvalues will lie in the left half-plane, this requirement is rather restrictive for large values of \( N \) — to the extent that spectral convergence seems to be compromised. This is illustrated in Figure 4.2. Multi-quadrics with equally spaced centers and nodes were used. In Figure 4.2(a) a plot of the largest value of the shape parameter \( c \) for which the spectrum of \( D \) lies on the left half-plane for several values of \( N \) is presented. Notice that the shape parameter must decay as \( O(1/N) \) in order to avoid unstable eigenvalues. In Figure 4.2(b) we show the corresponding error in the interpolation of \( \sin(4\pi x) \) if \( c \) is chosen according to Figure 4.2(a). Although the error decays for small values of \( N \), convergence stops for \( N \) about 100. The corresponding condition number of the interpolation matrix is also shown in Figure 4.2(c). For \( N = 100 \) the condition number is about \( 10^{11} \), which is not large enough to justify the behavior of the curve in Figure 4.2(b).

### 4.1.2 Spectra in the absence of boundaries

In polynomial approximation, boundaries play a major role in stability. Similar observations have been made experimentally in RBF approximation [29]. There is reason to think, then, that in the absence of boundaries (e.g. the differentiation matrix \( \tilde{D} \)), eigenvalue stability is possible. In this section we show that this is indeed the case.

Assume that nodes and centers coincide and recall from section 1.2.2 that for conditionally positive definite radial functions an RBF approximation takes the form

\[
\begin{bmatrix}
A & P \\
P^T & 0
\end{bmatrix}
\begin{bmatrix}
\lambda \\
\alpha
\end{bmatrix} =
\begin{bmatrix}
u \\
0
\end{bmatrix},
\]

where

- the elements of \( A \) are \( A_{i,j} = \phi(\|x_i - x_j\|) \),
- the elements of \( P \) are \( P_{i,j} = p_j(x_i) \),
- and \( u_i = u(x_i) \)
Figure 4.2: (a) Largest value of $c$ for which the spectrum of $D$ has non-positive real part — data obtained with multiquadrics with equally spaced centers and nodes. (b) The corresponding error in the interpolation of $\sin(4\pi x)$. (c) The corresponding condition number of the interpolation matrix.
We are interested in the spectrum of the finite-dimensional RBF operators that represent a linear differential operator \( \mathcal{L} \), like the Laplacian or a convection operator. From (1.5) we have that

\[
\begin{bmatrix}
\lambda \\
\alpha
\end{bmatrix}
= \begin{bmatrix} A^L & P^L \end{bmatrix}
\begin{bmatrix} \lambda \\ \alpha \end{bmatrix},
\]

(4.5)

where \( A^L \) and \( P^L \) are given by (1.3) and (4.7), respectively. Combining expressions (4.4) and (4.5) gives

\[
\begin{bmatrix}
\lambda \\
\alpha
\end{bmatrix}
= \begin{bmatrix} A^L & P^L \end{bmatrix}
\begin{bmatrix} A P \\ P^T 0 \end{bmatrix}^{-1}
\begin{bmatrix} I_{N+1} \\ 0 \end{bmatrix}
\begin{bmatrix} I_{N+1} \\ 0 \end{bmatrix}
\begin{bmatrix}
\lambda \\
\alpha
\end{bmatrix},
\]

where \( I_{N+1} \) is the identity matrix of order \( N+1 \). An RBF discretization of the operator \( \mathcal{L} \) can then be written as

\[
\begin{bmatrix} A^L & P^L \end{bmatrix}
\begin{bmatrix} A P \\ P^T 0 \end{bmatrix}^{-1}
\begin{bmatrix} I_{N+1} \\ 0 \end{bmatrix}
\begin{bmatrix} I_{N+1} \\ 0 \end{bmatrix}.
\]

(4.6)

**Theorem 4.1** Assume that

\[
\sum_{i=0}^{N} \sum_{j=0}^{N} \bar{\lambda}_i \mathcal{L} p_j(x)|_{x=x_i} = 0.
\]

(4.7)

The matrix \( L \) derived above for a conditionally positive definite function \( \phi \) has a purely imaginary spectrum if \( A^L \) is antisymmetric or a purely real spectrum if \( A^L \) is symmetric.

**Proof.** Suppose that \( \nu \) is an eigenvalue of \( L \) with eigenvector \( \mathbf{u} \). Then we have that

\[
L \mathbf{u} = \nu \mathbf{u} \iff A^L \lambda + P^L \alpha = \nu (A \lambda + P \alpha) \quad \text{and} \quad P^T \lambda = 0.
\]

Notice that \( \lambda^* P = 0 \) from (1.3) and \( \lambda^* P^L = 0 \) because of (4.7), where \( * \) denotes the complex conjugate transpose. We also have that (1.4) gives \( \lambda^* A \lambda > 0 \). Therefore, we obtain

\[
\nu = \frac{\lambda^* A^L \lambda}{\lambda^* A \lambda},
\]

66
and since $A$ is symmetric,

$$\bar{\nu} = \frac{\lambda^*(A^T_{L^T})\lambda}{\lambda^*A\lambda}.$$  

Thus if $A_{L^T}$ is symmetric, we have that $\nu = \bar{\nu}$, and if it is antisymmetric, $\nu = -\bar{\nu}$. □

A few remarks should be made about this theorem:

- The assumption (4.7) is automatically satisfied for any linear differential operator $\mathcal{L}$ if $\phi$ is a positive definite RBFs (e.g. Gaussians and inverse quadratics) or conditionally positive definite of order 1 (e.g. multiquadrics). Moreover, If $\mathcal{L}$ has constant coefficients then $\mathcal{L}p_j(x)$ is also a polynomial and (4.7) holds for any conditionally positive RBF as long as the coefficients $\lambda_j$ satisfy (1.3).

- The matrix of derivatives $\frac{d}{dx}\phi(\|x - x_j\|)\bigg|_{x=x_i}$ is antisymmetric. Hence, the differentiation matrix $\tilde{D}$ has only imaginary eigenvalues. Similar statement can be made about the matrix of partial derivatives.

- For conditionally positive definite RBFs, therefore, deviations of the spectrum from the imaginary axis occur when boundary conditions are enforced in $\tilde{D}$ to generate $D$.

- RBF methods for differential equations on the unit circle or unit sphere are boundary-condition free, making RBF methods suitable for time-dependent problems on such regions.

4.2 Periodic boundary conditions

Differential equations with periodic boundary conditions on an interval can be naturally mapped to a boundary-condition-free problem on the unit circle. For instance, solving $u_t = u_x$ with periodic boundary conditions in $[0, 2\pi]$ is equivalent to solving $u_t = u_\theta$ on the unit circle, where $\theta$ is the polar angle. Considering the
norm \( \|x_i - x_j\| = \sqrt{2 - 2 \cos(\theta_i - \theta_j)} \), one can easily show, in light of the observations above, that the RBF differentiation matrix in this case has only imaginary eigenvalues.

As a practical test, we consider the KDV equation

\[
    u_t + uu_x + u_{xxx} = 0. \tag{4.8}
\]

Among the solutions admitted by (4.8) are traveling waves called solitons. Figure 4.3 shows a numerical solution to the KDV equation with initial condition

\[
    u(0, x) = 1875 \text{sech}^2\left(\frac{25(x + 2)}{2}\right) + 768 \text{sech}^2\left(\frac{4(x + 1)}{2}\right),
\]

which is a superposition of two solitons. This problem was solved in [68] with Fourier spectral methods. Here we used Gaussian RBFs with 200 equally spaced centers and nodes and \( c = 0.09 \). We can use periodic boundary conditions, since the solution decays to zero exponentially in the domain of consideration. Fourth-order Runge-Kutta with \( \Delta t = 3 \times 10^{-6} \) was used to discretize the time derivative. A small time-step is needed in this problem due to the large spectral radius of the third-order differentiation matrix. In Table 4.1 we compare the error at \( t = 6 \times 10^{-3} \) of the approximations obtained with GRBFs and the Fourier spectral [68, Program 27]. To obtain the data presented in this table \( \Delta t = 0.5 \times 10^{-7} \) was used, so the dominant error is due to the spacial approximations. Notice that for both 200 and 256 degrees of freedom the Fourier method was more accurate. We point out the GRBF method was limited to small \( c \) to avoid ill-conditioning – larger values of \( c \) would probably give more accurate results. It is also important to mention that no attempt to use an adaptive resolution was made, which could improve the performance of the RBF method significantly.

It is worth pointing out that the spectrum of \( \tilde{D} \) is sensitive to perturbations on the nodes, i.e., small deviations of the set of nodes from the set of centers is likely to generate unstable eigenvalues. In Figure 4.4 the spectra of two differentiation
Figure 4.3: Numerical solution of the KDV equation — cf. [68, Output 27].

Table 4.1: Relative error \( \| u_{\text{approx}} - u_{\text{exact}} \|_{\infty}/\| u_{\text{exact}} \|_{\infty} \) in the computation of the solution of the KDV equation at \( t = 6 \times 10^{-3} \).
matrices are shown for the periodic transport problem on the unit circle. Both matrices were obtained with equally spaced centers and nodes, GRBFs, $N = 13$, and $c = 1$. In the first case we considered centers $\theta_j^c = 2\pi j/14, j = 0, ..., 13$, and nodes $\theta_j = \theta_j^c$. Imaginary spectrum is guaranteed in this case and the numerical results agree with this prediction. In the second case we used $\theta_j = \theta_j^c + 0.01$. The matrix obtained in this instance has real eigenvalues as shown in the figure.

Problems on the unit sphere are also boundary-free. As an example, consider the convective test problem presented in [28],

$$u_t + (\cos \alpha - \tan \theta \sin \varphi \sin \alpha) u_\varphi - (\cos \varphi \sin \alpha) u_\theta = 0, \quad (4.9)$$

where $\alpha$ is constant and the spherical coordinates are defined by $x = \cos \theta \cos \varphi$, $y = \cos \theta \sin \varphi$, and $z = \sin \varphi$. This equation describes a solid-body rotation in an arbitrary direction, determined by the parameter $\alpha$ (see [28] for details).
We again consider the Euclidean metric, so for nodes $x_j$ on the sphere,
\[
\|x_i - x_j\|^2 = 2 - 2(\cos \theta_i \cos \theta_j \cos (\varphi_i - \varphi_j) + \sin \theta_i \sin \theta_j).
\]

To demonstrate that explicit time integrators with a conditionally positive definite RBF of order at most one can be stably used for this problem, all is needed is to show that $A^L$ for the operator $L(\cdot) = -(\cos \alpha - \tan \theta \sin \varphi \sin \alpha)\partial_\varphi(\cdot) + (\cos \varphi \sin \alpha)\partial_\theta(\cdot)$ is antisymmetric. Straightforward calculations show that
\[
A^L_{i,j} = -2 \frac{\partial \delta(x)}{\partial x} |_{r^2=\|x_i - x_j\|^2} \left[ \cos \alpha \cos \theta_i \cos \theta_j \sin (\varphi_i - \varphi_j) 
+ \sin \alpha (\cos \theta_i \sin \theta_j \cos \varphi_i - \cos \theta_j \sin \theta_i \cos \varphi_j) \right],
\]
which is indeed antisymmetric.

Figure 4.5 shows a solid-body rotation on the unit sphere governed by (4.9) with $\alpha = 20^\circ$. The initial condition in this simulation is
\[
g(x, y, z) = \exp \left( -5 \left( x^2 + y^2 + (z - 1)^2 \right)^2 \right),
\]
for $(x, y, z)$ on the unit sphere. The solution was computed with Gaussians with $c = 0.4$ and 700 nodes. The node distribution was not uniform. The points used in the computations are shown in the first plot. Because the solution is nearly zero in the lower hemisphere for all times, we used more nodes in the north hemisphere than in the south. This illustrate an advantage that RBFs have over other spectral schemes on the sphere – the potential for adaptive implementations. Forth-order Runge-Kutta with $\Delta t = 0.02$ was used in time. The maximum error after one cycle ($t = 2\pi$) is 0.0075.

In the remainder of this chapter, we shall explore ways to stabilize RBF methods for time-dependent problems with non-periodic boundary conditions. We shall next study how the interpolation nodes can be used to avoid unstable eigenvalues.
Figure 4.5: Numerical solution of a rigid-body rotation on the unit sphere computed with Gaussians. Nodes used in the approximations are shown at $t = 0$. The $L_\infty$ norm of the error at $t = 2\pi$ is 0.0075.
4.3 Stable nodes for Gaussian RBFs

In Chapter 2, we have shown that for GRBFs with equally spaced centers in \([-1, 1]\), stable interpolation requires specific node placement. It is reasonable to think that the same node distributions can be used to stabilize GRBFs for time-dependent. Recall that as in polynomial interpolation, although convergence may be guaranteed for sufficiently smooth functions for a given set of interpolation nodes, approximations may not converge in the presence of rounding errors due to the rapid growth of the Lebesgue constant [70]. For GRBFs, we found that under most conditions these constants grow exponentially with \(N\). If nodes are obtained with optimal density functions, however, the growth of these constants seems to be logarithmic.

To illustrate how optimal nodes stabilize RBF approximations of time-dependent problems, in Figure 4.6 we show the behavior of the eigenvalues of the RBF differential matrix \(D\) with incorporated boundary conditions. The matrix \(D\) was derived with the barycentric algorithm of Section 2.5 and \(\beta = 2\). Now all eigenvalues lie in left-half plane, so standard explicit time integration techniques can be used together with this spatial approximation.

The spectral radius of \(D\) is shown in Figure 4.7 as a function of \(N\) for \(\beta = 0.1, 1,\) and 10. Notice that for large \(N\), the spectral radius grows as \(O(N^2)\). We observe that for \(\beta = 10\), the growth was \(O(N)\) for smaller values of \(N\), but for large \(N\) the effects of the rapid growth of the density function near the boundary (see Figure 2.8) forces clustering of the nodes. The spacing between nodes for large \(N\) near the ends of the interval seems to be approximately \(O(1/N^2)\). The restriction on time-step sizes for GRBFs on stable nodes, is therefore, similar to the one for polynomial approximation [24, 68].

The results in this section extend immediately to tensor-product regions of uniform center locations in higher dimensions. Although this type of region is usually
**Figure 4.6:** Eigenvalues of the GRBF differentiation matrix for $\beta = 2$ and stable nodes: $N=30$ (●); $N=50$ (○); and $N=70$ (★).

**Figure 4.7:** Spectral radius of the GRBF differentiation matrix as a function of $N$. For large values of $N$, the spectral radius grows as $O(N^2)$. 
of little interest to RBF users since they are within the scope of polynomials, they help us to illustrate the fact that the location of collocation nodes can be used to stabilize RBFs and improve accuracy. Figure 4.8 presents results for the convective test problem, \( u_t = u_x + u_y \), for \((x, y) \in [-1, 1] \times [-1, 1]\) and \( t > 0 \), with initial condition \( u(0, x, y) = \exp(-20((x - 0.2)^2 + (y - 0.2)^2)) \) and boundary conditions \( u(t, 1, y) = u(t, x, 1) = 0 \). We used GRBFs with 27 nodes, \( \beta = 2 \), and equally spaced centers. We can generate stable nodes in this square by taking the tensor product of stable nodes in \([-1, 1]\) (Figure 4.8(a)). In Figure 4.8(b) we show rescaled eigenvalues of the RBF convection matrix when \( \Delta t = 0.04 \). Notice that for stable nodes, they lie inside the fourth-order Runge-Kutta region of stability; for equispaced nodes, however, some lie outside. Figure 4.8(d) shows the computed solution with clustered nodes at \( t = 0.7 \) using fourth order Runge-Kutta.

Although asymptotically stable nodes for other radial functions, like multi-quadrics and inverse quadratics, are not known, Figure 4.9 indicates that clustering of nodes may also be used to stabilize discretizations obtained with these functions. This figure presents the spectrum of \( D \) obtained with multiquadrics, shape parameter \( c = 1 \), equally spaced centers, \( N = 19 \), and three sets of interpolation nodes. Notice that for Chebyshev and equispaced nodes the differentiation matrices present unstable eigenvalues, but for GRBF stable nodes for \( \beta = 1 \), \( D \) has only eigenvalues with non-positive real part.

### 4.4 GRBFs and mapped polynomial methods

Since the early 1990s attention has been given to mapped polynomial methods, such as the one introduced by Kosloff and Tal-Ezer [44]. The interpolant for mapped polynomial methods takes the form

\[
F(x) = \sum_{k=0}^{N} \lambda_k P_k(y),
\]
Figure 4.8: A 2-D test problem: (a) stable nodes; (b) Runge-Kutta stability region, scaled eigenvalues for stable nodes (●) and for equispaced nodes (×); (c) initial condition; (d) numerical solution at $t = 0.7$. 
Figure 4.9: Spectra of differentiation matrices generated with multiquadrics and three sets of interpolation nodes: equally spaced (○); Chebyshev (+); and GRBF stable nodes (*).

where $P_k$ form a polynomial basis and $y$ is some function of $x$ mapping $[-1, 1]$ to itself. The mapping proposed in [44] is

$$y = \frac{1}{\xi} \sin(x \sin^{-1} \xi), \quad 0 < \xi < 1.$$  

The stable interpolation nodes are thus given by Chebyshev nodes [68] in the variable $y$. The parameter $\xi$ controls the degree to which the grid is stretched under the mapping.

Different strategies to choose the parameter $\xi$ have been proposed in the literature (see [52] and references therein). The goal is to allow near-spectral convergence in space with a time-step restriction of $O(N^{-1})$. To achieve this objective one has to take $\xi$ close to 1 as $N$ is increased; effectively one makes an explicit tradeoff between accuracy and stability.

According to (2.3), GRBFs with equally spaced centers can be seen as a mapped polynomial method. We believe that in most cases, given $\beta$ one can find $\xi$ so that both methods present similar convergence and stability properties. Figure 4.10
shows limiting node density functions for the Kosloff-Tal-Ezer method. Comparing this with Figure 2.8, we see similar clustering behavior of the density functions near ±1.

4.5 Least-squares approximations

Least-squares approximations were investigated in Chapter 3 as an alternative to interpolation-based methods. The main motivation to use least-squares algorithms is that they do not require a specific node distribution; although clustering near the boundary is helpful. In this section we explore the stability of the method of lines when the differentiation matrices derived in Section 3.4 are used.

Figure 4.11 presents the eigenvalues of the least-squares differentiation matrix for multiquadrics. To generate this data we used twice as many nodes as centers. All numerical results presented in this section were obtained with Chebyshev nodes $(x_j = \cos(\pi j/(2N)), j = 0, 1, ..., 2N)$. This choice of nodes is not required for stability, and in several numerical tests equally spaced nodes were successfully used.
We emphasize the square collocation approach with either node distributions is most likely to be unstable. In Figure 4.11(a), we used shape parameter \( c = 1 \) and equally spaced centers. It can be observed in this figure that if \( N \) is increased, most the spectrum moves further to the left of the imaginary axis. In Figure 4.11(b) we fixed \( N = 13 \) and varied the shape parameter \( c \). All eigenvalues presented in this plot allow stable explicit time integration. For instance, one could use fourth order Runge-Kutta in time with \( \Delta t = 0.13 \) if \( N = 13 \) and \( c = 1 \).

The standard representations of smooth RBFs subspaces are in most circumstances ill-conditioned. This would make continuation of Figure 3.4 to, say, \( 10^{-10} \) virtually impossible in double precision. For GRBFs with equally spaced centers, however, well-conditioned representations can be computed. In Section 3.3 we presented an algorithm based on an Arnoldi-like iteration to generate orthogonal basis for GRBFs with equally spaced centers. This approach also relies on a close connection between GRBFs and polynomials. In Figure 4.12 the error of the approximations of the derivative of \( f(x) = \sin(\pi x) \) using GRBFs with \( \beta = 2 \) is presented.
The least-squares differentiation matrix was used for these approximations. Notice that the Arnoldi iteration permits approximations close to machine precision, while convergence for the standard GRBF basis stops with error of $O(10^{-6})$. Figure 4.13 shows the eigenvalues of the least-square GRBF differentiation matrix for $\beta = 2$ and $N = 30, 50,$ and $70$.

In Figure 4.14 we consider the numerical solution of the transport equation with initial condition $u_0(x) = \exp(-(5x - 3.5)^{10})$. In Figure 4.14(a) the exact solution is presented for $t = 1$ together with two numerical solutions obtained with GRBFs and 20 centers. Basis functions were computed with the Arnoldi iteration with $\beta = 1$. It can be observed that the least-squares method gives slightly better results. For the least-squares approximations we used twice as many Chebyshev nodes as centers. Figure 4.14(b) shows the maximum error, $\max |u_{\text{exact}}(t, x) - u_{\text{approx}}(t, x)|$, $(t, x) \in [0, 1] \times [-1, 1]$, for several values of $N$. In this instance the least-squares method and the collocation method on stable GRBF nodes presented
similar rates of convergence and errors. Fourth-order Runge-Kutta was used for
time-integration with time-step $\Delta t = 10^{-3}$.

To illustrate the least-squares scheme on 2-D regions, we solve the wave equa-
tion $u_{tt} = u_{xx} + u_{yy}$ with zero Dirichlet boundary conditions in a peanut-like region
defined by parametric equations $x = \sqrt{\cos^2 \theta + 4 \sin^2 \theta} \cos \theta$, $y = \sqrt{\cos^2 \theta + 4 \sin^2 \theta} \sin \theta$, $0 \leq \theta < 2\pi$. Figure 4.15 presents the eigenvalues of the Laplacian operator dis-
cretized with multiquadrics, shape parameter $c = 1$, and 244 uniformly spaced
centers. Notice that if the collocation method is used with nodes that coincide with
centers, we obtain a matrix that has complex eigenvalues (Figure 4.15(a)). Nor is
there a clear way to generalize the stable 1-D GRBF nodes to this region. Using
the least-squares method with 515 uniformly spaced nodes, on the other hand, one
obtains a matrix with almost purely real spectrum, as expected for the Laplacian,
and smaller spectral radius.

Figure 4.16 presents the numerical solution of the wave equation with ini-
tial conditions $u(0, x, y) = \exp(-30(x^2 + (y - 1)^2)) + \exp(-50(x^4 + (y + 1)^4))$, and
Figure 4.14: (a) Solution of the transport problem at \( t = 1 \) using \( N = 20 \) centers: exact solution (dotted line); least-squares method (solid line); and collocation method on GRBF nodes (dashed line). (b) maximum error for several values of \( N \) obtained with the collocation method on GRBF nodes (\( \circ \)) and the least-squares method (\( * \)).

Figure 4.15: Eigenvalues of the Laplacian operator on the peanut region discretized with multiquadrics: (a) collocation method on uniformly spaced nodes; (b) the least-squares method.
\( u_t(0, x, y) = 0 \). The least-squares method was used together with a leapfrog discretization in time with time step \( \Delta t = 0.005 \). Notice that this time-discretization scheme for the second order derivative requires a purely real spectrum for stability. A fine grid was used to plot the solution at \( t = 0, 0.33, 0.66, 1 \). The maximum norm of the error at \( t = 1 \) is approximately 0.005.

### 4.6 Summary

In this chapter, we explored the eigenvalue stability of RBF discretizations for time-dependent problems. We used the method of lines with RBFs approximations in space. We proved that RBF methods are eigenvalue stable for problems on periodic domains, when conditionally positive definite RBFs are used. Following this, we showed that GRBF collocation is stable when nodes are placed according with specific distributions derived in Section 2.3.3. This gives a conclusive demonstration that node locations can eliminate instability. We also mentioned that GRBFs in 1-D with equally spaced centers have features similar to the mapped polynomial method introduced by Kosloff and Tal-Ezer in [44]. For problems in higher dimensions and other shape functions, we proposed using differentiation matrices based on least squares. This offers flexibility on node placement and since this formulation does not impose restrictions on the location of centers, it can be used to adapt centers to data. We also used the Arnoldi iteration for GRBFs introduced in Section 3.3 to circumvent conditioning issues.
Figure 4.16: Numerical solution of a vibrating peanut-shaped membrane using multiquadrics. Estimated $L_{\infty}$ norm of the error at $t = 1$ is 0.05.
Chapter 5

EIGENVALUE PROBLEMS

Eigenvalue problems arise in many physical applications, such as in the study of vibrations and acoustic scattering. They have been extensively investigated both numerically and theoretically. The computation of eigenmodes of the Laplacian operator, in particular, has attracted the attention of many researchers — see e.g. [16, 17, 31, 46]. The diversity of areas of application creates a wide range of eigenvalue problems defined on domains with complicated boundary shape, domains with corners, and boundary points where one type of boundary is changed to another [80].

In this chapter we study RBF-based methods for eigenvalue problems. As expected, most features experienced by the use of RBFs in boundary value problems carry over for eigenvalue problems. We explore collocation methods and techniques that have been proposed in [25] and [23] to avoid degradation of the solution near the boundaries. For regions with corner singularities, we append terms to the RBF expansion that capture the singular behavior of an eigenfunction to maintain good convergence.

As an alternative to square collocation methods, we also explore least-squares formulations. Eigenmodes are obtained using the differentiation matrices derived in Section 3.4. It is shown that eigenvalues computed with a certain least-squares formulation are the local maximum of the pseudoresolvent of a rectangular pencil. This allows us to use a method based on the computation pseudospectra of rectangular matrices.
We carry out several experiments in 2-D regions for the Laplacian operator. We consider a smooth domain (a disk), a region with corners (a rhombus), and a region with a reentrant corner (an L-shaped region). These domains are pictured in Figure 5.1.

5.1 Computing eigenmodes of the Laplacian

Given a linear elliptic second order partial differential operator \( \mathcal{L} \) and a bounded region \( \Omega \) in \( \mathbb{R}^n \) with boundary \( \partial \Omega \), we seek eigenpairs \( (\nu, u) \in (C, C(\overline{\Omega})) \) satisfying

\[
\mathcal{L}u + \nu u = 0 \quad \text{in } \Omega \quad \text{and} \quad \mathcal{L}_B u = 0 \quad \text{on } \partial \Omega,
\]

where \( \mathcal{L}_B \) is a linear boundary operator of the form

\[
\mathcal{L}_B u = au + b(n \cdot \nabla u).
\]

Here \( a \) and \( b \) are given constants and \( n \) is the unit outward normal vector defined on the boundary. We assume that \( \Omega \) is open and that the eigenvalue problem is well-posed. We are particularly interested in the case where \( \mathcal{L} \) is the Laplacian operator, but the techniques presented here can be used for other linear operators.

In section 4.1.2 we derived an RBF discretization of the operator \( \mathcal{L} \), namely,

\[
L = \begin{bmatrix} A^L & P^L \\ P^T & 0 \end{bmatrix} \begin{bmatrix} A & P \\ P^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} I_{N+1} \\ 0 \end{bmatrix}.
\]
For the more general boundary condition in (5.1) \( L \) can be written as

\[
L = \begin{bmatrix} A_I^T & P_I^T \end{bmatrix} \begin{bmatrix} A_I & P_I \\ A_B & P_B \\ P^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} I_{N_I} \\ 0_{N_B+1\times N_I} \end{bmatrix}, \tag{5.3}
\]

where the matrices \( A_I \) and \( P_I \) contain the elements of RBF and polynomial basis functions corresponding to interior nodes, \( A_I^T \) and \( P_I^T \) are restrictions of \( A^L \) and \( P^L \) to interior nodes, and the elements of \( A_B \) and \( P_B \) are \( \mathcal{L}_B \phi(|x - x_j|)|_{x=x_i} \) and \( \mathcal{L}_B p_j(x)|_{x=x_i} \) with \( x_i \) on \( \partial \Omega \). Here \( N_I \) is the number of interior nodes and \( N_B \) is the number of nodes on the boundary. Notice that \( L \) is an \( N_I \times N_I \) matrix.

We computed the eigenmodes of a disk of radius \( \pi \) with Dirichlet and Neumann boundary conditions. We used multiquadric RBFs \( (\phi(r) = \sqrt{c^2 + r^2}) \) to generate the results and considered the expansion

\[
F(x) = \sum_{j=0}^{N} \lambda_j \phi(\|x - x_j\|) + \alpha_0. \tag{5.4}
\]

Results are presented for the shape parameter \( c = 3 \). Figure 5.2 shows a computed eigenfunction of the disk. We used 177 uniformly distributed nodes to approximated the solution. The computed eigenvalue was 57.58637, while the true eigenvalue corresponding to this mode is 57.58294.

Denote the exact \( j \)th eigenvalue, including multiplicities, by \( \nu^{(j)} \), \( 0 < \nu^{(1)} \leq \nu^{(2)} \leq \ldots \) (we exclude the zero Neumann eigenvalue, which has been computed to at least 11 accurate places in our experiments). For an RBF-computed eigenvalue we write \( \nu_N^{(j)} \) and define the mesh size by

\[
h \equiv \max_{i=1, \ldots, N} \left\{ \min_{j=1, \ldots, N} \|x_i - x_j\|_2 \right\}. \]

Figure 5.3 shows the relative error, \( |\nu^{(j)} - \nu_N^{(j)}|/\nu^{(j)} \), in selected eigenvalues of the disk as function of \( 2\pi/h \). For the Dirichlet problem, the 3rd, 5th, 8th and 12th
Eigenfunction corresponding to $\nu = 57.58294$.

Figure 5.2: Numerically computed eigenfunction (left) and 177 nodes used for collocation (right).

are repeated eigenvalues and the errors in these are not shown. Similarly, for the Neumann problem we only show the error in distinct eigenvalues. We note that the lower eigenvalues are usually resolved more accurately than higher ones for the Dirichlet problem, but not for the Neumann problem where in most cases the 5th eigenvalue was most accurate among the first 12. As expected, the error shows exponential decay.

5.2 Boundary treatment

As suggested in the literature [23, 25, 47], RBF approximations are, in general, inaccurate near the boundary. In light of the discussion in Chapter 2, we can relate this difficulty to a Runge phenomenon in 1-D. In this section, we consider two techniques for improving accuracy near the boundary that have been suggested in the literature.
5.2.1 Boundary clustering of nodes

In chapter 2 we found that clustering of nodes near the boundary is essential for good approximations in 1-D (theoretically proved for Gaussians with equally spaced centers, and experimentally for other RBFs). We shall investigate here the benefits of clustering in 2-D regions. In [25] numerical experiments were run based on nodes in \([-1, 1]\) with density proportional to \((1 - x^2)^{-\gamma}\). The case \(\gamma = 0\) gives equispaced nodes, \(\gamma > 0\) clusters nodes toward the endpoints, and \(\gamma = 0.5\) gives the Chebyshev points; see [24] for more detail.

Figure 5.4 shows node distributions obtained with \(\gamma = 0.4\). On top, the nodes on a 1-D interval are shown. Using these points, it is possible to generate nodes in 2-D regions as presented for the disk, triangle, and L-shaped region.

5.2.2 Collocation of the PDE on the boundary

The idea of adding equations obtained via collocation of the PDE on the boundary was introduced for boundary-value problems in [23]. We shall explore this approach in connection with the eigenvalue problem.
In order to add equations, an extra set of centers is needed. We add $N_B$ centers outside the domain and adjacent to the boundary, so that $N = N_I + 2N_B$.

In Figure 5.5, we show uniformly distributed nodes for the disk: the dots represent collocation nodes and the open circles represent centers.

The implementation of this scheme is straightforward. One just needs to consider the boundary collocation nodes as both boundary points and interior points (in the sense that both equations in (5.1) should be satisfied). The resulting $L$ matrix has structure similar to that presented previously.

### 5.2.3 Numerical results

In order to compare the boundary treatment techniques for the test problem described in section 5.1, we adopted the following abbreviations: BC for boundary clustering of nodes and PDECB for collocation of the PDE on the boundary. Clustering of nodes was obtained using $\gamma = 0.4$. 

\[ \text{Figure 5.4: Boundary clustering.} \]
Figure 5.5: Example of distribution of centers (circles) and collocation nodes (dots) in a disk for collocation of the PDE on the boundary.

Figure 5.6 presents the maximum relative error over the 12 first eigenvalues, \( \max_{1 \leq j \leq 12} |\nu^{(j)} - \nu^{(j)}_N|/\nu^{(j)} \), for several values of \( N \). The spectral convergence can be verified in the graphs of the error versus \( 2\pi/h \) (Fig. 5.6 left). We also plotted the error versus \( N \) (Fig. 5.6 right) since the work needed to computed the solution depends on \( N \).

Both BC and PDECB improved the solution, BC giving the most accurate results of the two schemes. Combining the two techniques did not improve the solution over BC alone for the Dirichlet problem. For the Neumann problem, the combination resulted in better results. Comparing the results for the Dirichlet problem and for the Neumann problem, we note that the results for the Dirichlet problem are, in most cases, more accurate than for the equivalent Neumann problem. We believe that resolution near the boundary plays a more important role for the Neumann problem since the derivatives of the expansions are needed at the boundary.

In Figure 5.7 we compare the performance of the RBF method to a spectral method and second-order finite elements. For the spectral scheme we used the algorithm described in [68], which uses Chebyshev-pseudospectral approximations in the radial direction and Fourier-pseudospectral approximations in the angular component. The finite element results were obtained with the PDE Toolbox for MATLAB. We used a 400MHz Linux PC to generate this data and all algorithms
Dirichlet boundary condition

Neumann boundary condition

Figure 5.6: Maximum relative error over the first 12 eigenvalues of the disk.
Figure 5.7: Maximum relative error over the first 12 eigenvalues versus CPU time for the disk with Dirichlet boundary condition.

were implemented in MATLAB. Although the RBF method did not converge as fast as the spectral method (which is able to exploit the special structure of the disk), its performance was significantly better than the low-order finite element method.

5.3 Corner singularities

We now turn our attention to the eigenvalue problem in 2-D regions with corners. In a corner with interior angle $\pi/\varphi$, an eigenfunction has a particular singular behavior if $\varphi$ is not integer. For the L-shaped region, for instance, a singularity occurs at the reentrant corner. Generic numerical procedures are inefficient because of the loss of smoothness at such corners. In particular, a typical RBF approximation gives a very poor approximation of an eigenfunction near a singularity. Clustering of points at the reentrant corner and collocation of the PDE on the boundary did not prevent poor convergence rates in our numerical experiments.

In order to improve our numerical scheme, we exploit the local behavior of the eigenfunction at a corner of the boundary. Let $(r, \theta)$ be polar coordinates originating
An eigenfunction \( u \) of \((5.1)\) with Dirichlet boundary condition can be represented as

\[
u(r, \theta) = \sum_{n=1}^{\infty} c_n J_{\nu n}(\sqrt{\nu r}) \sin(n\varphi \theta), \tag{5.5}
\]

where \( J_{\nu} \) is a Bessel function of first kind.

This information was used by Fox, Henrici, and Moler [31] to compute the eigenvalues of the L-shaped region by truncating \((5.5)\) and requiring the trial solution to be zero at collocation points along the boundary. The method then consists in finding the zeros of the determinant of the resulting matrix, which depends non-linearly on \( \nu \). However, as pointed out in [17], as the number of terms in the truncated series is increased the matrix becomes nearly singular and obtaining the true eigenvalues becomes numerically impossible. The algorithm used by Driscoll in [17] uses domain decomposition and treats the corner expansion locally rather than globally, avoiding this difficulty but making the implementation more complex. Recently, Betcke and Trefethen [5] propose a modification of the method of Fox, Henrici, and Moler to avoid the conditioning issue. The crucial changes are to introduce points in the interior of the region as well as on the boundary and to minimize a subspace angle rather than just the determinant. The algorithm, however, still require a
nonlinear minimization for each eigenmode computed, which make it less efficient if several eigenpairs need to be found.

The modification in the RBF-method we propose here is easy to implement and yet avoids degradation of the solution at singular corners. We consider the ascending series

\[ J_\nu(z) = \left( \frac{1}{2}z \right)^\nu \sum_{k=0}^{\infty} \frac{(-\frac{1}{4}z^2)^k}{k!\Gamma(\nu + k + 1)}, \]

so we have that (5.5) becomes

\[ u(r, \theta) = \sum_{n=1}^{\infty} \sum_{k=0}^{\infty} b_{nk}r^{n\varphi + 2k} \sin(n\varphi \theta), \]

where the coefficients \( b_{nk} \) are independent of \( r \) and \( \theta \).

Motivated by the last expansion, we add terms of the form \( r^{n\varphi + 2k} \sin(n\varphi \theta) \) to the RBF approximation, i.e.,

\[ F(x) = \sum_{j=0}^{N} \lambda_j \phi(\|x - x_j\|) + \sum_{n,k} \alpha_{n,k}r^{n\varphi + 2k} \sin(n\varphi \theta), \]

where \( n \) and \( k \) are chosen appropriately so that the smallest powers of \( r \) are used. We believe that these extra terms help to capture the singular behavior of an eigenfunction at a corner. For the Neumann problem, we have that

\[ u(r, \theta) = \sum_{n=0}^{\infty} \sum_{k=0}^{\infty} b_{nk}r^{n\varphi + 2k} \cos(n\varphi \theta), \]

which suggests a similar modification. We also think that similar singular term augmentation could be applied to RBF approximations of singular boundary value problems and problems in which the nature of the boundary condition changes abruptly, when the expansion of the singular part of the solution is known a priori [79].

We considered two regions with corner singularities in our numerical experiments: the L-shaped region of vertices \((0, \pi/2), (\pi/2, \pi/2), (\pi/2, 0), (\pi, 0), (\pi, \pi)\)
and \((0, \pi)\), and the rhombus with supplementary angles 60° and 120° and sides of length \(\pi\). We also used multiquadric RBFs to generate the results, with \(c = 0.6\) for the L-shaped region and \(c = 1.2\) for the rhombus.

The maximum relative error over the first 12 eigenvalues of the L-shaped region is presented in Figure 5.9 for several values of \(N\) and \(M\), where \(M\) is the number of singular terms appended to the RBF expansion. We note that the plain RBF expansion (1.2) does not converge and that clustering nodes near the singularity worsens the approximations for both Dirichlet and Neumann boundary conditions. Adding terms that exploit the reentrant corner information is essential for good approximations. In most cases, when only a few corner terms were used (1 to 20 terms) the rate of convergence appeared to be algebraic. Moreover, augmentation by only one corner term was not enough to avoid divergence when clustered nodes were used. The best results were obtained when the number of corner terms used was proportional to the number of nodes. In Fig. 5.9 we present results for \(M \approx N/3\).

Figure 5.10 shows the contour plot of the 20th eigenfunction of the L-shaped region. The graph on the left presents the 20 contour lines obtained with the PDE Toolbox. The graph on the right is the RBF approximation generated with 279 clustered nodes and 103 corner terms. The finite element approximation was obtained with 4064 triangles. To compute the first 20 eigenmodes, the finite element scheme took approximately 11 seconds and the RBF method 3.8 seconds. The 20th eigenvalue was computed to 3 accurate digits with RBFs and to 2 accurate digits with the finite element method.

Our final test region is the rhombus. This region has two singular corners with angle \(2\pi/3\). For the Dirichlet problem, the eigenfunctions that are antisymmetric (about the shorter diagonal) are smooth since they be constructed using the modes of a equilateral triangle [51]. Other eigenfunctions present singular behavior, as it is the case of the first mode. Although this singularity is not as strong as it is
Figure 5.9: Maximum relative error over the first 12 eigenvalues of the L-shaped region with and without corner terms augmentation.
Finite Element (4064 DOF)  

RBF (382 DOF)

Figure 5.10: Contour plot of the 20th eigenfunction of the L-shaped region with Neumann boundary condition (20 contour lines).

in a reentrant corner, it still results in poor convergence rates for global methods that use smooth functions to approximate the solution. In fact, in [78] an algebraic convergence rate is predicted when RBFs are used to interpolate such modes. Our results show that this behavior can be avoided by singular corner augmentation.

Figure 5.11 shows the error in the first eigenfunction of the rhombus. The errors presented range from 0 (white) to the maximum absolute error (black) indicated below each plot. One can see that the error is larger at the singular corners if the standard RBF expansion is used (left). When corner terms are added (center), the error decreases in magnitude and becomes more evenly distributed but it is still larger near the boundary. When one uses corner terms and clusters nodes more densely near the boundary (right), the error becomes evenly distributed in the entire domain.

Figure 5.12 presents the relative error of the first eigenvalue of the rhombus
for several values of $N$ and $M$. Although convergence can still be achieved without singular terms, the best results were obtained with corner-terms augmentation. We used terms from both corners evenly. In this case, which has a less severe loss of regularity than the L, adding few terms (2 to 20 terms) was enough to obtain good convergence rates.

5.4 Overdetermined formulations

While collocation of the PDE on the boundary and ad-hoc node clustering improve the accuracy of RBF collocation methods, approximations through these techniques are not optimal in general. In an attempt to further improve accuracy we again consider best approximations with respect to the discrete $L_2$ norm.

5.4.1 Least-squares differentiation matrices

In chapter 3 we derived matrices that are discretizations of linear differential operators. These matrices were derived so that the residual (3.4) is minimized. The spectra of such matrices was studied to some extent in Chapter 4 in the context of eigenvalue stability for discretizations of time-dependent problems. In this section
we briefly investigate how accurately eigenmodes of these matrices approximate the ones of the Laplacian operator.

In Figure 5.13 we compare approximations obtained with the square collocation method used in the previous section with the least-squares method – which consists of computing the eigenpairs of the least-squares matrix derived in Section 3.4 for the Laplacian. We obtain the eigenvalues of the disk when boundary clustering of nodes is used. Notice that square collocation gives eigenvalues with nonzero imaginary part, which are not physical. The least-squares method, on the other hand, gives only real eigenvalues, which is a desirable feature when time-dependent problems are concerned. Regarding accuracy, both methods present similar performance. The aspect ratio of the least-square matrices was kept at approximately 2. Multiquadrics with shape parameter \( c = 3 \) were used in the computations.

Figure 5.14 was generated with the same data as in Figure 5.13, except for the shape parameter; this figure was obtained with \( c = 1 \). We again notice that the eigenvalues of the least-squares approximations are real, while the square collocation gives some eigenvalues with nonzero imaginary part. The convergence
Figure 5.13: Eigenvalues of the disk computed using multiquadrics and $c = 3$. 
rates of the least-squares approach, however, are worse than the ones obtained with square collocation, and convergence is much less smooth.

5.4.2 Eigenvalues as local maxima of the pseudoresolvent

In the formulation of least-squares differentiation matrices, the goal is to minimize $\|A\lambda - u\|$, where $A\lambda$ is the approximation of $u$ in a finite dimensional subspace of smooth functions. Alternatively, one may seek least-squares solutions of eigenvalue problems is that minimize the discretized version of $\|\Delta u - \nu u\|$, for $u = 0$ on the boundary. The difficulty in this approach is that $\nu$ is not known a priori, which leads to non-linear minimizations. Fortunately, the non-linearity is only on $\nu$. Moreover, we can use tools of pseudospectra for rectangular pencils [76] in the minimization process.

5.4.2.1 Formulation

Using the notation introduced in section 3.4, the goal is to find local solutions of

$$\min_{\nu, \lambda} \|A_L^T \lambda - \nu A_I \lambda\|$$

subject to the constraints

$$\|A_I \lambda\| = 1 \text{ and } A_B \lambda = 0.$$  \hspace{1cm} (5.7)

The constraints are needed to normalize the computed eigenfunctions and to enforce the boundary condition (Dirichlet in this case). Although we do not pursue the possibility, the boundary condition need not be enforced strongly – a penalty method can be used to minimize errors on the boundary.

Following the procedure presented in Section 3.4, we can eliminate the boundary constraint through a direct elimination. In order to so, consider the QR decomposition of $A_B$:

$$A_B \Pi_B = Q_B R_B,$$
Figure 5.14: Eigenvalues of the disk computed using multiquadrics and $c = 1$. 
where $\Pi_B$ is a permutation matrix. We consider the case that $A_B$ may be rank-deficient and write $R_B$ as

$$R_B = \begin{bmatrix} R_{B1} & R_{B2} \\ 0 & 0 \end{bmatrix},$$

where $R_{B1}$ is nonsingular. The boundary constraint can now be written as

$$R_{B1} \hat{\lambda}_1 + R_{B2} \hat{\lambda}_2 = 0,$$

where $\Pi_B^T \lambda = \begin{bmatrix} \hat{\lambda}_1 \\ \hat{\lambda}_2 \end{bmatrix}$. This gives $\hat{\lambda}_1 = -R_{B1}^{-1}R_{B2} \hat{\lambda}_2$ and

$$\begin{bmatrix} \hat{\lambda}_1 \\ \hat{\lambda}_2 \end{bmatrix} = \begin{bmatrix} -R_{B1}^{-1}R_{B2} \\ I \end{bmatrix} \hat{\lambda}_2.$$

Here $I$ is an identity matrix – its size is determined by the size of the vector $\hat{\lambda}_2$.

Define the matrix

$$B = \Pi_B \begin{bmatrix} -R_{B1}^{-1}R_{B2} \\ I \end{bmatrix}$$

and notice that $\lambda = B \hat{\lambda}_2$. We can, therefore, rewrite the minimization problem with only one constraint:

$$\min_{\nu, y} \|(A_I^T B - \nu A_I B) \hat{\lambda}_2\|, \quad \|A_I B \hat{\lambda}_2\| = 1.$$

Let $QR = A_I B$ be a reduced QR factorization of $A_I B$ and define $y = R \hat{\lambda}_2$. The minimization statement now becomes:

$$\min_{\nu, y} \|((A_I^T B R^{-1} - \nu Q)y\|, \quad \|y\| = 1. \quad (5.8)$$

We shall call any matrix of the form $A - zB$ with $z \in \mathbb{C}$ a pencil [32].

Notice that for a square pencil $A - zB$, the resolvent norm $\|(A - zB)^{-1}\|$ is infinite at eigenvalues. For a rectangular pencil, the pseudoresolvent norm $\|(A - zB)^+\|$ may have no infinities, approximations to the continuous eigenvalue problem in this
Figure 5.15: Surface of the pseudoresolvent norm for a rectangular pencil obtained from a discretization of the Laplacian in the disk. Dashed lines mark the location of true eigenvalues.

Case are the local maxima instead. This is illustrated in Figure 5.15, where the pseudoresolvent norm for a rectangular pencil obtained from a discretization of the Laplacian eigenvalue problem of the disk is presented. Also notice that the height of the peak can be related to the accuracy of the approximation.

Fortunately, there are robust algorithms to treat (5.8). They stem from the computation of pseudospectra — see e.g. [67, 76] and references therein. Following [76], we have that (5.8) can be replaced with

$$
\min_{z} \sigma_{\text{min}} (z Q - A_L^{L} B R^{-1}),
$$

(5.9)

which is a one-parameter minimization problem. Here $\sigma_{\text{min}}(P)$ denotes the smallest singular value of $P$. The local solutions of (5.9) are the desired approximations of
the eigenvalues. The plot in Figure 5.15 was obtained with a MATLAB software package for the computation of pseudospectra called EigTool [75].

We point out that (5.9) is equivalent to

$$\min_z \sigma_{\min} \left( \begin{bmatrix} zB_1 - A_1 \\ A_2 \end{bmatrix} \right),$$

where $B_1$ and $A_1$ are $n \times n$ matrices and $A_2$ has at most $n$ rows, where $n$ is the number of columns of $Q$ — see Appendix A for details. This reduces computational costs as evaluations of the pencil for different values of $z$ are needed in the minimization process. In Appendix B we also show that if accuracy close to machine precision is desired, then this minimization problem can be replaced by a rootfinding problem.

### 5.4.2.2 Numerical experiments

Figure 5.16 shows a plot of $\sigma_{\min}(zQ - A_L^T BR^{-1})$ for several real values of $z$. Here we seek eigenvalues of the Laplacian in the disk. Computations were carried out with multiquadrics and shape parameter $c = 3$. To better illustrate the behavior of $\sigma_{\min}$, only 45 basis functions were used together with 94 clustered least-squares nodes (as in Figure 5.13). Notice that the local minima of $\sigma_{\min}$ is very well defined for the eigenmodes that are well resolved by the basis provided. As a consequence, only a few steps (sometimes only 2) are needed to converge to the solution. We point out however, that special care should be taken with initial guesses as iterations may converge to a local maximum if the derivative of $\sigma_{\min}$ is used in the solution search.

Figure 5.17 was generated with the same data as in Figure 5.13. The error decays at about the same rate as the square collocation on clustered nodes and the least-squares approximations obtained with differentiation matrices. The errors seem to be slightly smaller for the present method; the computational cost, on the other hand, was significantly higher than the needed to compute the eigenvalues with least-squares differentiation matrices since in this case a large number of eigenvalues
Figure 5.16: Plot of $\sigma_{\text{min}}(zQ-A_l^TBR^{-1})$ for several real values of $z$ for the eigenvalues of the disk. Dashed lines represent the values of true eigenvalues.

were computed. If only one or two eigenvalues are computed, this difference is significantly reduced. We emphasize that no attempt has been made to optimize the procedures that compute $\sigma_{\text{min}}$, this is the most time consuming step in our implementation.

5.4.2.3 Polynomial and singular basis

The least-squares methods used in this chapter are not restricted to RBFs. Other basis can be used according to the problem at hand. Consider, for instance, tensor product of polynomials in 2-D. In order to avoid ill-conditioning of the basis, we consider Chebyshev polynomials. We embed the domain in consideration $\Omega$ in $[-1,1] \times [-1,1]$ were these polynomials are well-defined. If the $\Omega$ covers a relatively large area of the square, the polynomial basis is well-conditioned as long as there are enough nodes to capture their oscillations.
We again consider the eigenvalues of the disk. We computed the eigenvalues using the least-squares method introduced in section 5.4.2.1 together with a polynomial basis. In Figure 5.18 presents the relative error for the first 11 eigenvalues (data for repeated eigenvalues are not shown). Eigenvalues are computed almost to machine precision.

For regions with corners we consider yet another basis. Recall from section 5.3 that an eigenfunction at a corner can be expanded as

\[ u(r, \theta) = \sum_{n=1}^{\infty} \sum_{k=0}^{\infty} b_{nk} r^{n\varphi+2k} \sin(n\varphi \theta). \] (5.11)

In section section 5.3 we appended functions from this expansion to an RBF sum in order to capture the singular behavior. Here we consider approximations with basis functions of the form \( \{r^{n\varphi+2k} \sin(n\varphi \theta)\} \). If there are no other singularities (besides
the point singularity at the corner) we found that this basis can be used to approximate the eigenmodes of the Laplacian. When used with square collocation methods, however, the resulting coefficient matrices are ill-conditioned or even singular. The Least-squares formulation allows stable computations.

In Figure 5.19 we show the relative error in the first 3 eigenvalues of the L-shaped region. The least-squares method described above together with basis functions of the form \( \{ r^{n\varphi + 2k} \sin(n\varphi \theta) \} \) was used in the approximations. Least-squares nodes were "roughly" equispaced and the aspect ratio was kept at approximately 4. The error that decays the fastest in this figure corresponds to the third eigenmode, which has an smooth eigenfunction. The convergence rates may be improved with boundary clustering of nodes.
**Figure 5.19:** Relative error in the first 3 eigenvalues of the L-shaped region. The basis functions are of the form \( r^{n+2k} \sin(n \varphi \theta) \).
6.1 Summary and discussion

In this dissertation we were primarily concerned with the accuracy and sta-
bility of global methods based on smooth RBFs. Theoretical and practical aspects
were considered. We investigated the role of interpolation nodes on the accuracy
and stability of GRBFs approximations. We showed that specific node locations can
be obtained to stabilize GRBF interpolation and discretizations for time-dependent
problems. We also explored more practical matters such as least-squares methods,
Arnoldi-like iterations for GRBFs, and algorithms for eigenvalue problems.

GRBFs using equally spaced centers are easily related to polynomials in a
transformed variable. This connection allows us to apply polynomial interpolation
and potential theory to draw a number of precise conclusions about the convergence
of GRBF interpolation. In particular, for a given interpolation node density one
can derive spectral convergence (or divergence) rates based on the singularity lo-
cations of the target function. Conversely, one can easily compute node densities
for which analyticity of the function in $[-1,1]$ is sufficient for convergence and for
which the Lebesgue constant is controlled. Furthermore, the polynomial connec-
tion allows us to exploit barycentric Lagrange interpolation to construct a simple,
explicit interpolation algorithm that avoids the ill-conditioning of the interpolation
matrix. Although these conclusions can only be proved for a very special class of
RBFs, they have strong implications on the statement that RBF collocation meth-
ods are *meshfree* numerical schemes. Gaussian RBFs with equally spaced centers
and nodes in 1-D, for instance, do not allow stable interpolation; they require specific node placement, as polynomials do, for good convergence. Global polynomial methods are not regarded as meshfree for this very reason; why should Gaussian RBFs be? Moreover, the stability results presented in chapters 2 and 4 also apply for tensor-product regions in \( \mathbb{R}^d \).

For problems in complicated geometries, finding universally stable nodes for RBF collocation seems daunting. In chapter 3 we proposed using least-squares approximations as the foundation of a differentiation matrix. This offers the possibility of separating the requirements of accuracy (governed mostly by the RBF centers) and stability (mandating clustering near boundaries). The added flexibility can be used to adapt centers to data, or to use an Arnoldi-like iteration for equispaced centers to circumvent RBF conditioning issues. Differentiation matrices based on the least-squares idea can incorporate boundary conditions strongly and remain as convenient as collocation methods for problems with variable coefficients and non-linearity.

Eigenvalue stability is a crucial factor in the usefulness of RBF discretizations for time-dependent problems. In chapter 4 we proved that under mild conditions, RBF methods are eigenvalue stable in the absence of boundaries, including methods on periodic domains. However, in the presence of boundaries, RBF collocation is quite likely to be unstable. We also showed that Gaussian RBF collocation is stable when special node distributions are used in one dimension. As far as we know, this is the first conclusive demonstration that node locations can eliminate instability asymptotically. While in principle this result should extend to tensor product regions, there is probably little practical interest in implementing RBF methods in such cases, since polynomials are also available. More promisingly for applications, differentiation matrices based on least squares can remain eigenvalue stable for widely different discretization parameters.
In chapter 5, we presented several numerical experiments for eigenvalue problems. They show that boundary clustering, even if imperfect, and the collocation of the PDE on the boundary prevent degradation of the solution near the boundary for moderate values of $N$. This conclusion is in agreement with the work of Fornberg and Flyer [26] and Fedoseyev et al. [23], who used these techniques for boundary-value problems. Corner singularities, however, can frustrate convergence even with clustering, unless singular terms are added to the RBF approximation. Least-squares methods were also used in the computation of eigenmodes. For the Laplacian operator, eigenvalues of least-squares matrices were found to have almost purely real spectrum, which is not the case for square matrices. The rate of convergence of least-squares methods, however, showed no improvement over the results obtained with square collocation and boundary clustering. A connection between least-squares approximations for eigenvalues and the pseudospectra of rectangular matrices has been explored in section 5.4.2. Algorithms for pseudospectra are robust and can be used eigenvalue approximation of a elliptic operator. The nonlinearity introduced by the method, however, makes it less attractive than other approaches if more than a few eigenvalues are required.

Despite all the difficulties presented in this work, RBFs have already secured their place in the fields of approximation theory and numerical analysis. Smooth RBFs are more general and flexible than polynomials. The question that still remains is how to take full advantage of the additional degrees of freedom they provide. Regarding the computational costs (most methods presented in this thesis require $O(N^3)$ operations), recent advances on domain decomposition [3] and fast evaluations of RBFs [13] offer viable approaches to overcome this difficulty.
6.2 Future work

RBFs are fascinating and promising, but there is a lot of work to be done to make RBF-based methods more competitive. They seem to be particularly useful for problems on the unit sphere and circle, were stability comes naturally. A systematic study to determine the advantages of RBF methods over traditional methods such as Fourier spectral methods is necessary. Would the flexibility on node locations for boundary-free problems allow efficient adaptive algorithms? Algorithms that avoid the use of ill-conditioned matrices are also fundamental to the application of RBFs to more demanding computational problems, such as those that stem from fluid mechanics. The work of Fornberg and Wright in [29] and our algorithm based on barycentric interpolation suggests that this might be achieved for more general settings.

Other research topics that are not particularly restricted to RBFs, that can be explored in the future, are the investigation of high-order/spectral methods for problems with point singularities and the use of least-squares matrices for more complex problems. The Arnoldi-like algorithm for bivariate polynomials depicted in Section 3.2 shall be investigated in the near future and compared other techniques such as domain embedding.
Appendix A

REDUCING THE SIZE OF A RECTANGULAR PENCIL

In this appendix we present a result obtained by T. Wright and L.N. Trefethen in [76] that reduces the size of a rectangular pencil \((A - zB)\) of aspect ratio larger than 2.

Assume that \(A\) and \(B\) are \(m \times n\) matrices, with \(m > n\). Let \(B = QR\) be a QR factorization of \(B\), then

\[
\|A - zB\| = \|Q(Q^T A - zR)\| = \|Q^T A - z \begin{bmatrix} \hat{R} \\ 0 \end{bmatrix}\| = \|Q^T A - z \begin{bmatrix} \hat{R} \\ 0 \end{bmatrix}\| = \left\| \begin{bmatrix} \tilde{A}_1 - z\hat{R} \\ \tilde{A}_2 \end{bmatrix} \right\|
\]

where \(\tilde{A}_1\) and \(\hat{R}\) are \(n \times n\) matrices. Notice that

\[
\left\| \begin{bmatrix} \tilde{A}_1 - z\hat{R} \\ \tilde{A}_2 \end{bmatrix} \right\|^2 = \|\tilde{A}_1 - z\hat{R}\|^2 + \|\tilde{A}_2\|^2.
\]  

(A.1)

For the case that \(m > 2n\) we can reduce the size of the pencil. In order to do so, let \(\tilde{A}_2 = Q_2R_2\) be a QR factorization. It then follows that

\[
\|\tilde{A}_2\| = \|R_2\| = \|\hat{R}_2\|
\]  

(A.2)

where \(\hat{R}_2\) is an \(n \times n\) upper triangular matrix obtained with a reduced QR factorization of \(\tilde{A}_2\). Collecting (A.1) and (A.2) gives

\[
\|A - zB\| = \left\| \begin{bmatrix} \tilde{A}_1 - z\hat{R} \\ \hat{R}_2 \end{bmatrix} \right\|.
\]

Notice that the pencil in the right hand side has at most 2\(n\) rows.
Appendix B

THE MINIMIZATION OF THE MINIMUM SINGULAR VALUE TO MACHINE PRECISION

It is well known that if rounding errors are $O(\varepsilon)$, the minimization of a function of one variable is in general $O(\sqrt{\varepsilon})$ accurate in the independent variable [71]. In other words, finding $x$ that minimizes $f(x)$ is usually computed at most to 8 digits in double precision, unless the derivative of $f$ is known, in which case one may seek the roots of $\dot{f}$ to minimize $f$. For the minimization in (5.9) or (5.10) we have the following theorem that avoids this difficulty.

**Theorem B.1** If $z$ minimizes

$$\sigma_{\min} \left( \left[ zB - A \right] \right), \quad (B.1)$$

then $u^T B v = 0$, where $u$ and $v$ are the left and right singular vectors corresponding to $\sigma_{\min}$.

**Proof.** Let $P = \left[ zB - A \right]$ and $\sigma$ be a singular value of $P$ with corresponding singular vectors $u$ and $v$,

$$\begin{bmatrix} 0 & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \sigma \begin{bmatrix} u \\ v \end{bmatrix}. $$

Define

$$M = \begin{bmatrix} 0 & P \\ P^T & 0 \end{bmatrix} \quad \text{and} \quad x = \begin{bmatrix} u \\ v \end{bmatrix}. $$

116
It now follows that
\[ \frac{dM}{dz} x + M \frac{dx}{dz} = \frac{d\sigma}{dz} x + \sigma \frac{dx}{dz}. \]

Multiplying through by \( x^T \) gives,
\[ x^T \frac{dM}{dz} x + x^T (M - \sigma I) \frac{dx}{dz} = 2 \frac{d\sigma}{dz}. \]

Since \( x^T (M - \sigma I) x = 0 \), we have that,
\[ \frac{d\sigma}{dz} = \frac{1}{2} x^T \frac{dM}{dz} x. \]

Notice that
\[ x^T \frac{dM}{dz} x = \begin{bmatrix} u^T & v^T \end{bmatrix} \begin{bmatrix} 0 & B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = v^T B^T u + u^T B v = 2 u^T B v. \]

Since \( \frac{d\sigma}{dz} = 0 \) at a local minimum, we have that \( u^T B v = 0 \). \( \square \)

This implies that instead of using a minimization procedure, we can seek the roots of \( u^T B v = 0 \) whenever accuracy of more than 8 digits is desired.
BIBLIOGRAPHY


