UPPSALA
UNIVERSITET

IT Licentiate theses
2020-002

Global radial basis function collocation methods for PDEs

ULRIKA SUNDIN

UPPSALA UNIVERSITY
Department of Information Technology

UPPSALA
UNIVERSITET

Global radial basis function collocation methods for PDEs

Ulrika Sundin
Ulrika.Sundin@it.uu.se

December 2019

Division of Scientific Computing
Department of Information Technology
Uppsala University
Box 337
SE-751 05 Uppsala
Sweden

http://www.it.uu.se/

Dissertation for the degree of Licentiate of Technology in Scientific Computing

ISSN 1404-5117
Printed by the Department of Information Technology, Uppsala University, Sweden

To my father

Abstract

Radial basis function (RBF) methods are meshfree, i.e., they can operate on unstructured node sets. Because the only geometric information required is the pairwise distance between the node points, these methods are highly flexible with respect to the geometry of the computational domain. The RBF approximant is a linear combination of translates of a radial function, and for PDEs the coefficients are found by applying the PDE operator to the approximant and collocating with the right hand side data. Infinitely smooth RBFs typically result in exponential convergence for smooth data, and they also have a shape parameter that determines how flat or peaked they are, and that can be used for accuracy optimization. In this thesis the focus is on global RBF collocation methods for PDEs, i.e., methods where the approximant is constructed over the whole domain at once, rather than built from several local approximations. A drawback of these methods is that they produce dense matrices that also tend to be ill-conditioned for the shape parameter range that might otherwise be optimal. One current trend is therefore to use over-determined systems and least squares approximations as this improves stability and accuracy. Another trend is to use localized RBF methods as these result in sparse matrices while maintaining a high accuracy. Global RBF collocation methods together with RBF interpolation methods, however, form the foundation for these other versions of RBF–PDE methods. Hence, understanding the behaviour and practical aspects of global collocation is still important. In this thesis an overview of global RBF collocation methods is presented, focusing on different versions of global collocation as well as on method properties such as error and convergence behaviour, approximation behaviour in the small shape parameter range, and practical aspects including how to distribute the nodes and choose the shape parameter value. Our own research illustrates these different aspects of global RBF collocation when applied to the Helmholtz equation and the Black–Scholes equation.

Acknowledgments

First of all I would like to thank my supervisor Elisabeth Larsson. It has been almost 20 years since we first got to know each other, when you were a PhD student and taught the very first scientific computing course I ever took. I quickly fell in love with the field and felt like I finally knew why I was in the engineering program and what I wanted to specialize in. I did not know much about PhD studies then, but I was intrigued and asked you to tell me more. A few years later you were the supervisor of my MSc project and our work continued seamlessly into my PhD studies. Then... life happened, as they say, and I had to prioritize and fight for my health and contemplate what I felt was truly important to me. Having a family was at the top of my list, and so my PhD studies had to wait while I was busy having and raising children. You never once questioned my choices and priorities, nor did you ever express any doubt in my ability to come back and finish my postgraduate studies. For this I am forever grateful. You have become so much more than a supervisor to me—I consider you my mentor, role model, sister and friend.

To my husband, Jonas Sundin: Thank you for loving me, especially at those times when I am perhaps less easy to love. We have been through a lot together and life has not always been easy, but you have supported me through it all. Thank you for making silly jokes that crack me up daily and for using your computer skills to find my email address that time when you first saw me in the campus computer lab. Clearly one of your best ideas ever, because here we are together two decades and four children later. Thank you for being such a loving father to our boys. We love you. To my boys Noah, Levi, Isak and Eli: Thank you for bringing so much love and laughter into my life—and just the right amount of mischief too. You remind me daily that there is so much more to life than work and you help me keep my priorities straight. I love you and I am so proud and blessed to be your mother.

To my father, Gunnar Pettersson: Thank you for introducing me to science and technology in so many different ways when I was a little girl. You took me out at night and showed me the moon and the stars and told me their names. I got to spend time with you out in the fields in a tractor as well as in your workshop watching you repair things, and sometimes I even got to hang out with you in your darkroom watching photographs magically appear on white sheets of paper. You only got to attend school for a few years but you knew so much. You helped me with my physics homework in high school and I admired your ability to construct and build things. I think you never realized how gifted you were. You may not have been quick to praise but I know you were proud of me and my academic results.

Unfortunately you did not get to see me complete my postgraduate studies, but I am happy you got to meet two of my children instead. I miss you. To my mother, Ingrid Pettersson: Thank you for always being there for me and for selflessly giving so much of your time and energy to my sisters and me when we were kids. Despite doing most of the housework and working outside the home you were never too busy for us. Thank you for being the best grandmother to my boys and their cousins and for spoiling us with "fika" every time we see you. To my sister, Kristina Zettergren: thank you for always making me feel safe and calm and for never outgrowing your sense of wonder and ability to find joy in the simple things, like buzzing bumblebees and cute animals. To my sister, Maria Wallander: Thank you for setting a high standard academically, pushing me to excel. You are the best mother to my nephews and niece. Thank you for finding time to listen to my problems despite having a hectic life yourself. To my uncle and aunt, Valter and Härdis Petersson and to my cousin Lena Adem: Thank you for always making me feel welcome in your home when visiting you growing up. I am especially thankful for your prayers. To my entire family, thank you for giving me a wonderful childhood. I love you all.

To my high school math and chemistry teacher Holger Johansson: thank you for your sense of humour and for making your classes fun and exhilarating. To Sandy Siegel: thank you for all your support, for allowing me the honour of writing the family article for your book, and for pushing me to continue pursuing my PhD. To my mentor and friend, Lilian Westin: thank you for encouraging me to believe in myself and my professional skills.

To Kerstin Forsberg, my best friend from high school: It took us almost three school years in the same class to find each other but we have been friends ever since. We have that magical friendship that stays close even if we do not see each other for years. You have come so far in your academic career, Docent Forsberg, and I am so proud to call you my friend. I am sure I will get to call you Professor Forsberg too one day.

Finally, to my friends Jenny Andersson, Madelene Westbom and Jenny Owetz: thank you for being there and listening to me in good times and in bad. To Sofia Bodare: thank you for fighting for what is important to you whatever life throws at you, and for inspiring me to do the same. And to my newfound friend Lael Rodriguez Olsson: Thank you for your prayers and for having such a generous heart despite having been through some of the hardest things a parent could ever be faced with. Your continuing faith is beautiful and inspires me to keep going when life is hard and my own faith may falter.

List of Papers

This thesis is based on the following papers

I U. Pettersson, E. Larsson, G. Marcusson and J. Persson. *Improved radial basis function methods for multi-dimensional option pricing*, J. Comput. Appl. Math. 222 (2008), pp. 82–93.

Contributions: The author of this thesis implemented the method, performed the numerical experiments and contributed to the analysis of the results.

II E. Larsson and U. Sundin. *An investigation of global radial basis function collocation methods applied to Helmholtz problems*, pp. 1–28. Submitted.

Contributions: The author of this thesis contributed to the experiments, the analysis of the flat RBF limit, and to the writing of the article.

Reprints were made with permission from the publisher.

Contents

Chapter 1

Introduction

In this introductory section we describe the fundamental ideas of radial basis function (RBF) approximation methods and we also give a brief historical overview, listing some of the important steps in the development of these methods. We also present the focus of this thesis, list the PDE applications that we have used in our own research, and provide a motivational example that illustrates the competitiveness of RBF methods. In Section 2 we describe radial basis functions and their properties in more detail and in Section 3 we list different versions of RBF methods for time-independent PDEs. In Section 4 we then describe how to solve time-dependent problems and we also briefly discuss stability issues. In Section 5 we present results regarding approximation in the limit where the RBFs become flat and in Section 6 we provide a more detailed description of RBF approximation errors. We conclude with some practical aspects of RBF approximation in Section 7 and a summary in Swedish in Section 8. Throughout this thesis a general overview of each topic is presented along with illustrating examples from our own research.

1.1 Fundamental ideas

A radial function, ϕ, defined on $\mathbb{R}^d$, is a function whose value at each point depends only on the distance between that point and the centre point of the function, i.e.,

$$\phi(\boldsymbol{x}) = \varphi(r), \qquad \boldsymbol{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d, \qquad r = \sqrt{x_1^2 + \cdots + x_d^2}. \tag{1.1}$$

In radial basis function (RBF) approximation, we first choose a set of node points or use a node set where data is available. At each node we centre a translate of a specific radial function, which is typically the same for all nodes. We then use these translates as basis functions (RBFs) and let a linear combination of them form our approximant, i.e., we have

$$s(\boldsymbol{x},\varepsilon)=\sum_{j=1}^{N}\lambda_j\phi(\|\boldsymbol{x}-\boldsymbol{x}_j\|,\varepsilon), \tag{1.2}$$

with the coefficients $\lambda_1,\ldots,\lambda_N$ for the N basis functions centred at the nodes $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N$.

For interpolation of the given data $\boldsymbol{f}=(f_1,\ldots,f_N)^T$ at the node points $(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N)^T$, we match the approximant with the corresponding data value at each node point, i.e.

$$s(\boldsymbol{x}_i,\varepsilon)=f_i, \qquad i=1,\ldots,N, \tag{1.3}$$

or in matrix form

$$A\boldsymbol{\lambda}=\boldsymbol{f}, \tag{1.4}$$

where

$$\begin{aligned} A_{ij} &= \phi(\|\boldsymbol{x}_i-\boldsymbol{x}_j\|,\varepsilon),\\ f_i &= f(\boldsymbol{x}_i). \end{aligned} \tag{1.5}$$

Once we have solved the system (1.4) for the coefficients λ_j, we can evaluate the approximant (1.2) at any point in the computational domain.

The corresponding technique for partial differential equations (PDEs) is called collocation. In this case we apply the PDE interior and boundary operators to our approximant and then match the resulting expressions with the given right hand side data at the corresponding nodes. Global collocation simply means that we approximate the solution function over the whole computational domain at once, in contrast to building the global solution from a combination of local approximations.

With RBF approximation methods, the only geometric information needed is the pairwise distance between points. This makes RBF methods easy to implement for problems of any dimension. There is no need to set up a computational mesh and the node distributions are not confined to specific structures like for example tensor product grids. The RBF methods are thus very flexible with respect to the geometry of the computational domain. Another advantage is that the infinitely smooth RBFs result in methods featuring exponential convergence for smooth solutions.

A drawback of global RBF methods is that the globally supported RBFs, i.e., the RBFs whose values are non-zero on the whole computational domain, result in dense matrices which makes the corresponding systems of equations computationally expensive to solve. This includes the infinitely smooth RBFs which typically also produce ill-conditioned matrices, especially for the parameter values that lead to the highest accuracy.

1.2 A brief historical overview

The RBF method using the so called multiquadric (MQ) RBF was introduced for interpolation of scattered topography data in the late 1960s [43, 44]. The MQ method however remained largely unknown until the publication of [38] in 1982, where the RBF method using MQ was found to perform the best in a comparison of various scattered data interpolation methods [50]. This result may have inspired other researchers to study the MQ method and in 1990 excellent results for this method were presented, not only for scattered data interpolation but also for the estimation of partial derivatives, as well as for the solution of PDEs [50, 51]. In the early years of research on RBF methods for PDEs, the focus was on global collocation methods. In more recent years, overdetermined RBF systems and least squares solution approximations have been favoured, as this improves stability and accuracy [93]. The current trend is also to use localized RBF methods as these result in sparse matrices while maintaining a high accuracy. One such method is the RBF-generated finite difference (RBF–FD) method, first described in [98] in 2000. The idea of the RBF–FD method is to form computational stencils, similar to those in finite difference methods, by constructing derivative discretizations based on RBF interpolants on localized node sets [26]. Another localized method is the radial basis function partition of unity method (RBF–PUM). The PUM method was presented in the context of finite element methods in [73, 1] in the 1990s. In the latter of these two articles, the authors list different shape functions used for data fitting, including RBFs, and mention that these can also be used in variational formulations for the solution of PDEs, thus implicitly suggesting the possibility of combining RBFs with PUM. An algorithm, similar to PUM, for scattered data fitting can also be found already in [37] from 1977. The theory of RBF interpolation was combined with a partition of unity method in [100] in 2002. Since then, RBF–PUM versions have also been developed for PDE applications [45, 87, 59, 10]. The basic concept of the RBF–PUM approach is to divide the computational domain into mildly overlapping subdomains that together cover the domain, and to then construct a local RBF approximation on each subdomain. The global solution is then given by the sum of these local approximations multiplied by partition of unity weight functions, i.e., a set of compactly supported functions whose sum is one at each point in the global domain. For more extensive recent works on RBF methods and related topics, see for example [5, 101, 94, 18, 88, 26].

1.3 Thesis focus, PDE applications and a motivational example

The focus of this thesis is on the theory, behaviour and practical aspects of global RBF collocation methods. As stated above, the trend is shifting from these methods towards localized and least squares versions. However, the results for global RBF collocation are still relevant, as interpolation and global collocation together constitute the foundation on which the localized methods are built.

For reference in subsequent chapters, where various results of our own research are presented as illustrations of different aspects of RBF approximation, we here list the application problems we have used. We also present a motivational example illustrating the efficiency of the RBF method.

1.3.1 The Helmholtz equation

The Helmholtz equation, which is a time-independent linear PDE, is in all examples given by

$$\mathcal{L}^1 u(\boldsymbol{x}) = -\Delta u(\boldsymbol{x}) - \kappa^2 u(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in \Omega^1 = \Omega. \tag{1.6}$$

We have chosen this problem because it is in general harder to solve than the Laplace and Poisson equations, especially for large wavenumbers. Some reasons for this are the indefiniteness of the operator, the wave nature of the solution, and the typically more complicated boundary conditions. There is also a problem parameter, κ, which can be varied in order to study its connection with the RBF method parameters.

The simplest model problem is one-dimensional, with $\Omega = (0, 1)$, and non-reflecting (or radiation) boundary conditions given by

$$\begin{array}{rclclr} \mathcal{L}^2 u(x) & = & -\dfrac{du}{dx}(x) - i\kappa u(x) & = & -2i\kappa, & x = 0, \\ \mathcal{L}^3 u(x) & = & \dfrac{du}{dx}(x) - i\kappa u(x) & = & 0, & x = 1. \end{array} \tag{1.7}$$

The analytical solution for this problem is $u(x) = \exp(i\kappa x)$, if κ is constant.

The second problem is two-dimensional with a rectangular domain $\Omega = (0, L_1) \times (0, 1)$. At the top and bottom boundaries, we use the Dirichlet boundary condition

$$\mathcal{L}^4 u(\boldsymbol{x}) = u(\boldsymbol{x}) = 0, \quad \boldsymbol{x} = (0, x_2) \text{ or } \boldsymbol{x} = (L_1, x_2), \tag{1.8}$$

indicating that we consider a waveguide type of problem. The conditions at the left and right boundaries are

$$\begin{aligned} \mathcal{L}^2 u(\boldsymbol{x}) &= -\frac{\partial u}{\partial x_2}(\boldsymbol{x}) - i\beta_m u(\boldsymbol{x}) &= -2i\beta_m \sin(\alpha_m x_1), \quad &\boldsymbol{x} = (x_1, 0), \\ \mathcal{L}^3 u(\boldsymbol{x}) &= \frac{\partial u}{\partial x_2}(\boldsymbol{x}) - i\beta_m u(\boldsymbol{x}) &= 0, \quad &\boldsymbol{x} = (x_1, 1), \end{aligned} \tag{1.9}$$

where $\alpha_m = \frac{m\pi}{L_1}$, $\beta_m = \sqrt{\kappa^2 - \alpha_m^2}$, and $m \geq 1$ is an integer. These conditions allow for just one propagating mode in the solution, which is given by $u(\boldsymbol{x}) = \exp(i\beta_m x_2)\sin(\alpha_m x_1)$, assuming that κ is constant.

The third and final problem is also two-dimensional, but the domain Ω is now enclosed between two curves $\gamma_1(x_2) < x_1 < \gamma_2(x_2)$, $x_2 \in (0,1)$, see Figure 1.1. The Dirichlet condition (1.8) is modified to hold at γ_1 and γ_2.

$$\mathcal{L}^4 u(\boldsymbol{x}) = u(\boldsymbol{x}) = 0, \quad \boldsymbol{x} = (\gamma_j(x_2), x_2), \quad j = 1, 2. \tag{1.10}$$

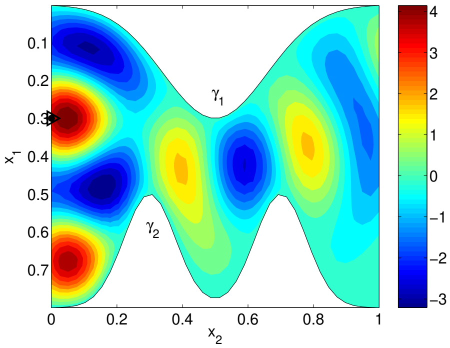


Figure 1.1: Wave propagation in an M-shaped duct. The source position is indicated by the marker at the left boundary and the wave number is $\kappa = 6\pi$. The real part of the solution is displayed.

At the left and right boundary, we here use so called Dirichlet–to–Neumann map (DtN) radiation boundary conditions [52]

$$\begin{aligned} \mathcal{L}^2 u(\boldsymbol{x}) &= -\frac{\partial u}{\partial x_2} - i\sum_{m=1}^{\infty} \beta_m \langle u(\cdot, 0), \psi_m^0 \rangle \psi_m^0(x_1) \\ &= -2i\sum_{m=1}^{\infty} A_m \beta_m \psi_m^0(x_1), \qquad x_2 = 0, \\ \mathcal{L}^3 u(\boldsymbol{x}) &= \frac{\partial u}{\partial x_2} - i\sum_{m=1}^{\infty} \beta_m \langle u(\cdot, 1), \psi_m^1 \rangle \psi_m^1(x_1) = 0, \quad x_2 = 1, \end{aligned} \tag{1.11}$$

where, for a fixed x_2, the modes $\psi_m^{x_2} = \sqrt{2}\sin(\alpha_m(x_1 - \gamma_1(x_2)))$, with $\alpha_m = \frac{m\pi}{\gamma_2(x_2) - \gamma_1(x_2)}$. The inner product is given by

$$\langle u(\cdot, x_2), \psi_m^{x_2} \rangle = \int_{\gamma_1(x_2)}^{\gamma_2(x_2)} u(x_1, x_2) \psi_m^{x_2}(x_1)\, dx_1, \tag{1.12}$$

and the amplitudes are chosen to emulate a point source, i.e., $A_m = \psi_m^0(x_s)$, where x_s is the position of the source in the vertical coordinate. The (DtN) conditions allow for any combination of modes to move transparently through the vertical boundaries. For practical and computational reasons, however, the infinite sum is truncated at $\mu(x_2) = \lfloor \frac{\kappa(\gamma_2(x_2) - \gamma_1(x_2))}{\pi} \rfloor$.

1.3.2 The Black–Scholes equation

The Black–Scholes equation is a time-dependent linear PDE that is posed as a final value problem in its original formulation. Here we use a transformed version of the PDE, where time is reversed to make standard texts on time-integration for PDEs applicable, and all variables have been scaled to be dimensionless. The details of the transformation can be found in [78]. The transformed problem reads

$$\left\{ \begin{array}{lcl} \dfrac{\partial}{\partial \hat{t}} P(\hat{t}, \boldsymbol{x}) & = & \mathcal{L}P(\hat{t}, \boldsymbol{x}), \quad \hat{t} \in \mathbb{R}_+, \quad \boldsymbol{x} \in \mathbb{R}_+^d, \\ P(0, \boldsymbol{x}) & = & \Phi(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}_+^d, \end{array} \right. \tag{1.13}$$

where

$$\mathcal{L}P = 2\bar{r} \sum_{i=1}^{d} x_i \frac{\partial P}{\partial x_i} + \sum_{i,j=1}^{d} \left[\bar{\sigma} \bar{\sigma}^T \right]_{ij} x_i x_j \frac{\partial^2 P}{\partial x_i \partial x_j} - 2\bar{r}P, \tag{1.14}$$

and $P(\hat{t}, \boldsymbol{x})$ is the value of the option at the transformed time $\hat{t}$ when the underlying scaled assets have the values given by $\boldsymbol{x}$. Furthermore, the coefficient $\bar{r}$ is the scaled short interest rate, $\bar{\sigma}$ is the scaled volatility and d denotes the number of underlying assets and thus the number of spatial dimensions of the problem. We use the following contract function for a European basket call option

$$\Phi(\boldsymbol{x}) = \max \left(\frac{1}{d} \sum_{i=1}^{d} x_i - \bar{K},\, 0 \right), \tag{1.15}$$

where the scaled strike price in our case is $\bar{K} = 1$.

In [49], the authors show that the problem we consider here is actually well-posed without boundary conditions as long as the growth at infinity is restricted. Therefore, we only use near- and far-field boundary conditions. This means that no boundary conditions are employed at boundaries of the type $\Gamma_i = \{\boldsymbol{x} \mid \boldsymbol{x} \in \mathbb{R}_+^d, \quad \boldsymbol{x} \neq \mathbf{0}, \quad x_i = 0\}$, $i = 1, \ldots, d$. The near-field boundary can be seen as the single point $\boldsymbol{x} = \mathbf{0}$, and there we enforce

$$P(\hat{t}, \mathbf{0}) = 0. \tag{1.16}$$

The problem is defined on $\mathbb{R}_+^d$, but for computational reasons we need to restrict the problem to a finite domain. Given the structure of the contract function (1.15) we choose a far-field boundary surface of the type $\sum_{i=1}^{d} x_i = C$, where the constant C is chosen to bring the surface far enough from the origin for the far-field solution (1.17) to be an accurate approximation. We then use this asymptotic solution as our far-field boundary condition

$$P(\hat{t}, \boldsymbol{x}) \to \frac{1}{d}\sum_{i=1}^{d} x_i - \bar{K}e^{-2\bar{r}\hat{t}}, \quad \|\boldsymbol{x}\| \to \infty. \tag{1.17}$$

1.3.3 A motivational example

As previously mentioned, one of the drawbacks of global RBF methods is that they produce dense matrices, increasing the storage space needed and making the corresponding systems of equations computationally expensive to solve. However, in Paper I, we solved the Black–Scholes equation (1.13) in 1D and 2D and compared the time and memory requirements of the RBF method for different solution errors with those of an adaptive finite difference (FD) method, which produces a sparse system matrix. The results can be seen in Figures 1.2 and 1.3. The RBF method is faster than the FD method in both 1D and 2D and the memory requirements are similar for the methods in 2D. This shows that global RBF methods can be competitive even in geometries where standard methods producing sparse matrices are available.

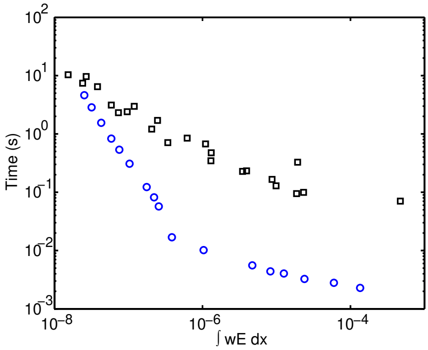


Figure 1.2: Time efficiency comparison between the RBF method (circles) and the second order accurate adaptive finite difference method (squares) for the one-dimensional Black–Scholes problem. $\int wE\,dx$ is the error in a weighted norm, where the main weight is concentrated around the region of financial interest.

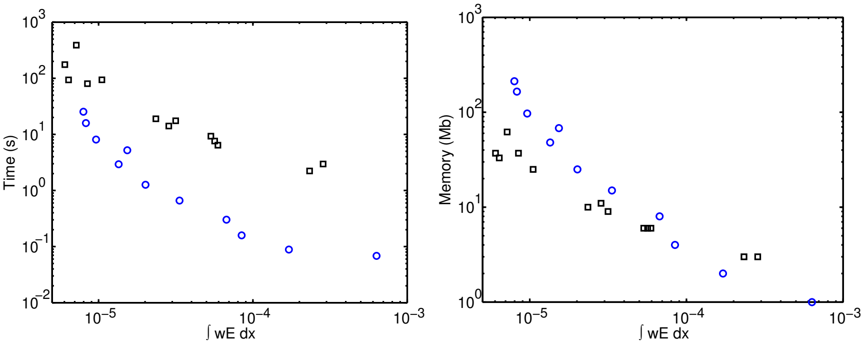


Figure 1.3: Time and memory efficiency comparison between the RBF method (circles) and the second order accurate adaptive finite difference method (squares) for the two-dimensional Black–Scholes problem. $\int wE\,d\boldsymbol{x}$ is the error in a weighted norm, where the main weight is concentrated around the region of financial interest.

Chapter 2

RBF types and well-posedness of the interpolation problem

There are different ways to categorize RBFs. In this section we briefly discuss the different types of RBFs and their characteristics as well as conditions for the well-posedness of the interpolation problem (1.2)–(1.5). A more detailed description, especially concerning the convergence and error behaviour associated with the different RBF types, is given in Section 6 and some general guidelines on the choice of RBF and related issues are discussed in Section 7. Examples of the most commonly used RBFs are listed in Tables 2.1 and 2.2.

2.1 RBF types

Our focus is mainly on the infinitely smooth RBFs, i.e., RBFs that are infinitely many times differentiable. These RBFs also have a shape parameter, ε, that determines how flat or peaked the RBFs are, see Figure 2.1. Flatter basis functions result in higher accuracy for smooth functions, but also worse conditioning and numerical instability. The shape parameter naturally introduces a possibility for optimization but it is typically difficult to know what shape parameter value to choose a priori. The infinitely smooth RBFs typically lead to exponential convergence as the node density increases and as the shape parameter decreases. This has been observed numerically for the commonly used RBFs and proved theoretically for the GA and (inverse) MQ RBFs [18, 70, 101, 67]. Unfortunately though, the condition numbers of the interpolation matrices of the infinitely smooth RBFs grow exponentially both with decreasing minimum distance between node points and with

Table 2.1: Some examples of strictly positive definite RBFs. Note that the Bessel-type RBFs ϕ_d are strictly positive definite in up to d dimensions for $d = 2, 3, \ldots$ [29]. K_ν is the modified Bessel function of the second kind and of order ν, Γ is the gamma function and $p_{d,k}(\varepsilon r) \in \mathcal{P}_{deg=k}$ are certain polynomials of degree k.

Strictly positive definite RBFs	
Infinitely smooth RBFs	$\boldsymbol{\phi(r, \varepsilon)}$
Gaussian (GA)	$e^{-(\varepsilon r)^2}, \quad \varepsilon > 0$
Inverse multiquadric (IMQ)	$\frac{1}{\sqrt{1+(\varepsilon r)^2}}$
Generalized IMQ	$\frac{1}{(1+(\varepsilon r)^2)^\beta}, \quad \beta > 0$
Inverse quadratic (IQ)	$\frac{1}{1+(\varepsilon r)^2}$
Bessel-type RBFs (ϕ_d) on $\mathbb{R}^k, \quad k \leq d$	$\frac{J_{d/2-1}(\varepsilon r)}{(\varepsilon r)^{d/2-1}}, \quad d = 2, 3, \ldots$
Piecewise smooth RBFs	$\boldsymbol{\phi(r, \varepsilon)}$
Matérn functions on $\mathbb{R}^d$	$\frac{K_{\beta-d/2}(\varepsilon r)(\varepsilon r)^{\beta-d/2}}{2^{\beta-1}\Gamma(\beta)}, \quad \beta > \frac{d}{2}$
Wendland CSRBF $\varphi_{d,k}$ on $\mathbb{R}^d$	$(1-\varepsilon r)_+^{\lfloor d/2 \rfloor + 2k+1} p_{d,k}(\varepsilon r)$

Table 2.2: Some examples of strictly conditionally positive definite RBFs.

Strictly conditionally positive definite RBFs		
Infinitely smooth RBFs	$\boldsymbol{\phi(r, \varepsilon)}$	Order
Multiquadric (MQ)	$\sqrt{1+(\varepsilon r)^2}$	1
Generalized MQ (GMQ)	$(1+(\varepsilon r)^2)^{\beta/2}$, $0 < \beta \notin 2\mathbb{N}$	$\lceil \frac{\beta}{2} \rceil$
Piecewise smooth RBFs	$\boldsymbol{\phi(r)}$	Order
Radial powers (RP$_n$)	$r^\beta, \quad \beta = 1, 3, 5 \ldots$	$\lceil \frac{\beta}{2} \rceil$
Thin plate splines (TPS$_n$)	$r^\beta \log r, \quad \beta = 2, 4, 6 \ldots$	$\frac{\beta}{2} + 1$

decreasing shape parameter value [18, 101].

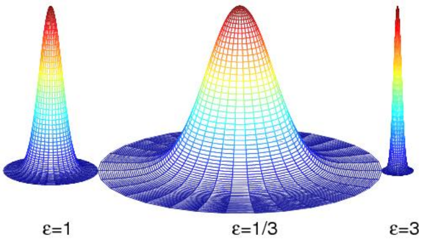


Figure 2.1: The shape of the Gaussian (GA) RBF for three different shape parameter values. (Image courtesy of Elisabeth Larsson.)

Most of the piecewise smooth RBFs do not have any shape parameter and thus they cannot be optimized in that sense. That makes them simpler to use as there is no need to choose a shape parameter value. They also lead to less ill-conditioned systems than the infinitely smooth RBFs with condition numbers growing algebraically with the node density [18, 101]. This type of RBF however can only give algebraic convergence [18, 105, 101].

Both the infinitely smooth and the piecewise smooth RBFs have global support, i.e., they are non-zero over the whole computational domain, which leads to dense matrices in the resulting systems of equations. But there are also compactly supported RBFs. These RBFs are non-zero only in a finite region around their centre and the support size is determined by the shape parameter value. The condition number of the interpolation matrix corresponding to this RBF type grows algebraically with decreasing minimum node distance and the compactly supported RBFs also result in sparse matrices, but similarly to the piecewise smooth RBFs they can only give algebraic convergence [18, 101]. This convergence also comes at the cost of keeping the support-size fixed when increasing the node density, which corresponds to increasing the bandwidth of the interpolation matrix. If the bandwidth is kept constant there will essentially be no convergence and so another approach, such as multilevel methods, is recommended for this type of RBF [18].

2.2 RBF positive definiteness and well-posedness of the interpolation problem

In the remainder of this section we focus on some conditions related to the well-posedness of the interpolation problem.

Definition 2.2.1 ([18], Theorem 3.2.). *A real-valued continuous function ϕ is* ***positive definite*** *on $\mathbb{R}^d$ if it is even, i.e., $\phi(\boldsymbol{x}) = \phi(-\boldsymbol{x})$, and if*

$$\sum_{i=1}^{N}\sum_{j=1}^{N} c_i c_j \phi(\boldsymbol{x}_i - \boldsymbol{x}_j) \geq 0, \tag{2.1}$$

for any N pairwise different points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^d$, and $\boldsymbol{c} = (c_1, \ldots, c_N)^T \in \mathbb{R}^N$. The function ϕ is ***strictly positive definite*** *on $\mathbb{R}^d$ if the quadratic form (2.1) is zero only for $\boldsymbol{c} \equiv \boldsymbol{0}$.*

Some examples of strictly positive definite RBFs are listed in Table 2.1. The strict positive definiteness of an RBF guarantees that the interpolation matrix A in (1.5) is positive definite too, and thus invertible.

Some RBFs fulfil similar but less strict conditions given in Definition 2.2.2.

Definition 2.2.2 ([18], Theorem 7.1.). *A real-valued continuous even function ϕ is* ***conditionally positive definite of order m*** *on $\mathbb{R}^d$ if*

$$\sum_{i=1}^{N}\sum_{j=1}^{N} c_i c_j \phi(\boldsymbol{x}_i - \boldsymbol{x}_j) \geq 0, \tag{2.2}$$

for any N pairwise different points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N \in \mathbb{R}^d$, and $\mathbf{c} = (c_1, \dots, c_N)^T \in \mathbb{R}^N$ satisfying

$$\sum_{i=1}^{N} c_i p(\boldsymbol{x}_i) = 0, \tag{2.3}$$

for any real-valued polynomial p of degree at most $m-1$. The function ϕ is ***strictly conditionally positive definite of order m*** *on $\mathbb{R}^d$ if the quadratic form (2.2) is zero only for $\mathbf{c} \equiv \mathbf{0}$.*

Some examples of strictly conditionally positive definite RBFs are presented in Table 2.2. Most of these RBFs do not automatically guarantee the invertibility of the interpolation matrix in its simplest form given by (1.5). Instead we need to append some extra terms to the interpolant as described below, see [18]. Using a radial function, $\phi(r)$, that is conditionally positive definite of order m, let the interpolant be

$$s(\boldsymbol{x}, \varepsilon) = \sum_{j=1}^{N} \lambda_j \phi(\|\boldsymbol{x} - \boldsymbol{x}_j\|, \varepsilon) + \sum_{k=1}^{M} \alpha_k p_k(\boldsymbol{x}), \qquad \boldsymbol{x} \in \mathbb{R}^d, \tag{2.4}$$

where $p_1, \dots, p_M$ form a basis for the $M = \binom{m-1+d}{m-1}$-dimensional linear space $\mathcal{P}^d_{m-1}$ of polynomials of total degree less than or equal to $m-1$ in d variables. Now we have $N+M$ unknown coefficients so we need to add M conditions, which are chosen as (2.5) below. Note the similarity between these conditions and the conditions in Definition 2.2.2, and that this particular choice together with some additional requirements specified in Theorem 1 guarantees well-posedness of the interpolation problem using (2.4).

$$\sum_{j=1}^{N} \lambda_j p_k(\boldsymbol{x}_j) = 0, \qquad k = 1, \dots, M. \tag{2.5}$$

We now have the following system of equations for the interpolation problem

$$\begin{pmatrix} A & P \\ P^T & O \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f} \\ \mathbf{0} \end{pmatrix}, \tag{2.6}$$

where $A_{ij} = \phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|, \varepsilon)$, $i, j = 1, \ldots, N$, $P_{ik} = p_k(\boldsymbol{x}_i)$, $i = 1, \ldots, N$, $k = 1, \ldots, M$, $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_N)^T$, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_M)^T$, $\boldsymbol{f} = (f_1, \ldots, f_N)^T$, $\mathbf{0}$ is a zero vector of length M and O is an $M \times M$ zero matrix.

For the purpose of presenting a theorem on the well-posedness of the interpolation problem (2.6) we introduce the concept of polynomial unisolvency.

Definition 2.2.3. *A finite set of points $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subset \mathbb{R}^d$ is said to be **m-unisolvent** if the only polynomial of total degree at most m interpolating zero data on $\mathcal{X}$ is the zero polynomial. This is equivalent to saying that any polynomial $p \in \mathcal{P}_m^d$ is uniquely determined by its values on $\mathcal{X}$.*

The following theorem now gives the conditions for which the interpolation problem (2.6) is well-posed:

Theorem 1 ([18], Theorem 7.2.)**.** *If the real-valued even function ϕ is strictly conditionally positive definite of order m on $\mathbb{R}^d$ and the points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ form an (m-1)-unisolvent set, then the linear system of equations (2.6) has a unique solution.*

We are really free to append any M linearly independent functions in (2.4) and not just polynomials. With the particular choice of adding polynomials of degree at most $m-1$ we get reproduction of polynomials of degree up to $m-1$ provided that the set $\mathcal{X}$ is $(m-1)$-unisolvent. This means that if the interpolation data comes from a polynomial of total degree at most $m-1$, then it is exactly interpolated by (2.4).

Some strictly conditionally positive definite RBFs of order one are special in the sense that they guarantee well-posedness of the simplest interpolation system (1.4), i.e., they do not need the extra terms in (2.4). These include the radial powers $\phi(r) = r^\beta$ for $0 < \beta < 2$, as well as the generalized MQ RBFs $\phi(r, \varepsilon) = (1 + (\varepsilon r)^2)^\beta$ with $0 < \beta < 1$ [18], including MQ as a special case, as conjectured in [39] and proved in [74].

Chapter 3

Global collocation methods for PDEs

In this section we describe three different versions of global RBF collocation. These methods have been compared in the literature and the accuracy has been found to be very similar for the non-symmetric and the Hermite-based methods [56, 18]. The third method, where the PDE is imposed also on the boundary, was found to be less accurate when keeping the total number of node points the same as for the other methods [56]. Some other advantages and drawbacks of each approach are briefly discussed at the end of each method description.

All methods are applied to the following general time-independent PDE problem on a given domain $\Omega \subset \mathbb{R}^d$

$$\begin{aligned} \mathcal{L}u(\boldsymbol{x}) &= f(\boldsymbol{x}), \qquad && \boldsymbol{x} \in \Omega, \\ \mathcal{L}_{\mathcal{B}}u(\boldsymbol{x}) &= g(\boldsymbol{x}), \qquad && \boldsymbol{x} \in \partial\Omega. \end{aligned} \tag{3.1}$$

3.1 Non-symmetric collocation

The non-symmetric collocation approach was first presented by Kansa [50, 51] using the MQ RBF and it is therefore also referred to as Kansa's method or sometimes the multiquadric method. Kansa also originally used different shape parameter values at the different node points but for simplicity and clarity of presentation we use a constant shape parameter here.

The non-symmetric method is the most straight-forward type of collocation. We begin by letting our RBF approximant be a linear combination of translates of the radial basis function, ϕ, centred at the N points $\boldsymbol{\xi}_j \in \Xi$.

That is,

$$s(\boldsymbol{x}, \varepsilon) = \sum_{j=1}^{N} \lambda_j \phi(\|\boldsymbol{x} - \boldsymbol{\xi}_j\|, \varepsilon). \tag{3.2}$$

The collocation or test points, $\boldsymbol{x}_i \in \mathcal{X}$, are often chosen to coincide with the centre points. Separating the collocation and centre points can sometimes be useful though, e.g., in optimal node placement techniques [64, 65, 66, 63] and to improve accuracy near the boundaries of the computational domain [25]. More details on this can be found in Section 7. Inserting (3.2) into equation (3.1) and collocating with the PDE at the interior points $\boldsymbol{x}_i \in \mathcal{I}$ and with the boundary conditions at the boundary points $\boldsymbol{x}_i \in \mathcal{B}$ then results in

$$\begin{array}{rcllr} \mathcal{L}s(\boldsymbol{x},\varepsilon)|_{\boldsymbol{x}=\boldsymbol{x_i}} & \equiv & \sum_{j=1}^{N} \lambda_j \mathcal{L}\phi(\|\boldsymbol{x} - \boldsymbol{\xi}_j\|, \varepsilon)|_{\boldsymbol{x}=\boldsymbol{x_i}} & = f(\boldsymbol{x}_i), & i \in \mathcal{I}, \\ \mathcal{L}_{\mathcal{B}}s(\boldsymbol{x},\varepsilon)|_{\boldsymbol{x}=\boldsymbol{x_i}} & \equiv & \sum_{j=1}^{N} \lambda_j \mathcal{L}_{\mathcal{B}}\phi(\|\boldsymbol{x} - \boldsymbol{\xi}_j\|, \varepsilon)|_{\boldsymbol{x}=\boldsymbol{x_i}} & = g(\boldsymbol{x}_i), & i \in \mathcal{B}, \end{array} \tag{3.3}$$

which can be written in block matrix notation as

$$\begin{pmatrix} L \\ B \end{pmatrix} (\boldsymbol{\lambda}) = \begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{g} \end{pmatrix}, \tag{3.4}$$

where

$$\begin{array}{rcllll} L_{ij} & = & \mathcal{L}\phi(\|\boldsymbol{x} - \boldsymbol{\xi}_j\|, \varepsilon)|_{\boldsymbol{x}=\boldsymbol{x_i}}, & \boldsymbol{x}_i \in \mathcal{I}, & \boldsymbol{\xi}_j \in \Xi, \\ B_{ij} & = & \mathcal{L}_{\mathcal{B}}\phi(\|\boldsymbol{x} - \boldsymbol{\xi}_j\|, \varepsilon)|_{\boldsymbol{x}=\boldsymbol{x_i}}, & \boldsymbol{x}_i \in \mathcal{B}, & \boldsymbol{\xi}_j \in \Xi, \\ f_i & = & f(\boldsymbol{x}_i), & \boldsymbol{x}_i \in \mathcal{I}, & \\ g_i & = & g(\boldsymbol{x}_i), & \boldsymbol{x}_i \in \mathcal{B}. & \end{array} \tag{3.5}$$

There is unfortunately no guarantee that the collocation matrix is non-singular for this type of collocation [46]. However, many researchers still prefer using this method because it is so simple to implement and because the node distributions for which the collocation matrix is singular seem to be rare [18, 46]. It is also the collocation type that we have used the most in our own research.

For PDEs with a parameter that can be varied, such as the wavenumber for the Helmholtz equation, it becomes particularly clear that the system matrix for non-symmetric RBF collocation may be singular. In Paper II we present singularity results obtained by analysis of an eigenvalue problem related to the 1D Helmholtz equation (1.6)–(1.7). We show that for any given node distribution (with distinct nodes) there are wavenumbers κ that lead to a singular collocation matrix. In Figure 3.1, the eigenvalues that lead to a singular system are computed for different problem sizes using MQ and GA RBFs. The interesting region is where the wavenumbers are real and lead to well resolved solutions. For MQ and GA there are no eigenvalues in this region.

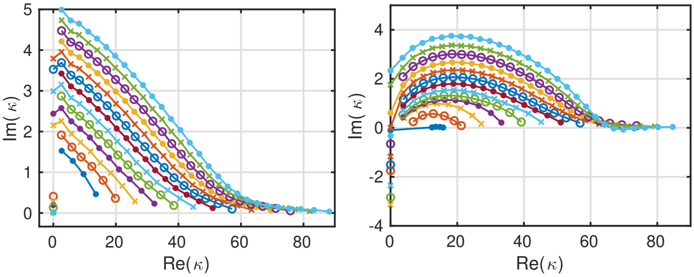


Figure 3.1: The wave numbers that lead to a singular system for the one-dimensional Helmholtz problem using $N = 6, 8, \ldots, 30$ from bottom to top, for MQ RBFs with $\varepsilon = 5$ (left) and GA RBFs with $\varepsilon = 10$ (right).

3.2 Hermite-based collocation

The Hermite-based collocation approach, which is also referred to as symmetric collocation, was first suggested in [17] and builds on the interpolation technique described in [104]. We now let the node and centre point sets coincide, so that $\mathcal{X} = \Xi$. This is a necessary condition for the theoretical foundation of the method, i.e., in order to achieve a symmetric collocation matrix for real valued operators and a Hermitian collocation matrix for complex valued operators.

Here we use an approximant of the following form

$$s(\boldsymbol{x},\varepsilon) = \sum_{j=1}^{N_{\mathcal{I}}} \lambda_j \overline{\mathcal{L}^{\boldsymbol{\xi}}} \phi(\|\boldsymbol{x}-\boldsymbol{\xi}\|,\varepsilon)|_{\boldsymbol{\xi}=\boldsymbol{\xi}_j} + \sum_{j=N_{\mathcal{I}}+1}^{N} \lambda_j \overline{\mathcal{L}_{\mathcal{B}}^{\boldsymbol{\xi}}} \phi(\|\boldsymbol{x}-\boldsymbol{\xi}\|,\varepsilon)|_{\boldsymbol{\xi}=\boldsymbol{\xi}_j}, \tag{3.6}$$

where $\overline{\mathcal{L}^{\boldsymbol{\xi}}}$ and $\overline{\mathcal{L}_{\mathcal{B}}^{\boldsymbol{\xi}}}$ are the complex conjugates of the PDE and boundary operators applied to $\boldsymbol{\xi}$, $N_{\mathcal{I}}$ is the number of interior centre points and N is the total number of centre points.

Inserting (3.6) into (3.1) and collocating with the PDE at the interior points and with the boundary conditions at the boundary points now yields the block matrix equation

$$\begin{pmatrix} A_{\mathcal{L}\overline{\mathcal{L}^{\boldsymbol{\xi}}}} & A_{\mathcal{L}\overline{\mathcal{L}^{\boldsymbol{\xi}}_{\mathcal{B}}}} \\ A_{\mathcal{L}_{\mathcal{B}}\overline{\mathcal{L}^{\boldsymbol{\xi}}}} & A_{\mathcal{L}_{\mathcal{B}}\overline{\mathcal{L}^{\boldsymbol{\xi}}_{\mathcal{B}}}} \end{pmatrix} (\boldsymbol{\lambda}) = \begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{g} \end{pmatrix}, \tag{3.7}$$

where

$$\begin{aligned}
(A_{\mathcal{L}\overline{\mathcal{L}^{\boldsymbol{\xi}}}})_{ij} &= \mathcal{L}\overline{\mathcal{L}^{\boldsymbol{\xi}}}\phi(\|\boldsymbol{x}-\boldsymbol{\xi}\|)|_{\boldsymbol{x}=\boldsymbol{x}_i,\boldsymbol{\xi}=\boldsymbol{\xi}_j}, && \boldsymbol{x}_i, \boldsymbol{\xi}_j \in \mathcal{I} \\
(A_{\mathcal{L}\overline{\mathcal{L}^{\boldsymbol{\xi}}_{\mathcal{B}}}})_{ij} &= \mathcal{L}\overline{\mathcal{L}^{\boldsymbol{\xi}}_{\mathcal{B}}}\phi(\|\boldsymbol{x}-\boldsymbol{\xi}\|, \varepsilon)|_{\boldsymbol{x}=\boldsymbol{x}_i,\boldsymbol{\xi}=\boldsymbol{\xi}_j}, && \boldsymbol{x}_i \in \mathcal{I}, \quad \boldsymbol{\xi}_j \in \mathcal{B}, \\
(A_{\mathcal{L}_{\mathcal{B}}\overline{\mathcal{L}^{\boldsymbol{\xi}}}})_{ij} &= \mathcal{L}_{\mathcal{B}}\overline{\mathcal{L}^{\boldsymbol{\xi}}}\phi(\|\boldsymbol{x}-\boldsymbol{\xi}\|, \varepsilon)|_{\boldsymbol{x}=\boldsymbol{x}_i,\boldsymbol{\xi}=\boldsymbol{\xi}_j}, && \boldsymbol{x}_i \in \mathcal{B}, \quad \boldsymbol{\xi}_j \in \mathcal{I}, \\
(A_{\mathcal{L}_{\mathcal{B}}\overline{\mathcal{L}^{\boldsymbol{\xi}}_{\mathcal{B}}}})_{ij} &= \mathcal{L}_{\mathcal{B}}\overline{\mathcal{L}^{\boldsymbol{\xi}}_{\mathcal{B}}}\phi(\|\boldsymbol{x}-\boldsymbol{\xi}\|, \varepsilon)|_{\boldsymbol{x}=\boldsymbol{x}_i,\boldsymbol{\xi}=\boldsymbol{\xi}_j}, && \boldsymbol{x}_i, \boldsymbol{\xi}_j \in \mathcal{B}, \\
f_i &= f(\boldsymbol{x}_i), && \boldsymbol{x}_i \in \mathcal{I}, \\
g_i &= g(\boldsymbol{x}_i), && \boldsymbol{x}_i \in \mathcal{B}.
\end{aligned} \tag{3.8}$$

The collocation matrix resulting from this approach is non-singular, in contrast to the non-symmetric collocation matrix, provided that the RBFs are strictly positive definite or strictly conditionally positive definite. In the latter case, suitable polynomial terms must also be appended to the approximant [18, 104]. The collocation matrix is also, depending on the PDE operator involved, symmetric or Hermitian which can enable more efficient implementations. However the matrix is more complicated to assemble and it requires smoother basis functions than the non-symmetric collocation method as it involves higher derivatives.

In Paper II we show that the collocation matrix of the Hermitian-based approach is indeed Hermitian for the one-dimensional Helmholtz problem (1.6)–(1.7). The approximant then takes the form

$$s(x) = \sum_{k=1}^{N_{\text{op}}} \sum_{j=1}^{N_k} \lambda_j^k \overline{\mathcal{L}^k_\xi} \phi(x, \xi_j^k),$$

where $\phi(x, \xi_j^k) = \phi(\|x - \xi_j^k\|)$, N_{op} is the total number of interior and boundary equations in the PDE and N_k is the number of centre points belonging to each corresponding operator $\overline{\mathcal{L}^k_\xi}$. Letting collocation and centre points coincide, i.e., letting $x_j = \xi_j$, leads to a system of equations with the following structure

$$\begin{pmatrix} \mathcal{L}^1_x\overline{\mathcal{L}^1_\xi}\phi & \mathcal{L}^1_x\overline{\mathcal{L}^2_\xi}\phi & \mathcal{L}^1_x\overline{\mathcal{L}^3_\xi}\phi \\ \mathcal{L}^2_x\overline{\mathcal{L}^1_\xi}\phi & \mathcal{L}^2_x\overline{\mathcal{L}^2_\xi}\phi & \mathcal{L}^2_x\overline{\mathcal{L}^3_\xi}\phi \\ \mathcal{L}^3_x\overline{\mathcal{L}^1_\xi}\phi & \mathcal{L}^3_x\overline{\mathcal{L}^2_\xi}\phi & \mathcal{L}^3_x\overline{\mathcal{L}^3_\xi}\phi \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda}^0 \\ \boldsymbol{\lambda}^1 \\ \boldsymbol{\lambda}^2 \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ -2i\kappa \\ 0 \end{pmatrix},$$

where the block $\mathcal{L}_x^j\overline{\mathcal{L}_\xi^k}\phi$ is of size $(N_j \times N_k)$, where N_j is the number of collocation points corresponding to operator $\mathcal{L}_x^j$. To see that the coefficient matrix really is Hermitian, we can use the following differentiation rules for the RBFs

$$\begin{aligned}
\frac{\partial^n}{\partial \xi^n}\phi(x_j, x_k) &= (-1)^n \frac{\partial^n}{\partial x^n}\phi(x_j, x_k), \qquad (3.9)\\
\frac{\partial^n}{\partial x^n}\phi(x_k, x_j) &= (-1)^n \frac{\partial^n}{\partial x^n}\phi(x_j, x_k). \qquad (3.10)
\end{aligned}$$

We can then show for the different blocks in the matrix that the matrix elements satisfy $m_{jk} = \overline{m}_{kj}$. As an example, for elements in the first two off-diagonal blocks we get

$$\begin{aligned}
\mathcal{L}_x^0\overline{\mathcal{L}_\xi^1}\phi(x_j, x_k) &= (-\frac{\partial^2}{\partial x^2} - \kappa^2)(-\frac{\partial}{\partial \xi} + i\bar{\kappa})\phi(x_j, x_k) = (-\frac{\partial^2}{\partial x^2} - \kappa^2)(\frac{\partial}{\partial x} + i\bar{\kappa})\phi(x_j, x_k),\\
\overline{\mathcal{L}_x^1\overline{\mathcal{L}_\xi^0}\phi(x_k, x_j)} &= (-\frac{\partial}{\partial x} + i\bar{\kappa})(-\frac{\partial^2}{\partial \xi^2} - \kappa^2)\phi(x_k, x_j) = (\frac{\partial}{\partial x} + i\bar{\kappa})(-\frac{\partial^2}{\partial x^2} - \kappa^2)\phi(x_j, x_k).
\end{aligned}$$

3.3 Collocation with the PDE on the boundary

The idea of this collocation approach, first presented in [21], is to add more information on the boundary of the computational domain. The motivation for this is that the accuracy in RBF approximation, as for most interpolation methods, tends to be the lowest near the boundaries [25, 56].

We use the following approximant

$$s(\boldsymbol{x}, \varepsilon) = \sum_{j=1}^{N+N_B} \lambda_j \phi(\left\|\boldsymbol{x} - \boldsymbol{\xi}_j\right\|, \varepsilon). \qquad (3.11)$$

Since we want to add extra information on the boundary, i.e., collocate with more conditions there, we also need to add the corresponding number of centres in order to have as many unknowns as equations in our system. We place these extra centres just outside the computational domain, as this keeps the average distance between the node points about the same thus preserving the conditioning. We let the other centres coincide with the collocation points as follows

$$\boldsymbol{\xi}_j = \begin{cases} \boldsymbol{x}_j & \boldsymbol{x}_j \in \mathcal{I} \cup \mathcal{B}, \\ \text{a point outside } \Omega & \boldsymbol{\xi}_j \in \Xi \setminus (\mathcal{I} \cup \mathcal{B}). \end{cases} \qquad (3.12)$$

Collocation with the PDE at the interior node points and with both boundary conditions and PDE at the boundary points results in

$$\begin{pmatrix} L \\ B \end{pmatrix} (\boldsymbol{\lambda}) = \begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{g} \end{pmatrix}, \tag{3.13}$$

where

$$\begin{aligned} L_{ij} &= \mathcal{L}\phi(\|\boldsymbol{x} - \boldsymbol{\xi}_j\|, \varepsilon)|_{\boldsymbol{x}=\boldsymbol{x_i}}, & \boldsymbol{x}_i &\in \mathcal{I} \cup \mathcal{B}, \quad \boldsymbol{\xi}_j \in \Xi, \\ B_{ij} &= \mathcal{L}_{\mathcal{B}}\phi(\|\boldsymbol{x}_i - \boldsymbol{\xi}_j\|, \varepsilon)|_{\boldsymbol{x}=\boldsymbol{x_i}}, & \boldsymbol{x}_i &\in \mathcal{B}, \quad \boldsymbol{\xi}_j \in \Xi, \\ f_i &= f(\boldsymbol{x}_i), & \boldsymbol{x}_i &\in \mathcal{I} \cup \mathcal{B}, \\ g_i &= g(\boldsymbol{x}_i), & \boldsymbol{x}_i &\in \mathcal{B}. \end{aligned} \tag{3.14}$$

It is important to remember that adding extra information and thus adding RBF centres actually increases the size of the system to be solved. We can think of this as needing to add centre points in one dimension less than the dimension of the computational domain and this is not a negligible contribution to the total number of node points. Therefore, in order to get a fair comparison between this approach and the other collocation methods, we need to make sure that the total number of node points is the same for all methods.

Chapter 4

Time-dependent problems

There are mainly two different approaches to solving time-dependent PDEs using global RBF collocation. The first method is the so called method of lines and the second approach is to use space-time RBFs.

4.1 The method of lines

In the method of lines technique the PDE is semi-discretized, i.e., the spatial operator is discretized resulting in a system of ordinary differential equations in time. This system can then be solved by discretization in time using a standard time-integration method typically based on finite differences. The solution is thus computed sequentially in time with intervals given by the time-step, Δt. When solving time-dependent linear PDEs, the discretized problem is usually formulated in such a way that the solution values are computed directly rather than via the explicit computation of the RBF approximant coefficients. Solving the PDE for the nodal solution values directly is sometimes referred to as the pseudospectral RBF method (RBF-PS).

If we have the following general time-dependent PDE (with suitable initial and boundary conditions not explicitly written out here)

$$\frac{\partial u}{\partial t}(\boldsymbol{x}, t) = \mathcal{L}u(\boldsymbol{x}, t) + f(\boldsymbol{x}, t), \tag{4.1}$$

where $\mathcal{L}$ is the spatial differentiation operator, we get the following general method of lines formulation with global RBF collocation in space

$$\frac{d\boldsymbol{u}}{dt}(t) = A_{\mathcal{L}}A^{-1}\boldsymbol{u}(t) + \boldsymbol{f}(t). \tag{4.2}$$

Here, $A_{\mathcal{L}}A^{-1}$ is called the spatial differentiation matrix,

$$\begin{array}{rcl}(A_{\mathcal{L}})_{ij} & = & \mathcal{L}\Phi(\|\boldsymbol{x}-\boldsymbol{x}_j\|,\varepsilon)|_{\boldsymbol{x}=\boldsymbol{x}_i}, \\ A_{ij} & = & \Phi(\|\boldsymbol{x}_i-\boldsymbol{x}_j\|,\varepsilon),\end{array} \tag{4.3}$$

and $\boldsymbol{u}$ is a vector containing the solution values at the different spatial nodes. The chosen time-integration method is then applied to (4.2). Depending on the number of time levels used in the finite difference approximation of the time derivative, a number of starting values might need to be computed through another time-stepping method before switching to the main method.

4.1.1 Stability

As a rule of thumb, the method of lines is stable if the eigenvalues of the spatial differentiation matrix scaled by the time-step lie within the so called stability region of the time-discretization operator. Another rule of thumb is that regardless of the stability properties of the time-stepping method, the real part of the eigenvalues of the spatial differentiation matrix should be non-positive in order to avoid growing solution components over time. For non-normal operators one also needs to study the pseudospectrum, as small perturbations may shift a part of the numerical spectrum out of the stability region [99, 87]. The details of the pseudospectrum analysis are however beyond the scope of this thesis.

For parabolic PDEs implicit time-stepping methods are often used as the stability restrictions on the time-step would be severe for explicit methods. For hyperbolic PDEs explicit methods are instead often used because this class of equations allows larger time-steps. The cost per time step is bigger for implicit time-stepping methods than for explicit methods, because implicit methods require the solution of a linear system of equations at each time step, but implicit methods can still outperform explicit ones in terms of total cost. This is because implicit methods allow significantly larger time-steps than the explicit methods. In [96] this is demonstrated using an RBF–PUM method for the Black–Scholes equation for American option pricing, where the fully implicit method is much more time efficient than the implicit-explicit method which in turn is more efficient than the fully explicit method, all because of the time-step restrictions for the explicit and implicit-explicit methods.

As previously mentioned it is a problem if the spatial differentiation matrix has eigenvalues with a positive real part as this causes growth in the corresponding solution components for time-dependent problems. For non-dissipative operators, this is unfortunately often true for differentiation matrices obtained with RBF collocation using infinitely smooth basis functions. In [82] the authors prove that RBF methods with conditionally positive definite RBFs are time-stable for differential operators with constant coefficients for any node distribution in periodic domains if the matrix $A_{\mathcal{L}}$ is anti-symmetric. For positive definite RBFs the same holds for linear operators without the constant coefficients condition. The anti-symmetry leads to a purely imaginary spectrum, but if the centre points are slightly shifted from the node points, the anti-symmetry property is lost which might again cause instabilities. The authors also show that for the Gaussian RBF, certain node distributions can lead to a time-stable RBF method in 1D. However, stable node distributions are not known for general domains in higher dimensions and they do not allow adaptive resolution. The authors therefore instead suggest constructing differentiation matrices by using a least-squares approximation with boundary conditions enforced strongly.

In [72] the first known Lax-stability analysis of the RBF collocation method for convection problems on the circle and sphere is presented suggesting that stability can be achieved on equispaced collocation points. Small shifts in the node placement however cause instabilities and the ideal node set might also be impractical or unavailable. Moreover, RBF methods are the most attractive when collocation points are not equispaced. The authors therefore suggest a least-squares method in order to minimize the effect of unstable eigenvalues of the differentiation matrix.

For purely convective PDEs another stabilization technique often used for other methods is to add a so called hyperviscosity term which is a dissipative term consisting of a power of the Laplacian. The idea is to dampen high frequency modes while leaving the low frequency modes intact. For global collocation methods a more efficient approach with a similar effect is to add $-\gamma A^{-1}u$ to the spatial operator of the PDE, where $\gamma > 0$ and A is the RBF interpolation matrix (1.5) [30]. The eigenvalues of A are of the order of increasing powers of the shape parameter for infinitely smooth RBFs and the corresponding eigenvectors are increasingly oscillatory. The inverse of A has the same eigenvectors as A, but the eigenvalues are the inverses of those of A. For positive definite RBFs the inverse eigenvalues are positive and will grow increasingly fast for higher frequency modes. Using the inverse of A as a filter will therefore dampen low eigenmodes slowly and high eigenmodes fast. The parameter γ can be used for fine-tuning of the filter. A small value will lead to gentle damping of the high frequency modes while a wide range of low frequency modes are left mostly intact.

A semi-Lagrangian method for the simulation of transport on a sphere using global RBF collocation is presented in [95]. This method does not require the use of stabilizing terms such as hyperviscosity and thus reduces the number of parameters that need to be tuned. The idea is to solve a simple ODE involving the velocity field of the PDE in order to find the departure point of a so called solution parcel. The idea is that a parcel reaching a specific node point at a certain time left the departure point at the previous discrete time. The current solution is then interpolated from the node points to the departure points and the new solution value at each node point is then set to the interpolated value at the corresponding departure point. The finding of the departure points is a so called upwinding technique ensuring that the numerical domain of dependence matches the physical domain of dependence.

4.1.2 The method of lines for the Black–Scholes equation

We now demonstrate the RBF approximation and time-stepping procedure for the Black–Scholes problem (1.13) from Paper I. We approximate the solution with a time-dependent linear combination of RBFs centred at the node points $\boldsymbol{x}_k$, $k = 1, \dots, N$,

$$u(\hat{t}, \boldsymbol{x}) = \sum_{k=1}^{N} \lambda_k(\hat{t}) \phi(\varepsilon \|\boldsymbol{x} - \boldsymbol{x}_k\|) = \sum_{k=1}^{N} \lambda_k(\hat{t}) \phi_k(\boldsymbol{x}). \tag{4.4}$$

For interior node points $\boldsymbol{x}_k$, $k = 1, \dots, N_i$ we collocate with equation (1.13) and for node points at the near or far field boundaries, $\boldsymbol{x}_k$, $k = N_i{+}1, \dots, N$, we enforce (1.16) or (1.17), respectively. Now let $\boldsymbol{u}_i(\hat{t}) = (u(\hat{t}, \boldsymbol{x}_1), \dots,$ $u(\hat{t}, \boldsymbol{x}_{N_i}))^T$ and $\boldsymbol{u}_b(\hat{t}) = (u(\hat{t}, \boldsymbol{x}_{N_i+1}), \dots, u(\hat{t}, \boldsymbol{x}_N))^T$. Then from (4.4)

$$\begin{pmatrix} \boldsymbol{u}_i(\hat{t}) \\ \boldsymbol{u}_b(\hat{t}) \end{pmatrix} = \begin{pmatrix} A_{ii} & A_{ib} \\ A_{bi} & A_{bb} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda}_i(\hat{t}) \\ \boldsymbol{\lambda}_b(\hat{t}) \end{pmatrix}, \tag{4.5}$$

where the total coefficient matrix A has elements $a_{jk} = \phi(\varepsilon \|\vec{x}_j - \vec{x}_k\|)$ and the indicated block structure is due to the decomposition of interior and boundary node points. Furthermore, A is non-singular for standard choices of RBFs [74], and

$$\begin{aligned} \mathcal{L}\boldsymbol{u}_i(\hat{t}) &= \begin{pmatrix} B_{ii} & B_{ib} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda}_i(\hat{t}) \\ \boldsymbol{\lambda}_b(\hat{t}) \end{pmatrix} = \begin{pmatrix} B_{ii} & B_{ib} \end{pmatrix} A^{-1} \begin{pmatrix} \boldsymbol{u}_i(\hat{t}) \\ \boldsymbol{u}_b(\hat{t}) \end{pmatrix} \\ &\equiv \begin{pmatrix} C_{ii} & C_{ib} \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_i(\hat{t}) \\ \boldsymbol{u}_b(\hat{t}) \end{pmatrix}, \end{aligned} \tag{4.6}$$

where the matrix elements of B are $b_{jk} = \mathcal{L}\phi(\varepsilon \|\boldsymbol{x}_j - \boldsymbol{x}_k\|)$, for $j = 1, \dots, N_i$ and $k = 1, \dots, N$.

The eigenvalues of C_{ii} determine the stability limits for the time-steps of different time advancing methods. For the particular problem considered here, the size range of the eigenvalues is quite large, but there are no eigenvalues with positive real part. Therefore, we use the unconditionally stable BDF–2 method [42] for the time evolution of the problem. We use a constant time-step k, set $\hat{t}^n = kn$ and let $\boldsymbol{u}_i^n \approx \boldsymbol{u}_i(\hat{t}^n)$. The time-stepping scheme applied to (1.13) now yields

$$\boldsymbol{u}_i^n + \beta_1 \boldsymbol{u}_i^{n-1} + \beta_2 \boldsymbol{u}_i^{n-2} = k\beta_0 \boldsymbol{\mathcal{L}} \boldsymbol{u}_i^n, \tag{4.7}$$

where $\beta_0 = 1$, $\beta_1 = -1$, and $\beta_2 = 0$ for the first time-step and $\beta_0 = \frac{2}{3}$, $\beta_1 = -\frac{4}{3}$, and $\beta_2 = \frac{1}{3}$ for subsequent steps.

The boundary conditions are enforced at each new time level through

$$\boldsymbol{u}_b^n = \boldsymbol{g}_b^n, \tag{4.8}$$

where $\boldsymbol{g}_b^n = (g(\hat{t}^n, \boldsymbol{x}_{N_i+1}), \ldots, g(\hat{t}^n, \boldsymbol{x}_N))^T$, and

$$g(\hat{t}, \boldsymbol{x}) = \left\{ \begin{array}{rrcl} 0, & \boldsymbol{x} & = & \boldsymbol{0} \\ d^{-1} \sum_{i=1}^{d} x_i - \bar{K} \mathrm{e}^{-2\bar{r}\hat{t}}, & \|\boldsymbol{x}\|_1 & = & C. \end{array} \right. \tag{4.9}$$

Combining (4.6), (4.7), and (4.8) gives the overall scheme for advancing all unknowns one step in time,

$$\begin{pmatrix} I - k\beta_0 C_{ii} & -k\beta_0 C_{ib} \\ 0 & I \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_i^n \\ \boldsymbol{u}_b^n \end{pmatrix} = \begin{pmatrix} -\beta_1 \boldsymbol{u}_i^{n-1} - \beta_2 \boldsymbol{u}_i^{n-2} \\ \boldsymbol{g}_b^n \end{pmatrix} \tag{4.10}$$

The initial condition from (1.13) in discrete form is

$$\boldsymbol{u}_i^0 = \boldsymbol{f}_i = (\Phi(\boldsymbol{x}_1), \ldots, \Phi(\boldsymbol{x}_{N_i}))^T. \tag{4.11}$$

Due to the change in β_0 between the first and second time-step, we need to factorize the matrix block $I - k\beta_0 C_{ii}$ twice. However, this can be avoided by choosing the time-step in a special way [55]. Note that it is important to incorporate the boundary conditions into the numerical scheme as in (4.10) rather than updating them separately after each-time step. The latter approach would introduce an error in the whole domain through the global coupling of the unknowns and lead to a loss of time continuity. Figure 4.1 shows that (4.10) yields the second-order convergence expected for the BDF–2 method.

4.2 Space-time RBFs

The second main strategy for solving time-dependent PDEs is to use space-time RBFs $\Phi_j(\boldsymbol{x}, t) = \Phi(\|(\boldsymbol{x}, t) - (\boldsymbol{x}_j, t_j)\|)$, i.e., radial basis functions whose values depend on the pairwise distance between points in space-time, typically with the Euclidean space-time norm

$$\|(\boldsymbol{x}, t) - (\boldsymbol{x}_j, t_j)\| = \sqrt{\sum_{k=1}^{d} (\boldsymbol{x}^k - \boldsymbol{x}_j^k)^2 + (t - t_j)^2}. \tag{4.12}$$

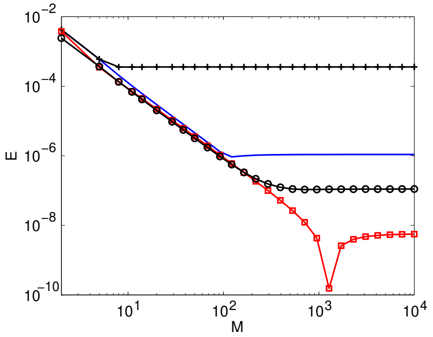


Figure 4.1: The error E as a function of the number of time-steps M. Maximum error over the whole region (+), financial norm (solid), weighted integral norm (∘) and error at $x = \bar{K}$ (□). For more details, see Paper I.

The general time-dependent equation (4.1) is then solved by global RBF space-time collocation as

$$A_t A^{-1} \boldsymbol{u} = A_{\mathcal{L}} A^{-1} \boldsymbol{u} + \boldsymbol{f}, \tag{4.13}$$

where

$$\begin{array}{rcl} (A_t)_{ij} & = & \frac{\partial \Phi}{\partial t}(\|(\boldsymbol{x}_i, t) - (\boldsymbol{x}_j, t_j)\|, \varepsilon)|_{t=t_i}, \\ (A_{\mathcal{L}})_{ij} & = & \mathcal{L}\Phi(\|(\boldsymbol{x}, t_i) - (\boldsymbol{x}_j, t_j)\|, \varepsilon)|_{\boldsymbol{x}=\boldsymbol{x}_i}, \qquad \text{and} \\ A_{ij} & = & \Phi(\|(\boldsymbol{x}_i, t_i) - (\boldsymbol{x}_j, t_j)\|, \varepsilon). \end{array} \tag{4.14}$$

In [76], different versions of space-time-interpolants are listed, for example one based on products of a spatial RBF and a time-dependent function and another one based on space-time RBFs. The space-time RBF approach is also applied to the Burgers' equation in [16] and the solution is there computed in time slabs with adaptive node redistribution via so called residual subsampling for each new time slab.

Chapter 5

The flat RBF limit

When the RBF method was first applied to interpolation and PDE problems the resulting systems of equations were solved by applying a direct solution method (RBF–Direct), i.e., by simply inverting the collocation system matrix. This remained the standard (and only available) method for quite some time. During this period it was widely observed that small shape parameter values, corresponding to flatter basis functions, led to high accuracy but severe ill-conditioning. An RBF uncertainty principle was formulated in [89], stating that there is a trade-off between accuracy and conditioning in RBF approximation methods. This unfortunately led to the widespread misconception that the ill-conditioning for small values of the shape parameter was somehow inherent to the RBF methods and so the small shape parameter range, referred to as the flat RBF limit, was considered to be of less interest. However, the ill-conditioning has to do with the fact that RBFs, while actually spanning an excellent approximation space, approach linear dependence as the shape parameter value decreases. The authors of [13] show that if stability is analysed in the function space without considering any specific basis, RBF interpolation is stable, provided that the data is distributed in a reasonable way. This in turn means that while applying the RBF–Direct method leads to divergence in the coefficients followed by numerical cancellation when the approximant is computed, the approximant itself actually stays bounded and well-behaved in exact arithmetic. The ill-conditioning is thus not intrinsic to RBF approximation, but rather related to a specific implementation of the process of finding the RBF coefficients. Indeed, the limit as $\varepsilon \to 0$ was investigated analytically in the groundbreaking article [15] and usually found to be well-behaved. This inspired the development of stable methods for computation in the flat RBF limit, starting with [34]. These methods then revealed the true behaviour of RBF approximation in the small ε range, including the existence of optimal shape parameter values that were previously hidden by the ill-conditioning of the RBF–Direct method. The flat limit behaviour and the different types of stable methods that have been developed so far for the small shape parameter range are described in the remainder of this section.

5.1 The flat RBF limit for interpolation

The flat RBF limit is investigated analytically in [15]. The authors note that although the linear RBF system becomes highly ill-conditioned and the expansion coefficients diverge with a direct implementation, the limiting interpolants often exist and take the form of polynomials. They also prove that in the 1D case and with some minor and typically satisfied constraints on the basis functions, the limit is the Lagrange interpolating polynomial. This is the lowest order interpolating polynomial, i.e., of degree $n-1$ if there are n data points. The authors also make some observations about the 2D case where they note that the limit may not exist if the nodes make a tensor-product grid and that when the limit does exist its exact form sometimes depends on the particular RBF.

The limit behaviour in higher dimensions than 1D is further examined in [35]. The authors conclude that the limit, when it exists, takes the form of a multivariate polynomial. It is also noted that the existence of the limit for most RBFs depends critically on the node point distribution and specifically how it relates to the concept of polynomial unisolvency. The authors state that under some mild conditions on the Taylor expansion coefficients of the radial basis function, and provided that the data distribution is unisolvent with respect to a basis for the set of all polynomials of degree $\leq K$, the limiting RBF interpolant is the unique interpolating polynomial of degree $\leq K$ to the given data. This is proved in different ways in [57] and in [90]. If the number of node points is not equal to the dimension of the space of polynomials of degree $\leq K$, the limit exists and is still a polynomial of degree K, but it is no longer unique. If the unisolvency condition is not fulfilled, the limiting interpolant may diverge or be a non-unique, possibly higher degree polynomial. Explicit criteria for the different types of limits are given in [57]. It is stated that the given conditions on the basis function are typically fulfilled by the standard analytic RBFs. For the condition that is related to the expansion coefficients of the RBF, this is just conjectured. It is later proved in [62], first for strictly positive definite RBFs and then also for strictly conditionally positive definite analytic RBFs. In [92] the results of the articles [57] and [62] are reached through a different method which also allows the author to investigate scenarios where the data points approach each other. The authors of [35] also conjecture that the Gaussian

RBF never diverges in the $\varepsilon \to 0$ limit. This is proved in [90].

In [29] a certain class of oscillatory RBFs, including Gaussians as a special case, is studied and found to feature unconditional non-singularity with respect to distributions of distinct nodes as well as appearing immune to divergence in the flat limit.

In [6] the limit interpolant is shown to exist and to be independent of the RBF choice when the nodes lie on a circle in $\mathbb{R}^2$, as long as the Taylor expansion of the RBF contains only non-zero coefficients. This is the case for all standard choices of RBFs. In [7] the corresponding result is derived for RBFs that have zero-coefficients in their Taylor expansion. The only difference is that in this case the limit may not exist, but when it does it is again unique and independent of the RBF. The authors give an example of a case where divergence occurs, for three points that form a orthogonal triangle in $\mathbb{R}^2$, but they also state that if node distributions are chosen randomly, divergence is unlikely.

In [97] and [60], flat limit results are presented for RBFs of finite smoothness. The RBF interpolants studied are shown to converge to polyharmonic spline interpolants in the limit. In [61] a more detailed result is proved, which states that the RBF interpolant converges to a polynomial interpolant in the flat limit when the RBF has a specified finite smoothness related to the number of given centres. If failing to fulfil this smoothness condition the RBF interpolant converges to a polyharmonic spline interpolant.

5.2 The flat RBF limit for PDEs

In Paper II we investigate the flat RBF limit for PDE problems and present a theorem for the limit behaviour along with corresponding conditions on the RBFs and node sets. The results build on the work in [57].

We define

$$N_{K,d} = \begin{pmatrix} K+d \\ K \end{pmatrix}, \tag{5.1}$$

which is the dimension of the space of polynomials of degree K in $\mathbb{R}^d$. If $N = N_{K,d}$, and the node set is unisolvent, then the (infinitely smooth) flat limit RBF interpolant reproduces the multivariate polynomial interpolant of degree K on these nodes. When we solve a PDE using the non-symmetric RBF collocation method, the RBF approximant has the same general form as the interpolant (1.2), and we can derive corresponding results for the PDE limit.

In order to express the conditions for the different limit results, we need to define two matrices, P and Q, from which we can determine polynomial unisolvency and unisolvency of the discrete PDE problem.

Let $\{p_j(\boldsymbol{x})\}_{j=1}^N$ be N linearly independent monomials of minimal degree in d dimensions. For example, for $N = 7$ and $d = 2$, we can choose $\{1, x, y, x^2, xy, y^2, x^3\}$. If $N_{K-1,d} < N \leq N_{K,d}$, the degree of $p_N(\boldsymbol{x})$ is K. As mentioned in Definition 2.2.3, the set of node points $\{x_i\}_{i=1}^N$ is polynomially unisolvent if there, for any given data at the node points, exists a unique linear combination $\sum_{j=1}^N \beta_j p_j(\boldsymbol{x})$ that interpolates the data. This is equivalent to non-singularity of the matrix

$$P = \begin{pmatrix} p_1(\boldsymbol{x}_1) & p_2(\boldsymbol{x}_1) & \cdots & p_N(\boldsymbol{x}_1) \\ p_1(\boldsymbol{x}_2) & p_2(\boldsymbol{x}_2) & \cdots & p_N(\boldsymbol{x}_2) \\ \vdots & \vdots & & \vdots \\ p_1(\boldsymbol{x}_N) & p_2(\boldsymbol{x}_N) & \cdots & p_N(\boldsymbol{x}_N) \end{pmatrix}. \tag{5.2}$$

In cases where P is singular, we instead construct a minimal non-degenerate basis [57]. Such a basis can be constructed by choosing N monomials of smallest possible degree under the constraint that they give linearly independent columns in the matrix P. The highest selected monomial degree M is then also the degree of $p_N(\boldsymbol{x})$.

Similarly, the set of node points $\{x_i\}_{i=1}^N$ satisfies unisolvency of the discrete PDE problem with respect to $\{p_j(\boldsymbol{x})\}_{j=1}^N$ if there is a unique linear combination $\sum_{j=1}^N \beta_j p_j(\boldsymbol{x})$ that satisfies the collocation conditions

$$\sum_{j=1}^N \beta_j \mathcal{L}^k p_j(\boldsymbol{x}_i^k) = f^k(\boldsymbol{x}_i^k), \quad i = 1, \ldots, N_k, \quad k = 1, \ldots, N_{\text{op}},$$

where N_{op} is the number of different operators (equations) in the PDE. This is equivalent to non-singularity of the matrix

$$Q = \begin{pmatrix} \mathcal{L}^1 p_1(\boldsymbol{x}_1^1) & \cdots & \mathcal{L}^1 p_N(\boldsymbol{x}_1^1) \\ \vdots & & \vdots \\ \mathcal{L}^{N_{\text{op}}} p_1(\boldsymbol{x}_{N_{N_{\text{op}}}}^{N_{\text{op}}}) & \cdots & \mathcal{L}^{N_{\text{op}}} p_N(\boldsymbol{x}_{N_{N_{\text{op}}}}^{N_{\text{op}}}) \end{pmatrix}. \tag{5.3}$$

We need the RBFs to fulfil three conditions in order to get the results in the theorem given below. We list these conditions and briefly discuss their validity here, but for a full explanation see [57].

(I) The RBF $\phi(r)$ can be Taylor expanded as $\phi(r) = \sum_{j=0}^\infty a_j r^{2j}$.

(II) The RBF collocation matrix is non-singular in the interval $0 < \varepsilon \leq R$, for some $R > 0$.

(III) Certain matrices $A_{p,J}$, built from the coefficients a_j in the Taylor expansion of $\phi(r)$, are non-singular for $0 \leq p \leq d$ and $0 \leq J \leq K$.

Condition (I) is true for all infinitely smooth RBFs that are commonly used. Condition (II) is likely to hold for some value of R, but the results in Section 3.1 shows that the collocation matrix can become singular at any ε, given a specific combination of PDE problem and node points. As mentioned before, condition (III) is shown to hold for these RBFs in [62].

The following theorem gives the different possibilities for the limiting RBF approximant as the shape parameter $\varepsilon \to 0$.

Theorem 2. *Assume that the RBF $\phi(r)$ fulfils conditions (I)–(III) and that the number of node points satisfies $N_{K-1,d} < N \leq N_{K,d}$. The degree of a minimal non-degenerate basis for the point set is either K for a unisolvent set or M for a non-unisolvent set. If*

(i) P and Q are non-singular, the limit exists and is a polynomial of deg K. If $N = N_{K,d}$ it is the unique degree K polynomial solution to the discrete PDE problem, otherwise the final polynomial depends on the choice of RBF.

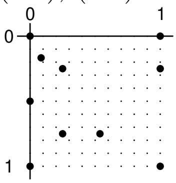
(ii) P is singular, but Q is non-singular, the limit exists and is an RBF-dependent polynomial of degree M.

(iii) P is non-singular, but Q is singular, divergence will occur unless the right hand side $\boldsymbol{f}$ of system (3.4) happens to be in the range of Q. If there is just a single null-space polynomial $n(\boldsymbol{x})$ of degree K, the divergent term is proportional to $\varepsilon^{-2}n(\boldsymbol{x})$.

(iv) P has a nullspace of dimension $m > 0$ and Q has a nullspace of dimension $p > 0$, then if $m \geq p$ the limit is likely, but not certain to exist. If it exists it is of degree M. If $m < p$ divergence is likely, but not certain.

The proof builds on the results for RBF interpolation in [57]. More details can be found in the appendix of Paper II.

Below, we give an example of type (iii) degeneracy, i.e., a node set is that is not PDE-unisolvent, for the two-dimensional Helmholtz problem given by (1.6), (1.8) and (1.9) with $m = 1$.



The $N = 10$ points are $(0,0)$, $(1/2,0)$, $(1,0)$, $(0,1)$, $(1/4,1)$, $(1,1)$, $(1/6,(2545-23\sqrt{9233})/3936)$, $(1/4,1/4)$, $(3/4,1/4)$, and $(3/4,969/1804)$. For $\kappa = 4\sqrt{246}/9$ the matrix Q has a nullspace defined by $q(\boldsymbol{x}) = -\frac{5}{32}x_2(x_2+1) + \frac{x_1}{16}(8-24x_1+3x_2+16x_1^2+4x_1x_2-7x_2^2)$.

In this case, we get divergence of order ε^{-2} as $\varepsilon \to 0$ for all RBFs that obey conditions (I)–(III). This can be observed not only in exact arithmetic, but also in for example a double precision numerical simulation. However, if we move just one of the points or change κ slightly, there is no longer a nullspace. This kind of degeneracy is very rare, since it requires very special combinations of wavenumber and node points.

If we use node sets that are not unisolvent, e.g., Cartesian nodes, both PDE approximation and interpolation are expected to behave poorly for small shape parameter values. The condition number of the linear system is larger than in the unisolvent case, and the result contains a term that diverges as $\varepsilon \to 0$.

5.3 Stable solution methods for small ε values

The methods that have been developed for RBF approximation in the flat limit range so far fall into two categories. These are rational approximation methods, which build on contour integration in the complex shape parameter plane, and methods where a well-conditioned basis is formed spanning the same space as the ill-conditioned near-flat RBFs. Brief descriptions of some methods that are stable in the small shape parameter range are given in the remainder of this section. There are also other methods involving a change of basis for a general stabilization of the problem, see for example [75, 77, 11, 12].

5.3.1 The Contour–Padé method

The Contour–Padé method, or RBF–CP, [34] was the first method that made it possible to numerically explore the behaviour of RBF approximation in the limit of flat RBFs. It allows stable computation of RBF approximants for all values of ε including the limit where $\varepsilon = 0$ and the basis functions are perfectly flat. This made it possible to numerically show that there are true optimal shape parameter values that are hidden by the ill-conditioning when computing the approximant directly. Furthermore these optimal shape parameter values can lead to errors that are orders of magnitude smaller than what is possible to achieve within the shape parameter interval that has acceptable conditioning for the direct method.

The key idea of the algorithm is to view the approximant, not as a function of a real valued shape parameter, but as an analytic function of a complex valued shape parameter. It is then written as a sum of a rational function in ε and a power series in ε, the coefficients of which are determined numerically in a stable way. This is done by first evaluating the approximant by solving the approximation problem using RBF–Direct around some circle in the complex ε-plane where the conditioning is not too high. The approximant values are then used when determining the coefficients of the previously mentioned rational function and power series. This is done by taking the inverse FFT of the approximant values around the circle and adding some extra steps to handle any poles inside the circle. Working with finite and not infinite expansions leads to some extra requirements, e.g., regarding the sampling density of the approximant on the circle and the placement of the circle. The computations have to be done for each evaluation point in the computational domain, but some of the computational work can be recycled when evaluating the approximant at new points thus decreasing the cost. The method has a large initial cost, but evaluating for many shape parameter values is almost free and the computational cost of the algorithm does not increase as $\varepsilon \to 0$. The Contour-Padé method is limited to relatively small node sets (N slightly less than a hundred in two dimensions, more in three dimensions) and is mainly meant to be a tool to investigate properties of RBF approximations and not to solve problems involving large data sets.

5.3.2 The RBF–RA method

In [103] the authors present a method, called RBF–RA, that similarly to the Contour–Padé is based on rational approximation. The interpolant is viewed as a vector-valued function of the shape parameter in the complex plain, the components of which are the interpolant values at the different evaluation points in the computational domain. The interpolant is then approximated by a vector-valued rational function, where the denominator coefficients, representing the poles, are common for all components and the numerator coefficients are specific for each component. A circle in the complex shape parameter plane is chosen, where the interpolant can be evaluated in a stable way using RBF–Direct. These values are then used when finding the coefficients of the rational approximant. For each of the interpolant components, the enforcement of the interpolation conditions results in a coupled linear system of equations for the coefficients of the rational approximant. After some manipulations of the complete system, a decoupled overdetermined system for the denominator coefficients can be solved using least squares. The remaining systems for the component specific numerator coefficients can then be solved using the denominator coefficients The rational approximant can then be used for evaluation of the interpolant for arbitrarily small shape parameter values.

The RBF–RA method has several advantages over the RBF–CP method. It has significantly higher accuracy for the same computational cost and the code is simpler involving fewer parameters and less use of complex floating point arithmetic. The algorithm for computing the poles of the rational approximation is also more robust. Like the RBF–CP method, the RBF–

RA method is limited to a relatively low number of node points (hundreds) and so its main benefits are for applications that involve relatively small node sets, such as for example RBF–FD formulas, the RBF–PUM method and domain decomposition. The method is flexible and applies to any type of smooth RBF, to any dimension and to more generalized interpolation techniques such as Hermite interpolation and appending polynomials to the basis.

5.3.3 The RBF–QR method

The RBF–QR method was first presented for computations using node distributions on the surface of a sphere [32]. The idea is to write the basis function as a series expansion in powers of the shape parameter. This is done using spherical harmonics and it results in a matrix product of a coefficient matrix C, a diagonal matrix E and a column vector consisting of spherical harmonics, i.e. $\phi = CEY$. The ill-conditioning due to the scaling of the RBFs has now been confined to the diagonal matrix E which consists of powers of the shape parameter. It is possible to multiply this expression from the left with any non-singular matrix to obtain new basis functions without changing the space that is spanned by them. If this matrix is wisely chosen, it is thus possible to create a new well-conditioned basis in the same space. In this case this is achieved by first splitting the coefficient matrix C into a QR factorization, where Q is unitary and R is upper triangular, and then multiplying the basis function expression from the left by $E_N^{-1}Q^*$, where N is the number of node points (basis function centres), E_N is the first $N \times N$ part of E, and Q^* is the Hermitian transpose of Q. The expression for the new well-conditioned basis is $\Psi = E_N^{-1}REY$. The number of independent functions associated with each shape parameter power in the expansion used determines the rate by which the powers enter in the diagonal of the E-matrix and this sequence happens to perfectly match the sequence of eigenvalue sizes for the matrix of the direct RBF method for the sphere, expressed in terms of shape parameter powers. Thus the RBF–QR method improves the conditioning at the same rate as it would otherwise have deteriorated and the method remains stable for small shape parameter values. More details on the eigenvalue patterns for the standard RBF interpolation matrix can be found in [36].

In [28] the authors present an RBF–QR method for node sets in general computational domains in one, two and three dimensions. The method requires different versions depending on the dimension of the computational domain and it can handle thousands of node points in two and three dimensions. Here the GA RBF is factorized and Taylor expanded in such a way that the number of functions, in this case monomials, corresponding to each shape parameter power again matches the number of eigenvalues of that size of the RBF–Direct method. The effect will again be that the conditioning improvement matches the deterioration related to the direct method. Since many of the monomials used in this version of the method become nearly linearly dependent as their degrees increase, the authors convert the expansion to polar coordinates. Because high powers of r also tend to be nearly linearly dependent the authors then convert some of the powers of r to Chebyshev polynomials in such a way that the improvement rate of the method still matches the deterioration rate of the direct method. Apart from the increased condition number the authors also mention another reason why errors in typical RBF implementations eventually grow with the number of nodes, namely an intrinsic ill-conditioning of spectrally accurate methods on quasi-uniform node sets leading to large errors near boundaries [83]. They therefore state that nodes must be clustered towards the boundaries in order to counteract this problem. In [58] this version of the RBF–QR method is extended to the computation of differentiation matrices and stencil weights for the solving of PDEs. An expression that reduces the computational cost of computing hyperviscosity for stencils is also presented.

In [19] a different version of RBF–QR is presented following the approach in [32] but using an eigenfunction expansion of the GA RBF instead of spherical harmonics. This strategy has some limitations for larger numbers of nodes due to the computational cost of the algorithm. In order to compensate for this, a new approach was devised involving a projection onto a reduced set of basis functions. This eliminates high-order eigenfunctions which contribute greatly to the cost but minimally to the solution.

In [54] the authors suggest a new stabilization algorithm, called HermiteGF–QR, for multivariate interpolation with isotropic or anisotropic Gaussians. This method applies to problems of any number of dimensions and builds on an expansion of isotropic or anisotropic Gaussian functions, derived from the generating function of the Hermite polynomials. A new analytic cutoff criterion for the generating function expansion is also derived and analysed. This criterion allows for adjusting the number of basis functions based on the desired accuracy in the basis.

5.3.4 The RBF–GA method

The RBF–GA method [31] is based on the Gaussian RBFs and the idea is again, as for the RBF–QR method, to find a numerically well-conditioned basis function set in the function space spanned by the ill-conditioned near-flat Gaussian RBFs. This is done by factorizing, scaling and Taylor expanding the Gaussian RBFs. Since the exact remainder term is available in closed form for any truncation of the Taylor expansion of the exponential function, any use of truncated infinite expansions can be avoided. The dominant leading Taylor terms however feature a very strong linear dependence between different node values. A new basis is therefore formed by constructing suitable linear combinations of the Taylor expanded functions. This is done is such a way that the leading Taylor coefficients are cancelled out analytically, thus allowing these terms to be omitted rather than being cancelled numerically which would lead to a loss of significant digits.

For small shape parameter values the condition number stays favourable and changes very little as the shape parameter value decreases. Moving from moderate to large shape parameter values, the condition number increases rapidly. The reason for this has to do with the behaviour of the remainder term in the Taylor expansion which for large shape parameter values leads to a degradation of the basis function independence.

The RBF–GA and the RBF–QR methods both have the disadvantage of being limited to Gaussian-type RBFs. The RBF–GA method is easier to implement and computationally faster, but less robust than the RBF–QR method.

5.3.5 The Hilbert–Schmidt SVD method

The so called Hilbert–Schmidt SVD method is presented in [8] for general positive definite kernels (including RBFs). The authors point out that kernel-based interpolation, approximation and PDE problems can be solved without ever forming the kernel matrix. In addition to that, a closed form of the kernel does not even have to be known. It is enough to have a series representation of it. The authors use a Hilbert–Schmidt series expansion of the kernel to find a decomposition, the so called Hilbert–Schmidt SVD, of the kernel matrix without actually forming this matrix. For N data points this results in a new basis consisting of the N first eigenfunctions from the Hilbert–Schmidt expansion plus a correction in the form of a linear combination of all of the infinitely many eigenfunctions with index greater than N. This correction is guaranteed to make a finite contribution because the original series is uniformly convergent, and for practical purposes the correction is truncated at a suitable point. The resulting basis is better conditioned than the original one, since it consists of small corrections to the N first eigenfunctions which are orthogonal with respect to the $L_2(\Omega, \rho)$ inner product. The authors also introduce the family of univariate, iterated Brownian bridge kernels, which generalize the Brownian bridge kernel which plays an important role in many statistics and finance applications. Similarly to the Matérn functions, the iterated Brownian bridge kernels are defined using two parameters that determine the kernel smoothness and flatness respectively. The smoothness and flatness of these kernels only affect the Hilbert–Schmidt eigenvalues, while the eigenfunctions are the same for all the kernels in this family. These kernels result in an impressively short Matlab code, consisting of just a few lines, for the entire interpolation problem. The authors state that the Hilbert-Schmidt SVD method can be used to perform accurate and stable interpolation with positive definite kernels even in their flat limit and they demonstrate this for the iterated Brownian bridge kernels, whose flat limits are piecewise polynomial splines.

Chapter 6

Convergence properties and error estimates

Different theoretical error estimates for RBF interpolation have been derived and presented in the literature. These estimates sometimes require the interpolated function to belong to the native space of the RBF, which is defined as

$$\mathcal{N}_{\phi_\varepsilon}(\mathbb{R}^d) = \left\{ f \in C(\mathbb{R}^d) \cap L_2(\mathbb{R}^d) : \|f\|_{\mathcal{N}_{\phi_\varepsilon}} := \int_{\mathbb{R}^d} \frac{|\hat{f}(\omega)|^2}{|\hat{\phi}_\varepsilon(\omega)|}\, d\omega < \infty \right\}, \tag{6.1}$$

where $\hat{f}$ and $\hat{\phi}_\varepsilon$ are the Fourier transforms of f and ϕ_ε, respectively. This means that in order for a function to belong to the native space, the square of the Fourier transform of the function must decay faster than the Fourier transform of the RBF. The native space depends on the shape parameter and for all the infinitely smooth RBFs, smaller shape parameter values lead to a faster decay of their respective Fourier transforms and thus to a smaller native space. Moreover, the Fourier transforms of the infinitely smooth basis functions decay exponentially which means that many analytic functions are excluded from the native space [80]. However, the differences in the theoretical error estimates between functions inside and outside the native space do not necessarily lead to a dramatic difference in the interpolation error. In [80] the author states that even if a function f does not belong to the native space, RBF interpolants may still converge to f under certain assumptions on the node distribution. The author also proves that under mild conditions, IQ RBF interpolants of one-dimensional functions that are analytic inside the strip $|\text{Im}(z)| < (1/2\varepsilon)$, converge exponentially. In [53] the authors test the convergence of a GA Galerkin–RBF approximation of the one-dimensional harmonic oscillator for shape parameter values around the breaking point where the solution goes from being included in, to being excluded from the native space of the GA RBF. The convergence is exponential and no significant changes can be seen in the error curves except for the ill-conditioning which is as usual worse for smaller shape parameter values.

In Paper II we derive error bounds and approximations of the error for non-symmetric RBF collocation for the Helmholtz problems (1.6)–(1.11) in 1D and 2D. We also investigate the validity of our results through numerical experiments. Our findings are summarized in the remainder of this section.

6.1 General error estimates using Green's functions

Consider the following general PDE problem

$$\mathcal{L}^i u(\boldsymbol{x}) = f^i(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega^i, \quad i = 1, \ldots, N_{\text{op}}, \tag{6.2}$$

where $\mathcal{L}^i$ is a linear operator, u is the solution function, f^i is a given function, $\boldsymbol{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$, and $\Omega^i \subseteq \bar{\Omega}$ is a region in the computational domain or a boundary segment.

We define the error function as the difference between the RBF approximant and the exact solution to the PDE problem (6.2)

$$e(\boldsymbol{x}) = s(\boldsymbol{x}) - u(\boldsymbol{x}). \tag{6.3}$$

For interpolation, the error and the residual are the same, and the error can be explicitly computed if the function $u(\boldsymbol{x})$ is known. For PDEs we can compute the residual for each operator and the error is governed by the same type of PDE as the solution

$$\mathcal{L}^i e(\boldsymbol{x}) = \mathcal{L}^i s(\boldsymbol{x}) - f^i(\boldsymbol{x}) \equiv r^i(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega^i, \quad i = 1, \ldots, N_{\text{op}}, \tag{6.4}$$

where r^i are residuals. The error could be computed by solving this PDE, but the residuals are zero at the collocation points and thus highly oscillatory which means that error approximation via the PDE (6.4) is more computationally expensive than solving the original PDE. We therefore instead formulate *a posteriori* error estimates in terms of the residual, by using Green's functions that satisfy the boundary conditions.

For the one-dimensional Helmholtz problem (1.6) with boundary conditions (1.7), the Green's function is given by

$$G(x, \xi) = \frac{i}{2\kappa} e^{i\kappa|x-\xi|}, \tag{6.5}$$

and this results in the estimate

$$\|e\|_\infty \leq \int_0^1 |G(x,\xi)||r(x)|\,dx = \frac{1}{2\kappa}\int_0^1 |r(x)|dx \leq \frac{1}{2\kappa}\|r\|_\infty. \tag{6.6}$$

For the two-dimensional Helmholtz problem (1.6) in a rectangular domain with boundary conditions (1.8)–(1.9), the Green's function is

$$G(\underline{x},\underline{\xi}) = \sum_{m=1}^{\infty} \frac{i}{2\beta_m} e^{i\beta_m|x_2-\xi_2|}\psi_m(x_1)\psi_m(\xi_1), \tag{6.7}$$

and we get the following error estimate

$$\begin{aligned}\|e\|_\infty &\leq \sum_{m=1}^{\mu_0} \frac{1}{\sqrt{2}\beta_m}\int_0^1 |r_m(x_2)|\,dx_2 \\ &+ \sum_{m=\mu_0+1}^{\infty} \frac{\sqrt{2}}{|\beta_m|^2}(1-e^{-\frac{1}{2}|\beta_m|})\int_0^1 |r_m(x_2)|\,dx_2 \\ &\leq \sum_{m=1}^{\mu_0} \frac{1}{\sqrt{2}\beta_m}\|r_m\|_\infty + \sum_{m=\mu_0+1}^{\infty} \frac{\sqrt{2}}{|\beta_m|^2}(1-e^{-\frac{1}{2}|\beta_m|})\|r_m\|_\infty. \end{aligned} \tag{6.8}$$

For the two-dimensional Helmholtz problem given by (1.6) and (1.10)–(1.11) in a domain with curved boundaries, we cannot provide an explicit Green's function. However, if we view the curved domain as a sequence of narrow almost rectangular domains, we can modify the previous estimate to get the following heuristic approximation of the error

$$\begin{aligned}\|e\|_\infty &\approx \sum_{m=1}^{\infty} \int_{\Re e(\beta_m)>0} \frac{|r_m(x_2)|}{\sqrt{2}\beta_m}\,dx_2 \\ &+ \sum_{m=1}^{\infty} \int_{\Im m(\beta_m)>0} \frac{\sqrt{2}}{|\beta_m|^2}(1-e^{-\frac{1}{2}|\beta_m|})|r_m(x_2)|\,dx_2. \end{aligned} \tag{6.9}$$

We evaluate the performance of this error approximation in Section 6.5, and we use it as an aid in the choice of shape parameter values in Section 7.1.3.

6.2 Convergence properties for small ε

As discussed in Section 5, we approach the polynomial limit, $s(\boldsymbol{x}) = p(\boldsymbol{x})$, as $\varepsilon \to 0$. In Paper II we follow the steps for the proof of the polynomial interpolation error [41, pp. 43–44], and get the following estimate for the polynomial residual.

Theorem 3. *For a one-dimensional linear PDE problem*

$$\begin{cases} \mathcal{L}^1 u(x) &= f^1(x), \quad x_1 < x < x_N, \\ \mathcal{L}^2 u(x) &= f^2(x), \quad x = x_1, \\ \mathcal{L}^3 u(x) &= f^3(x), \quad x = x_N, \end{cases}$$

with a polynomial solution $p(x)$ determined through collocation at the nodes x_i, $i = 1, \ldots, N$ the residual $r(x) = \mathcal{L}^1 p(x) - f(x)$ has the form

$$r(x) = \frac{\prod_{j=2}^{N-1}(x - x_j)}{(N-2)!} r^{(N-2)}(\xi),$$

where $\xi \in (x_1, x_N)$. For equispaced points, $x_{j+1} - x_j = h$, this can be estimated by

$$|r(x)| \leq \frac{h^{N-2}}{N-2} \max_{\xi \in (x_1, x_N)} |r^{(N-2)}(\xi)|.$$

By inserting this residual estimate in the error estimate (6.6) for the one-dimensional Helmholtz problem (1.6)–(1.7), we get

$$\|e\|_\infty \leq \frac{1}{2\kappa} \frac{h^{N-2}}{N-2} \|r^{(N-2)}\|_\infty. \tag{6.10}$$

In the flat limit, the residual is $r(x) = -p''(x) - \kappa^2 p(x)$, where $p(x)$ is the limit polynomial of degree $N-1$. Then $r^{(N-2)}(x) = -\kappa^2 p^{(N-2)}(x)$. Under the assumption that $p(x) \approx u(x) = \exp(i\kappa x)$, we get $|p^{(N-2)}| \approx |\frac{d^{N-2} \exp(i\kappa x)}{dx^{N-2}}| = \kappa^{N-2}$. We use this to get an approximate expression for the error in the limit

$$\|e\|_\infty \approx \frac{1}{2\kappa} \frac{h^{N-2}}{N-2} \kappa^2 \kappa^{N-2} = \frac{\kappa(\kappa h)^{N-2}}{2(N-2)} \approx \frac{1}{2}(\kappa h)^{N-1}. \tag{6.11}$$

Note that the quantity κh is small only if the problem is adequately resolved.

For the two-dimensional Helmholtz problem given by (1.6) and (1.8)–(1.9) in a rectangular domain, the limit polynomial is zero at the interior node points. To get an estimate for the residual in terms of its derivatives, we can therefore try using a sampling inequality such as [69, Theorem 3.5], which says that for all $h \leq h_0$,

$$\|r\|_\infty \leq C_k h^k \sum_{|\sigma|=k} \|D^\sigma r\|_\infty, \tag{6.12}$$

where h_0 depends on the geometry of Ω. The condition $h \leq h_0$ is too restrictive for our case, but from practical experience the result holds also for larger h, and we will therefore use (6.12) to approximate the residual.

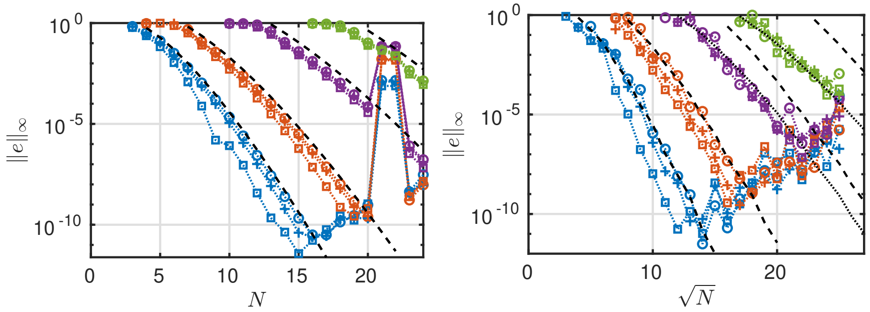


Figure 6.1: The computed errors using the Gaussian RBF and the RBF–QR method for $\varepsilon = 0.5$ (□), $\varepsilon = 0.25$ (+), and $\varepsilon = 0.01$ (○) for $\kappa = \pi$, 2π, 4π, 6π, from left to right, together with the approximation $\|e\|_\infty \approx \frac{1}{2}(\kappa h)^{N-1}$ for the one-dimensional problem (left), and for $\kappa = 1.2\pi$, 2.4π, 4.8π, 7.2π, from left to right, together with the approximation $\|e\|_\infty \approx (\kappa h)^K$ for the two-dimensional problem (right) (dashed curves). For $\kappa = 4.8\pi$ and 7.2π in the two-dimensional case, we also show the error approximation using $C = 1/40$ and $C = 1/800$, respectively (dotted lines).

In this case, using that $r(\boldsymbol{x}) = -\Delta p(\boldsymbol{x}) - \kappa^2 p(\boldsymbol{x})$, and, for $|\sigma| = K-1$, $D^\sigma r(\boldsymbol{x}) = -\kappa^2 D^\sigma p(\boldsymbol{x}) \approx -\kappa^2 D^\sigma u(\boldsymbol{x})$, we get

$$\|r\|_\infty \leq C_{K-1} h^{K-1}\kappa^2 \sum_{|\sigma|=K-1} \beta_1^{\sigma_2}\alpha_1^{\sigma_1} \leq C_{K-1}\kappa^2 K(\kappa h)^{K-1}. \tag{6.13}$$

Combining the approximate expression for the residual with the error estimate (6.8) restricted to the first mode (scaled by $1/\sqrt{2}$) gives

$$\|e\|_\infty \approx \frac{1}{2|\beta_1|} C_{K-1}\kappa^2 K(\kappa h)^{K-1}. \tag{6.14}$$

Numerical experiments show that $KC_{K-1} = C/(K-1)$ provides the appropriate behaviour with respect to N. (Both K and h are coupled with N). This leads to

$$\|e\|_\infty \approx \frac{C\kappa^2(\kappa h)^{K-1}}{2|\beta_1|(K-1)} \approx \tilde{C}(\kappa h)^K. \tag{6.15}$$

From our numerical investigations, we can see that the behaviour of the computed errors for the one-dimensional and two-dimensional problems agree well with the derived error approximations, see Figure 6.1.

6.3 Convergence properties for larger ε

The convergence of a PDE approximation can be expressed in terms of the approximation properties of the interpolant (consistency error) and a stability term [91, 93, 59]. The consistency error of the PDE operator can be expressed as

$$\mathcal{E}_{\mathcal{L}} = \mathcal{L}(I_h(u) - u),$$

where $I_h(u)$ interpolates u on a node set with fill distance h, where the fill distance is defined as

$$h = h_{\mathcal{X},\Omega} = \sup_{\boldsymbol{x}\in\Omega} \min_{\boldsymbol{x}_j\in\mathcal{X}} \|\boldsymbol{x} - \boldsymbol{x}_j\|_2. \tag{6.16}$$

Several authors have derived exponential convergence results for RBF interpolation [84, 71, 68, 4, 105, 85]. The first papers describe estimates for interpolation errors, while [85] also provides estimates for derivatives of functions with many zeros, such as the interpolation error.

Based on the results in [85] we assume that the error has the form

$$\|e\|_\infty = A_M \exp(-C_M f(h)), \tag{6.17}$$

with $C_M > 0$. The native space norm has been absorbed into the constant A_M and the form of $f(h)$ depends on the type of domain and the PDE operator in question. If this assumption is correct, a plot of the logarithm of the error against $f(h)$ should result in a straight line. From Figure 6.2, it is clear that $f(h) = 1/h$ is a better fit compared with $f(h) = 1/\sqrt{h}$. The dashed lines correspond to a fit of the model with $f(h) = 1/h$ to the actual errors, where the results suffering from ill-conditioning effects have been ignored.

6.4 Convergence as a function of the shape parameter

The results in Section 6.3 hold for a fixed value of ε. However, using a shape parameter $\varepsilon_0 \neq 1$ for an interpolation problem in the domain Ω with fill distance h is equivalent to using a shape parameter $\varepsilon_1 = 1$ for a problem in the scaled domain $\varepsilon_0\Omega$ with fill distance $\varepsilon_0 h$. This can be understood by noting that $\phi(\varepsilon_0\|\boldsymbol{x}_i - \boldsymbol{x}_j\|) = \phi(1 \cdot \|\varepsilon_0\boldsymbol{x}_i - \varepsilon_0\boldsymbol{x}_j\|)$. This means that the native space norm is the same in both cases, and so are the errors.

Letting the constants A_M and C_M in the error estimate for a specific domain Ω and shape parameter ε be denoted by $A_M(\Omega,\varepsilon)$ and $C_M(\Omega,\varepsilon)$, we have

$$A_M(\Omega,\varepsilon)e^{-C_M(\Omega,\varepsilon)/h} = A_M(\varepsilon\Omega,1)e^{-C_M(\varepsilon\Omega,1)/(\varepsilon h)}. \tag{6.18}$$

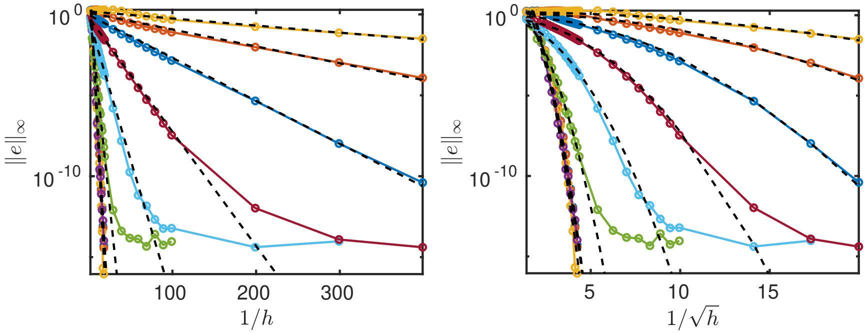


Figure 6.2: The error in the one-dimensional Helmholtz solution when multiquadric RBFs are used as a function of $1/h$ (left) and $1/\sqrt{h}$ (right) for shape parameters $\varepsilon = 10^{-2+\frac{4}{9}q}$, $q = 1, \ldots, 9$ (left to right). The dashed black lines/curves correspond to a fit of $\|e\|_\infty = A_M \exp(-C_M/h)$ to the error data (in both cases).

Hence, the convergence rate for a fixed value of ε increases for smaller shape parameter values outside the small shape parameter range.

Figures 6.3 and 6.4 show the error as a function of ε for two one-dimensional problems, and one two-dimensional problem, respectively. The error curves are typical for smooth solution functions. Starting from a large shape parameter and moving towards smaller values, the error first decreases rapidly then reaches an optimal region, and finally levels out at the polynomial approximation error.

The convergence curves for different shape parameter choices given by $\varepsilon = Ch^\beta$ and different exponents β are shown as dashed lines in Figures 6.3 and 6.4. As expected, the stationary choice, $\beta = -1$ levels out as N increases. For $\beta > -1$ we get convergence along different paths. Choosing $\beta = 0$ corresponds to the exponential convergence case for fixed shape parameter values. For these particular Helmholtz problems, the curve with $\varepsilon = Ch^{3/2}$ corresponds well with the optimal shape parameter values. For other problems the relation would be different. For the two-dimensional problem, several terms in the error interact, leading to more irregular curves [57]. The overall results of the different shape parameter choices are still very similar to the results of the one-dimensional case.

If we assume that $-C_M(\varepsilon\Omega, 1)$ in (6.18) does not vary a lot with ε, which we have verified through experiments, we can provide a convergence rate for the scaled ε convergence case. If we have exponential convergence as $1/\varepsilon h$ and $\varepsilon = Ch^\beta$ we get

$$\|e\|_\infty = A_M^\varepsilon e^{-C_M^\varepsilon/h^{\beta+1}}, \quad -1 < \beta \leq 0, \tag{6.19}$$

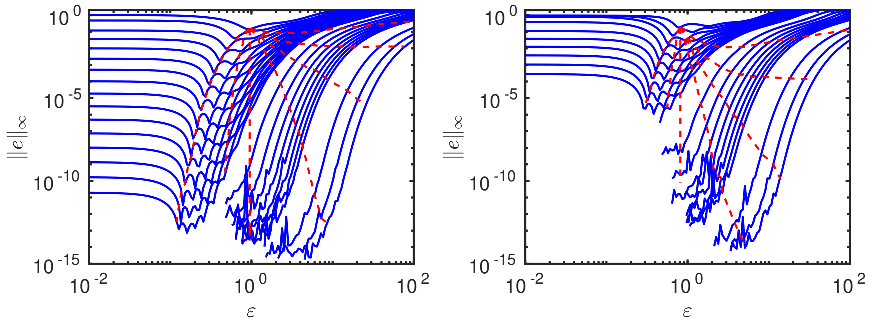


Figure 6.3: The maximum error as a function of ε for $\kappa = 2\pi$ (left) and $\kappa = 4\pi$ (right) using multiquadric RBFs for the Helmholtz problem in 1D. The number of node points is from top to bottom $N = 6, 7, \ldots, 21, 30, 40, \ldots, 100, 200, 300, 400$ in the left subfigure, and $N = 10, 11, \ldots, 20, 30, \ldots, 100, 200, 300, 400$ in the right subfigure. The dashed lines show how the error curves are traversed if the shape parameter is chosen as $\varepsilon = Ch^\beta$, with $\beta = \frac{3}{2}, \frac{1}{2}, 0, -\frac{1}{2}, -\frac{3}{4}, -1, -\frac{3}{2}$ from left to right.

where $C_M^\varepsilon > 0$ and the superscript indicates the potential ε-dependence. The validity of this is expression is confirmed numerically in Section 6.5.

6.5 Numerical validation of our error formulas

In this section we validate our error formulas for the two-dimensional Helmholtz problem with curved boundaries (1.6), (1.10), (1.11), using shape parameter values given by $\varepsilon = C/\sqrt{h}$. With this choice of shape parameter scaling, the error should according to equation (6.19) be of the form

$$\|e\|_\infty = A_M \exp(-C_M/\sqrt{h}). \tag{6.20}$$

Figure 6.5 shows the relative error and the relative error estimate based on (6.9) against $1/\sqrt{h}$. Lines describing the error formula (6.20) have been fitted to the data and the slopes C_M are 0.78 for the error and 0.75 for the error estimate. This means that the relative error estimate is a good fit for the trend in the relative error, even if the constant is not precise. The constant A_M is 3.0 times larger for the error estimate than for the error. Based on experiments we expect A_M to be problem and/or parameter dependent.

We also used the error estimate (6.9) for the two-dimensional Helmholtz problem with curved boundaries for larger wavenumbers. We did this by solving the problem for three node sets of different sizes. We computed the relative errors of the coarser solutions with respect to the finest solution and

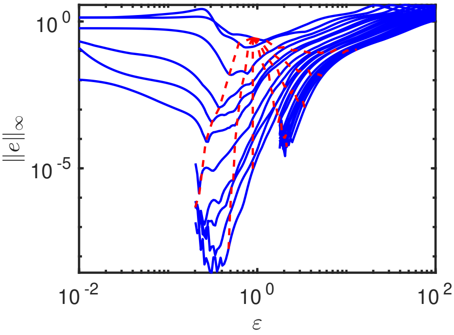


Figure 6.4: The maximum error as a function of ε for $\kappa = 2.2\pi$ for the two-dimensional Helmholtz problem on a rectangle, using multiquadric RBFs. The number of node points is from top to bottom $N \approx n^2$, for $n = 3, \ldots, 25$. The dashed lines show how the error curves are traversed if the shape parameter is chosen as $\varepsilon = Ch^\beta$, with $\beta = \frac{3}{2}, \frac{1}{2}, 0, -\frac{1}{2}, -\frac{3}{4}, -1, -\frac{3}{2}$ from left to right.

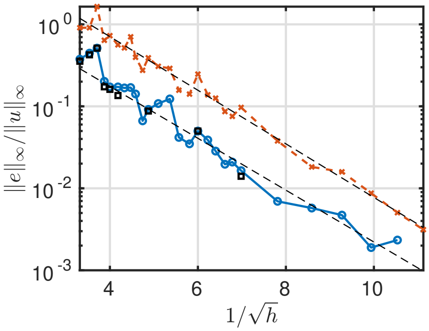


Figure 6.5: The relative error estimate (6.9) (×) and the relative error against the reference solution as a function of $1/\sqrt{h}$. The dashed lines represent the error formula (6.20) fitted to the data points.

then compared the errors with the error estimates to find the approximate ratio between the estimates and the real errors. Finally we used this ratio to obtain an improved error estimate for the finest solution.

Chapter 7

Global RBF collocation in practice

We begin this section with some general guidelines on global RBF collocation. The first choices to consider are which collocation method and RBF type to use. A general recommendation is to use the non-symmetric collocation method for time dependent problems with explicit time-stepping and the symmetric collocation method for time-independent PDEs and possibly also for time-dependent PDEs with implicit time-stepping [18]. In practice though many researchers prefer to use the more flexible and more easily implemented non-symmetric method even when the collocation matrix needs to be inverted. This usually causes no problems as the cases that lead to singularity seem to be rare [46]. As mentioned in Section 2 when it comes to the RBF type, the infinitely smooth RBFs have the highest potential for fast convergence for smooth data. On the other hand they often lead to ill-conditioned problems, especially for the shape parameter values that tend to yield the highest accuracy. The piecewise smooth RBFs lead to less ill-conditioned problems, but they only yield algebraic convergence. These RBFs are especially relevant for interpolation and statistics applications but maybe less so for PDEs [23]. For interpolation problems there is most likely little point in using RBFs that are much smoother than the given data, see for example Guideline 7.6. in [94]. This guideline states that the smoothness of the interpolated function determines the attainable approximation rate for non-stationary interpolation. However, the situation can be different when it comes to PDEs [33]. Regarding the shape optimization of the infinitely smooth RBFs, the smaller shape parameter range tends to result in higher accuracy provided that the problem does not become too ill-conditioned and that the solution does not have large local gradients. An intuitive reason why solutions with large local gradients require larger shape parameter values is that it is difficult to achieve local peaks by combining very flat global functions. The shape of some RBFs can also be optimized through a second parameter. For the generalized MQ RBFs, it is shown in [102] that it is more important to optimize the MQ exponent (β in Table 2.2) than to optimize the shape parameter ε in order to accelerate convergence. The optimal value of the exponent increases as a function of the number of node points. Finally, when using RBF methods to solve larger systems it is, just as for other methods, a good idea to consider additional approaches such as domain decomposition and iterative methods.

7.1 How to choose an appropriate shape parameter value

What strategy to use when choosing the shape parameter depends on the task to be performed and the available data. If the idea is to theoretically study the accuracy of the RBF solution and the exact solution is known, a trial and error approach can be used. A shape parameter interval is then chosen and the problem is solved repeatedly for different shape parameter values using either the direct method or when necessary one of the stable methods described in Section 5.3. This is of course computationally costly and as the optimal shape parameter depends on the specific problem to be solved it is not possible to reuse a particular optimal value for other problems and motivating the high cost that way. However, if the idea is to actually study the behaviour of a particular RBF method for different shape parameter values and finding an optimal value as a result of this, trial and error is of course the most natural approach. For all other purposes than studying the method itself, approximating a solution is quite pointless if the exact solution is already known. A shape parameter can still be chosen by trial and error even when the exact solution is not known but then the method becomes rather subjective since there is no definite way to decide what the best shape parameter value is. Since it is generally known that the direct method becomes increasingly ill-conditioned as the shape parameter value decreases, one approach suggested in [18] for solution in Matlab is to use the smallest shape parameter value for which there is no near-singular warning. Despite the ambiguities and high cost the trial and error approach is widely used, mainly because of its simplicity.

7.1.1 Leave-one-out cross validation (LOOCV)

A shape parameter strategy that requires no known exact solution is the cross validation method. This technique is suggested in [40] in the context of solution of elliptic PDEs with the dual reciprocity method based on RBF interpolation. In [86] a version called leave-one-out cross validation (LOOCV) that is frequently used in statistics is described and applied to RBF interpolation. The idea is to compare the given data at each node point with the value of the interpolant based on all other node points, excluding the comparison node. The result is a vector with an approximate interpolation error from which an error norm can be computed. The strategy is repeated for different shape parameter values and the shape parameter that minimizes the norm of the approximate error is chosen. We have the following RBF interpolant to the given data $(f_1, \ldots, f_{k-1}, f_{k+1}, \ldots, f_N)$:

$$s_k(\boldsymbol{x}, \varepsilon) = \sum_{j=1, j\neq k}^{N} \lambda_j^k \Phi(\|\boldsymbol{x} - \boldsymbol{x}_j\|, \varepsilon), \tag{7.1}$$

satisfying

$$s_k(\boldsymbol{x}_i) = f_i, \qquad i = 1, \ldots, k-1, k+1, \ldots N. \tag{7.2}$$

The so called cost vector, here based on the approximate interpolation error at each left out node point, is given by

$$E_k = f_k - s_k(\boldsymbol{x}_k), \qquad k = 1, \ldots, N. \tag{7.3}$$

Naively implemented the LOOCV strategy is computationally expensive as it involves the solution of N systems with $N-1$ unknowns requiring $\mathcal{O}(N^4)$ operations for each shape parameter value. In [86] however it is shown that the error vector (7.3) can alternatively be computed by

$$E_k = \frac{\lambda_k}{A_{kk}^{-1}}, \tag{7.4}$$

where the coefficients λ_k and matrix A correspond to the full interpolation problem including all data points. This means that only one system needs to be solved for each shape parameter value, reducing the cost to $\mathcal{O}(N^3)$. Using (7.4) thus results in the same cost for the LOOCV strategy as for the trial and error approach.

For PDE problems the cost vector needs to be modified as there is no way to directly compare the approximant with an exact solution or given data. Using the residual to predict the error behaviour for different shape parameter values when solving PDE problems through collocation was suggested in [9]. Basing the cost vector on the residual is also the most straightforward way to modify LOOCV to fit the PDE framework [22]. The cost vector formulas corresponding to (7.3) and (7.4) for non-symmetric collocation applied to the PDE (3.1), with $N_{\mathcal{I}}$ interior points and a total of N node points are then given by

$$E_k = \begin{cases} f_k - \mathcal{L}s_k(\boldsymbol{x}_k), & k = 1, \dots, N_{\mathcal{I}}, \\ g_k - \mathcal{L}_{\mathcal{B}}s_k(\boldsymbol{x}_k), & k = N_{\mathcal{I}} + 1, \dots, N, \end{cases} \tag{7.5}$$

and

$$E_k = \frac{\lambda_k}{K^{-1}_{kk}}, \tag{7.6}$$

where

$$K = \begin{pmatrix} L \\ B \end{pmatrix} \tag{7.7}$$

is the non-symmetric collocation matrix in (3.4).

A different cost vector based on comparisons with the PDE operator applied to each basis function was defined for the RBF–PS method in [20]. This version however does not take the dependence of the approximation on the given data into account which is a disadvantage.

The LOOCV approach is still rather expensive even when using formula (7.6). In [106] a variation of the LOOCV approach referred to as a doubly stochastic radial basis function method (DSRBF) is introduced which decreases the overhead cost associated with the LOOCV shape parameter selection from $\mathcal{O}(N^3)$ to $\mathcal{O}(N^2)$ operations. The idea is to work with an overdetermined system with data given in N points and to perform a number of observation experiments using different randomly selected subsets of N collocation points. For each subset of data points the coefficients and collocation matrix inverse in (7.6) are then approximated rather than computed by solution of the corresponding system of equations. By keeping the number of observation experiments limited it is possible to keep the cost down to the $\mathcal{O}(N^2)$ operations of the approximate solution process. Each observation experiment results in an estimated optimal shape parameter and the mean of these estimated values is then used to compute N randomly distributed shape parameter values, each corresponding to one specific basis function. The full problem is then solved using these shape parameter values.

7.1.2 Variable shape parameter values

Using variable shape parameters, i.e., allowing different shape parameter values for different basis functions, can improve the conditioning of the problem as well as the accuracy of the solution [36, 102, 88] and it can also decrease effects of the Gibbs phenomenon associated with the approximation of discontinuous functions [88]. There is less theoretical knowledge in the case of variable shape parameters, but sufficient conditions for the non-singularity of the interpolation matrix A in (1.5) are provided in [3]. One way of choosing the variable shape parameters is through the DSRBF described in Section 7.1.1, but they can also be chosen through greedy algorithms [88]. Another strategy is to choose a shape parameter and scale it with the inverse distance to the nearest node point in order to achieve appropriate variable shape parameter values [36].

7.1.3 Some shape parameter strategies from our research

In Paper I we vary the shape parameter as $\varepsilon = 1 + N/20$, when solving the 1D Black–Scholes equation, see Figures 7.1 and 7.5. This formula works reasonably well for a limited range of problem sizes. However for other values of N other choices are better, see Figure 7.2. For the 2D Black–Scholes equation in Paper I, we use shape parameter values that have been optimized locally in a small interval close to the ill-conditioned zone for each experiment.

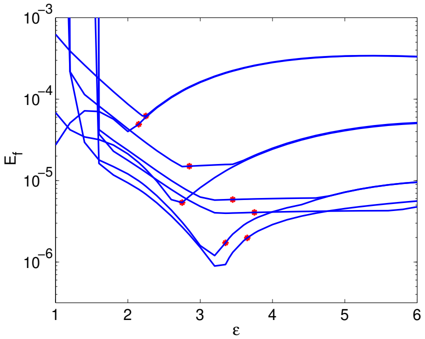


Figure 7.1: The financial error norm as function of ε for the 1D Black–Scholes equation with $N \in [20, 60]$. The stars show $\varepsilon = 1 + N/20$. Note that the ill-conditioning hides the true optimal shape parameter values.

For the Helmholtz equation in Paper II, the results of different shape parameter choices can be seen in Figures 6.3 and 6.4. We also use the residual-based error estimate (6.9) to aid us in finding appropriate shape parameter values. As shown in Section 6.4, a practical way to achieve convergence in spite of the ill-conditioning is to choose the shape parameter as $\varepsilon = Ch^{\beta}$, with $\beta > -1$. We use $\beta = -1/2$, for a trade-off between convergence rate and conditioning, but we also need to know the value of C.

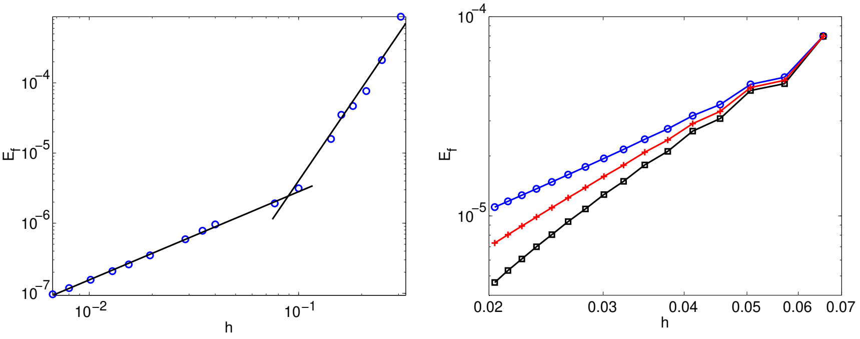


Figure 7.2: Different shape parameter choices for the 1D Black–Scholes equation. Left: Financial error norm for the choice $\varepsilon = 1 + N/20$. Right: Financial error norm for $\varepsilon = 1 + N/20$ (o), $\varepsilon = a + bN^{3/4}$ (+) and $\varepsilon = c + dN^{1/2}$ (□), where a, b, c, and d are constants.

Compared with the full solution, solving a much less resolved problem a few times for different shape parameter values is computationally affordable. We use the error estimate (6.9) to find the best shape parameter values for this smaller problem, and from there the C to use. Figure 7.3 shows the relative error estimate as well as the relative ℓ_2-norm of the residual together with the actual error against the reference solution. In the first example, both the error estimate and the residual norm are good indicators of the true optimum. In the second example, the minimum for the error estimate is a bit higher than the true value.

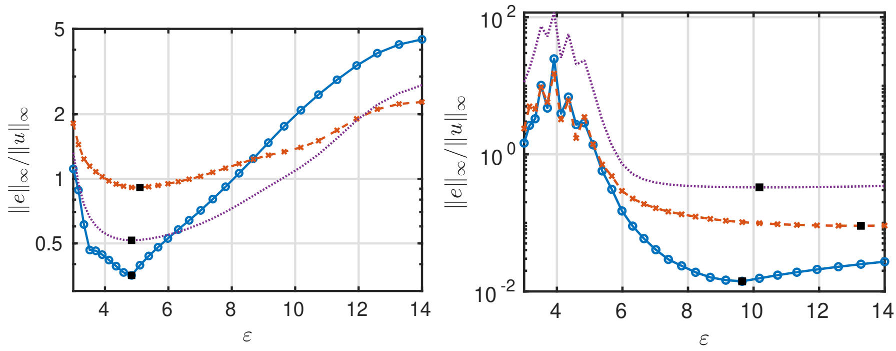


Figure 7.3: The error estimate (6.9) (×), the ℓ_2-norm of the residual (dotted line) and the error against a highly resolved reference solution (∘) for the 10×12 (left) and 40×50 (right) node sets. The minima are indicated by black squares. Computed C-values corresponding to the average of the two estimates were $C = 1.5$ (left) and $C = 1.7$ (right).

We also tried to use residual-based LOOCV on the Helmholtz problems in this paper, but the preliminary results suggested that our error approximation strategy was more successful in finding relatively optimal shape parameter values in this case.

7.2 How to distribute the node points

As mentioned already in the introduction, one of the advantages of RBF methods is their meshless nature. This feature not only makes these methods more flexible in terms of the geometry of the computational domain but it also offers a potential for accuracy improvement by adapting the node distribution to the computational problem at hand. Typical regions where one might want a denser node placement are boundaries, where the errors of RBF methods tends to be the largest, and any region where the interpolated function or approximated solution of a PDE might have finer features that need more nodes to be properly resolved. The change in node density between different regions should be smooth and gradual in order to avoid artificial effects in the solution [24].

In terms of node distributions in 2D in general without any local refinement, a hexagonal or approximately hexagonal node placement has been found to be optimal [47, 48, 27]. So called Halton points often perform relatively well. Halton node distributions are quasi-random sets with points from the Halton sequence. They tend to fill up the computational domain in a uniform manner, provided that the number of node points is large enough relative to the dimension of the domain [18]. Distributions for which the coordinates have been calculated from independent uniformly distributed random numbers are, however, not recommended. The reason for this is that they tend to result in quite severe local node clusterings in a way that does not reflect PDE or data behaviour, while other areas can lack nodes, which leads to large errors and condition numbers. When using a constant shape parameter, placing nodes extremely close to each other is of course not a good idea in general because it leads to severe ill-conditioning. Cartesian node distributions sometimes perform reasonably well, but they can often be worse than Halton points. One reason for this is that the approximation quality of Cartesian node layouts is different in different spatial directions [27]. Cartesian node distributions are also non-unisolvent in the flat RBF limit, leading to higher degree limit polynomials and worsened conditioning, see Section 5. It is also proved in [83] that no method can be simultaneously stable and exponentially convergent on equispaced nodes. Often though the difference in accuracy between different (reasonable) node distributions is not that dramatic.

7.2.1 How to mitigate or bypass the Runge phenomenon

The Runge phenomenon is a well known problem in polynomial interpolation. Basically it is the divergence of the interpolant at the boundaries of the computational domain as the number of node points goes to infinity. This happens if the approximated function has singularities within a certain domain in the complex plane called the Runge zone or Runge region. These regions depend on the particular basis and computational grid. In [80], Runge regions very similar to those for polynomials are demonstrated for RBF interpolation in 1D, showing that the RBF method is also vulnerable to the Runge phenomenon. As RBF interpolation with infinitely smooth basis functions approaches polynomial interpolation in the flat limit this is perhaps not surprising, and the Runge phenomenon explains why the error after decreasing with decreasing shape parameter values, at some point starts to increase again and level out in the flat limit.

For polynomial interpolation the usual way to mitigate the Runge phenomenon is by using Chebyshev points instead of equidistant points. In [81] the authors show that, for RBF interpolation in 1D with the GA RBF, a distribution approaching the Chebyshev points is suitable for small shape parameter values while an equidistant distribution works better for larger shape parameter values. In [80] the author uses a family of node distributions depending on a parameter varying between zero and one. The distribution approaches the Chebyshev points as the parameter approaches zero and as the parameter approaches one the distribution becomes equidistant. The parameter is then optimized in order to minimize the so called Lebesgue constant, which is a constant influencing the interpolation accuracy. In [2], three different strategies for 1D GA RBF interpolation are suggested. One strategy is to vary the shape parameter as $\mathcal{O}(N^{3/4})$. The results for this shape parameter choice for the 1D Helmholtz equation can be seen in Figure 6.3. The second suggested strategy is a three layer method in which the boundary layers and the centre layer are treated differently and the third strategy is to use a GA RBF extension. In this extension strategy the data on the computational interval is approximated using a basis of GA cardinal functions with centres placed evenly on a slightly extended interval. Another extension strategy based on Fourier extensions and frames is presented in [79]. A Fourier extension method is based on making the period of the Fourier modes larger than the computational domain and then approximating the function to be interpolated by a Fourier series with this larger period. Frames are bases that have been augmented with redundant elements which results in flexibility and robustness. The author shows that Fourier series are a special case of RBF expansions in the flat limit when the RBF centres are placed around the unit circle. This is then used for the construction of an RBF extension method where the centres are placed around the unit circle but with the collocation data given only at a portion of the circle. The author also suggests choosing the shape parameter based on the rank of the RBF collocation matrix. The method is shown to be both stable and accurate with a superalgebraic convergence even for nodes scattered on the domain, i.e., without node clustering near the boundaries. This is in some sense the best result one can hope for since it is proved in [83] that no method can be simultaneously stable and exponentially convergent on equispaced nodes.

In [25] a few different strategies for decreasing the errors in the boundary regions are suggested. The first one is adding low order polynomial terms and corresponding constraints typically stating that the sum of the RBF coefficients multiplied by the different polynomial terms equals zero which leads to a minimization of the far field values of the RBF approximant. Another strategy is to simply cluster nodes near the boundaries. A third method is to move the outermost or the two outermost RBF centre layers outside the domain. These versions of the same technique are referred to as the Not-a-Knot and the Super Not-a-Knot method respectively. The authors state that the Super Not-a-Knot strategy is the most successful of their suggested methods.

Local refinement can also lead to a Runge phenomenon in the areas where the nodes are sparser and the solution thus less pinned down. This can be controlled by letting the shape parameter vary with the node density with higher shape parameter values in areas with a higher node density. A suggested strategy is to choose a shape parameter and scale it with the inverse distance to the nearest node point. This essentially means that if the node space is stretched to make the nodes equidistant, all basis functions would have the same shape [36]. This strategy is used in [24], where rotational transport equations are solved on a sphere modelling moving vortex roll-up in atmospheric dynamics, e.g., hurricanes. A common near-uniform node distribution on a sphere is the minimum energy (ME) distribution. As the name implies ME nodes minimize the potential energy for electrostatic repulsion of point charges scattered on the surface of the sphere. The authors of [24] describe a local node refinement scheme simulating electrostatic repulsion on the surface of the sphere to a low order, keeping it computationally cheap. By assigning the same charge to all the nodes the result will be an approximate ME set. Instead different charges are assigned to different nodes, based on the angular wind velocity so that nodes in areas where more resolution is needed are assigned lower charges resulting in a higher node density. Nodes are allowed to move until force equilibrium is reached with respect to a given tolerance and this results in a node distribution that both reflects the physics of the PDE and that varies smoothly over the computational domain.

7.2.2 Adaptive node placement

Here we briefly describe some strategies for adaptive node placement, of which almost all fall into the category of greedy methods, i.e., basically node methods that search for the maximum of some quantity chosen to describe the distribution quality in some sense and adding nodes based on this.

In [14] two strategies for choosing near-optimal data independent node distributions are described. Convergence proofs are provided and the authors also show that good interpolation points are always uniformly distributed in a certain sense. The first strategy is based on the so called power function which is a function that depends on the specific RBF and the node set, and that limits the data independent part of the approximation error. The idea is to start with an arbitrary node point in the computational domain, Ω, and then evaluate the power function of the previous node set over some very large point set $\mathcal{X} \in \Omega$. The point where the power function attains its maximum is then added to the node set and the procedure is repeated until the power function value falls under some given tolerance level. The second strategy is geometric and independent of the RBF. For bounded domains in $\mathbb{R}^d$ this method constructs asymptotically uniformly distributed node sets that cover the domain in an asymptotically optimal way. Here the point that has the largest distance to its nearest neighbour in the previous node set is added at each step. According to the authors, practical examples show that the power function based strategy tends to fill the currently largest hole in the node distribution by placing a node close to the centre of this hole. The geometric strategy instead tends to find node sets for which the largest hole in the node distribution is similar in size to the minimum separation distance between the node points.

In [64] the authors prove non-singularity of the non-symmetric collocation method provided that the functions resulting from applying the PDE operators to the basis functions are continuous and the trial centres are properly chosen. The test and trial centres are hence separated. A data-dependent greedy method is described, where a new test and trial centre pair is added at each iteration. The selection of the test centres is based on the maximum residual value over a large set of points and the trial centre points are chosen based on the determinant of the current system matrix to maintain reasonable conditioning. The inverse of the system matrix is built up gradually which decreases the cost of the method to $\mathcal{O}(K^3 + K^2M + K^2N)$ operations where M is the number of test centre candidates, N is the number of trial centre candidates and K is the number of iterations and also the size of the final system, which is typically small compared with M and N. The final step of the method is to solve the collocation problem using the last system matrix, which is square. This algorithm prefers centres outside the computational domain for small shape parameter values. In [65] another adaptive greedy algorithm is constructed from theoretical results for convergence by combining the method just described with linear optimization. This method uses the overdetermined system consisting of the selected trial centres and all the test functionals thus minimizing the maximum error in many points. In [66] an improved version of the algorithm in [64] is presented, where the best test and trial centres are selected based on the maximum primal and dual residual corresponding to the minimization problem associated with the underdetermined system. A fast block-greedy algorithm for quasi-optimal meshless trial subspace selection is introduced in [63]. This method is also based on the primal/dual residual criterion but it adds several test and trial centres at once decreasing the cost to at most $\mathcal{O}(NK^2)$, where N is the number of trial centre candidates, K is the number of selected trial centres and the number of test centre candidates is smaller than or equal to the number of trial centre candidates.

In [16] the authors describe a method they call residual subsampling. Here nodes are added or removed adaptively based on the value of the interpolation error or the PDE residual between the current centre points. The shape parameter value is here chosen in such a way that the product of the shape parameter value and the local node spacing is kept constant, i.e., when the node spacing is halved the shape parameter value is doubled. The authors also note that a smaller starting shape parameter value leads to refinement less obviously connected to the data, while a larger starting value leads to nodes clustering in a more intuitive way.

7.2.3 Examples of node distributions from our research

For the Black–Scholes equation in Paper I we use node distributions where the nodes are clustered in the most interesting region, i.e., around the strike price. Example distributions in 1D and 2D can be seen in Figure 7.4.

In one dimension, the node points are placed in the following way. If $N = 3p + 2$, for some integer p, we distribute $p + 1$ points uniformly in each of the intervals $[0, \bar{K} - \delta]$ and $[\bar{K} + \delta, 2\bar{K}]$. Then we place the remaining p points in the last part of the computational domain. The small distance, δ, from $\bar{K}$ is chosen as $\delta = 1/(N - 1)$ and the symmetric placement around $\bar{K}$ is motivated by numerical experiments showing that errors are reduced by this choice. In two dimensions a similar distribution is chosen in the diagonal direction.

Figure 7.5 shows a comparison between the errors using a uniform and

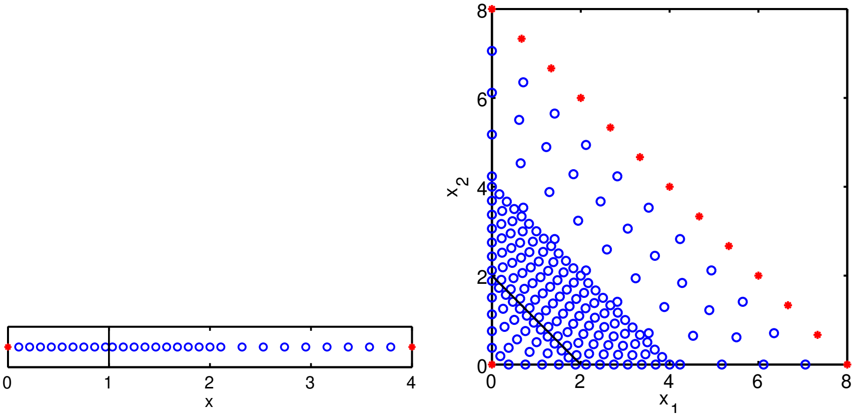


Figure 7.4: Non-uniform node distributions for the Black–Scholes equation in 1D (left) and 2D (right).

non-uniform distribution in one dimension. Figure 7.6 also shows an example of the error for the two distributions. A comparison of the errors for the two types of distributions in two dimensions is shown in Figure 7.7. As can be seen from the results, the errors are smaller for the non-uniform distributions. It should be noted, however, that the non-uniform distributions also worsen the condition numbers of the system matrices.

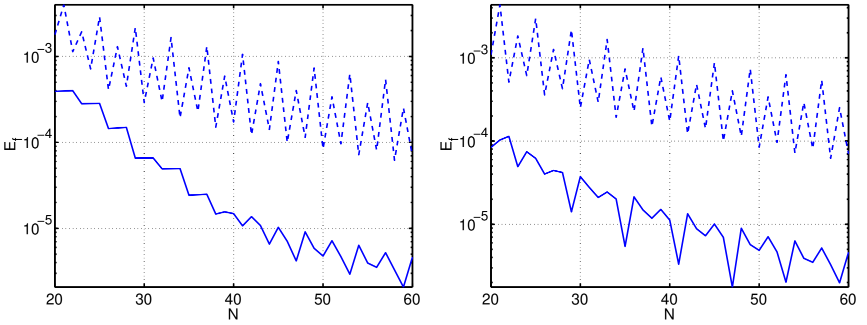


Figure 7.5: Financial error norms for uniform (dashed) and non-uniform (solid) node distributions for the Black–Scholes equation in 1D with $\varepsilon = 4$ (left) and $\varepsilon = 1 + N/20$ (right).

For the Helmholtz problem the most interesting node distribution is for the problem with curved boundaries, see Figure 7.8. Here we use quasi-uniform nodes that are constructed from the input parameters n_1 and n_2, that specify the number of nodes along the left boundary and the number of nodes in the horizontal direction, respectively. We define the step sizes

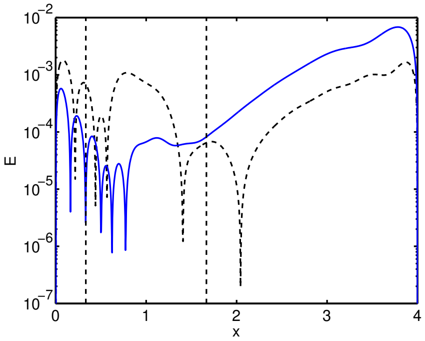


Figure 7.6: The absolute value of the error $E(x)$ for $N = 20$ and $\varepsilon = 2$ for uniform (dashed) and non-uniform (solid) node distributions for the Black–Scholes in 1D.

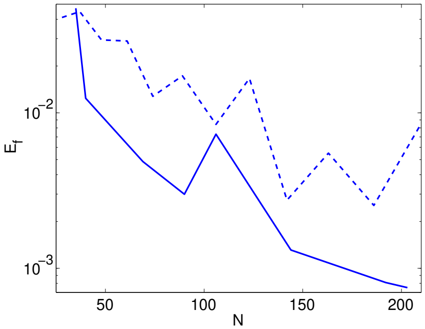


Figure 7.7: Financial error norms for uniform (dashed) and non-uniform (solid) distributions for the Black–Scholes equation in 2D with $\varepsilon = 1$.

$h_1 = L_1/(n_1 - 1)$ and $h_2 = L_2/(n_2 - 1)$, where L_1 and L_2 are the lengths of the domain in each dimension. Based on these step sizes, the nodes are then placed uniformly along vertical lines with as similar node distance as possible. The nodes at the top and bottom boundaries are placed uniformly with respect to the arc length. Finally, we add a random perturbation to each node in order to avoid too regular node patterns, which could cause non-unisolvency and high conditioning.

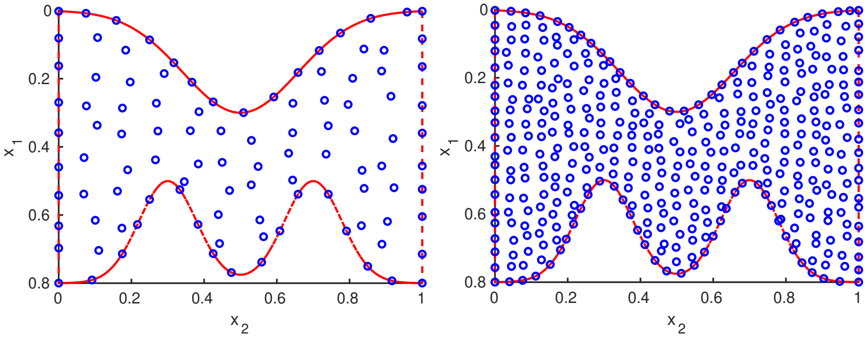


Figure 7.8: Examples of node distributions for the Helmholtz problem.

Chapter 8

Sammanfattning på svenska

Numeriska metoder för lösning av partiella differentialekvationer (PDE:er) kräver ofta ett beräkningsnät, det vill säga en fördelning av beräkningsnoder med någon form av mönster eller koppling mellan de olika nodpunkterna. Finita differensmetoder och pseudospektrala metoder till exempel, kräver mycket regelbundna nodmönster vilket gör dessa metoder mindre flexibla med avseende på beräkningsområdets geometri. Finita elementmetoder är istället geometriskt flexibla, men nätgenereringen för dessa metoder är ofta beräkningstung. Radiella basfunktionsmetoder (RBF-metoder) är, till skillnad från de tidigare uppräknade metoderna, nätfria. Den enda geometriska egenskap de använder sig av är parvisa avstånd mellan olika nodpunkter. Dessa metoder är därför mycket flexibla och kan användas för beräkningar på geometriskt komplicerade områden och de är dessutom lätta att implementera även för problem i högre dimensioner. Grundidén är att man konstruerar en approximant som en linjärkombination av translaterade radiella basfunktioner (RBF:er), det vill säga funktioner vars värde endast beror av avståndet till funktionens centrumpunkt. Därefter utför man så kallad kollokation, det vill säga man sätter in approximanten i PDE:n och med hjälp av givna data i vissa nodpunkter bygger man sedan upp ett linjärt ekvationssystem vars lösning utgörs av approximantens koefficienter. De RBF:er som är oändligt deriverbara brukar resultera i exponentiell konvergens för data genererade av en tillräckligt glatt funktion. Detta kan jämföras med finita elementmetoder och finita differensmetoder som bara kan ge algebraisk noggrannhet. De oändligt deriverbara RBF:erna har också en formparameter som bestämmer hur platta eller spetsiga de är och denna formparameter kan användas för att förbättra metodens noggrannhet.

I denna avhandling fokuserar vi på så kallade globala kollokationsmetoder med RBF:er, det vill säga metoder där man konstruerar approximanten över hela beräkningsområdet direkt, till skillnad från lokaliserade metoder där approximanten byggs upp genom flera lokala approximationer. En nackdel med de globala metoderna är att de resulterar i fyllda matriser som dessutom tenderar att vara illa-konditionerade i det formparameter-intervall som annars kanske hade varit optimalt. Trenden går därför mot en ökad användning av överbestämda system och minsta-kvadrat-approximationer eftersom detta förbättrar såväl stabilitet som noggrannhet och man går också alltmer över till lokaliserade metoder, vilka ger glesa matriser men fortsatt hög noggrannhet. Globala kollokationsmetoder utgör dock, tillsammans med interpolationsmetoder med RBF:er, grunden även för de lokaliserade metoderna. Det är därför fortfarande viktigt att studera och förstå beteendet hos, samt olika praktiska aspekter av, globala kollokationsmetoder. I denna avhandling presenteras en översikt av global RBF-kollokation, med fokus på olika kollokationsvarianter och metodegenskaper. Vi berör till exempel fel- och konvergensbeteende samt approximationsbeteende för små formparametervärden, liksom olika praktiska aspekter som hur man bör välja nodfördelning och formparametervärden. Våra egna forskningsresultat illustrerar olika egenskaper hos global RBF-kollokation med hjälp av Helmholtz ekvation samt Black–Scholes ekvation för prissättning av europeiska korgköpoptioner.

För Black–Scholes ekvation föreslår vi problemanpassade nodfördelningar i 1D och 2D och visar att dessa nodfördelningar resulterar i högre noggrannhet än motsvarande likformigt fördelade nodmängder. Vi föreslår också en formel för formparametervalet för detta specifika problem i 1D. Denna formel resulterar i detta specifika fall i en lösningsnoggrannhet som ligger kring den högsta nåbara utanför det illa-konditionerade formparameter-intervallet. Vi visar att metoden uppnår den förväntade konvergenshastigheten i rum och tid och vi gör en effektivitetsjämförelse mellan global RBF-kollokation och en adaptiv finit differensmetod. Denna jämförelse visar att den globala RBF-metoden är snabbare i både 1D och 2D. Metoderna kräver också ungefär samma minnesutrymme i 2D. Detta visar alltså sammantaget att den globala RBF-kollokationsmetoden kan vara mer effektiv än finita differensmetoder, trots att den resulterar i fyllda matriser.

För Helmholtz ekvation beskriver vi både icke-symmetrisk och symmetrisk kollokation. En nackdel med den icke-symmetriska metoden är att den inte garanterar en icke-singular systemmatris, men då den är lättare att implementera och kräver lägre deriverbarhet hos basfunktionerna än den symmetriska metoden, föredras den ändå ofta. I praktiken är matrisen i de flesta fall icke-singulär. Situationen är dock lite speciell för de PDE:er som har en parameter, som till exempel vågtalet i fallet Helmholtz ekvation. Vi visar för det endimensionella problemet att det för vilken given nodfördelning som helst (med distinkta noder) existerar vågtal som resulterar i en singulär matris. Vi visar dock även numeriskt att systemmatrisen för det endimensionella problemet är icke-singular för väl upplösta problem med reella vågtal. Vi studerar också beteendet hos RBF-lösningen till Helmholtz ekvation i gränsen där formparametern går mot noll, det vill säga där basfunktionerna går mot att bli helt platta. RBF-metoden beter sig här som polynominterpolation och det specifika beteendet är nära knutet till frågan om entydig lösbarhet för polynominterpolationen, samt för PDE-problemet applicerat på polynom, på den givna nodmängden. Vi presenterar ett teorem som beskriver de olika gränsfallen samt respektive villkor och vi ger också exempel på nodmängder som motsvarar de olika villkoren för ett tvådimensionellt Helmholtz-problem. Vi studerar också konvergensbeteendet hos felet som funktion av antalet nodpunkter såväl för mindre som för större formparametervärden. Vi härleder teoretiska feluppskattningar och visar att de stämmer väl överens med numeriska resultat. Olika formler för formparametern har föreslagits i litteraturen och vi visar numeriskt vilken effekt olika val har på konvergensen. Vi tittar också teoretiskt på konvergensen som funktion av formparametern.

Bibliography

[1] I. Babuška and J. M. Melenk, *The partition of unity method*, Internat. J. Numer. Methods Engrg., 40 (1997), pp. 727–758.

[2] J. P. Boyd, *Six strategies for defeating the Runge phenomenon in Gaussian radial basis functions on a finite interval*, Comput. Math. Appl., 60 (2010), pp. 3108–3122.

[3] M. Bozzini, L. Lenarduzzi, M. Rossini, and R. Schaback, *Interpolation by basis functions of different scales and shapes*, Calcolo, 41 (2004), pp. 77–87.

[4] M. Buhmann and N. Dyn, *Spectral convergence of multiquadric interpolation*, Proc. Edinburgh Math. Soc. (2), 36 (1993), pp. 319–333.

[5] M. D. Buhmann, *Radial basis functions: theory and implementations*, vol. 12 of Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 2003.

[6] M. D. Buhmann and S. Dinew, *Limits of radial basis function interpolants*, Commun. Pure Appl. Anal., 6 (2007), pp. 569–585.

[7] M. D. Buhmann, S. Dinew, and E. Larsson, *A note on radial basis function interpolant limits*, IMA J. Numer. Anal., 30 (2010), pp. 543–554.

[8] R. Cavoretto, G. E. Fasshauer, and M. McCourt, *An introduction to the Hilbert-Schmidt SVD using iterated Brownian bridge kernels*, Numer. Algorithms, 68 (2015), pp. 393–422.

[9] A. H.-D. Cheng, M. A. Golberg, E. J. Kansa, and G. Zammito, *Exponential convergence and h-c multiquadric collocation method for partial differential equations*, Numer. Methods Partial Differential Equations, 19 (2003), pp. 571–594.

[10] S. De Marchi, A. Martínez, E. Perracchione, and M. Rossini, *RBF-based partition of unity methods for elliptic PDEs: adaptivity and stability issues via variably scaled kernels*, J. Sci. Comput., 79 (2019), pp. 321–344.

[11] S. De Marchi and G. Santin, *A new stable basis for radial basis function interpolation*, J. Comput. Appl. Math., 253 (2013), pp. 1–13.

[12] ——, *Fast computation of orthonormal basis for RBF spaces through Krylov space methods*, BIT, 55 (2015), pp. 949–966.

[13] S. De Marchi and R. Schaback, *Stability of kernel-based interpolation*, Adv. Comput. Math., 32 (2010), pp. 155–161.

[14] S. De Marchi, R. Schaback, and H. Wendland, *Near-optimal data-independent point locations for radial basis function interpolation*, Adv. Comput. Math., 23 (2005), pp. 317–330.

[15] T. A. Driscoll and B. Fornberg, *Interpolation in the limit of increasingly flat radial basis functions*, Comput. Math. Appl., 43 (2002), pp. 413–422.

[16] T. A. Driscoll and A. R. H. Heryudono, *Adaptive residual subsampling methods for radial basis function interpolation and collocation problems*, Comput. Math. Appl., 53 (2007), pp. 927–939.

[17] G. E. Fasshauer, *Solving partial differential equations by collocation with radial basis functions*, in Surface Fitting and Multiresolution Methods, Vanderbilt Univ. Press, Nashville, TN, 1997, pp. 131–138. (Edited by L. Le Méhauté, C. Rabut and L.L. Schumaker.).

[18] ——, *Meshfree approximation methods with MATLAB*, vol. 6 of Interdisciplinary Mathematical Sciences, World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2007.

[19] G. E. Fasshauer and M. J. McCourt, *Stable evaluation of Gaussian radial basis function interpolants*, SIAM J. Sci. Comput., 34 (2012), pp. A737–A762.

[20] G. E. Fasshauer and J. G. Zhang, *On choosing "optimal" shape parameters for RBF approximation*, Numer. Algorithms, 45 (2007), pp. 345–368.

[21] A. I. Fedoseyev, M. J. Friedman, and E. J. Kansa, *Improved multiquadric method for elliptic partial differential equations via PDE collocation on the boundary*, Comput. Math. Appl., 43 (2002), pp. 439–455.

[22] A. J. M. Ferreira, C. M. C. Roque, R. M. N. Jorge, G. Fasshauer, and R. Batra, *Analysis of functionally graded plates by a robust meshless method*, J. Mech. Adv. Mater. Struct., 14 (2007), pp. 577–587.

[23] N. Flyer and B. Fornberg, *Radial basis functions: developments and applications to planetary scale flows*, Comput. & Fluids, 46 (2011), pp. 23–32.

[24] N. Flyer and E. Lehto, *Rotational transport on a sphere: local node refinement with radial basis functions*, J. Comput. Phys., 229 (2010), pp. 1954–1969.

[25] B. Fornberg, T. A. Driscoll, G. Wright, and R. Charles, *Observations on the behavior of radial basis function approximations near boundaries*, Comput. Math. Appl., 43 (2002), pp. 473–490.

[26] B. Fornberg and N. Flyer, *A primer on radial basis functions with applications to the geosciences*, vol. 87 of CBMS-NSF Regional Conference Series in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2015.

[27] B. Fornberg, N. Flyer, and J. M. Russell, *Comparisons between pseudospectral and radial basis function derivative approximations*, IMA J. Numer. Anal., 30 (2010), pp. 149–172.

[28] B. Fornberg, E. Larsson, and N. Flyer, *Stable computations with Gaussian radial basis functions*, SIAM J. Sci. Comput., 33 (2011), pp. 869–892.

[29] B. Fornberg, E. Larsson, and G. Wright, *A new class of oscillatory radial basis functions*, Comput. Math. Appl., 51 (2006), pp. 1209–1222.

[30] B. Fornberg and E. Lehto, *Stabilization of RBF-generated finite difference methods for convective PDEs*, J. Comput. Phys., 230 (2011), pp. 2270–2285.

[31] B. Fornberg, E. Lehto, and C. Powell, *Stable calculation of Gaussian-based RBF-FD stencils*, Comput. Math. Appl., 65 (2013), pp. 627–637.

[32] B. Fornberg and C. Piret, *A stable algorithm for flat radial basis functions on a sphere*, SIAM J. Sci. Comput., 30 (2007/08), pp. 60–80.

[33] ——, *On choosing a radial basis function and a shape parameter when solving a convective PDE on a sphere*, J. Comput. Phys., 227 (2008), pp. 2758–2780.

[34] B. Fornberg and G. Wright, *Stable computation of multiquadric interpolants for all values of the shape parameter*, Comput. Math. Appl., 48 (2004), pp. 853–867.

[35] B. Fornberg, G. Wright, and E. Larsson, *Some observations regarding interpolants in the limit of flat radial basis functions*, Comput. Math. Appl., 47 (2004), pp. 37–55.

[36] B. Fornberg and J. Zuev, *The Runge phenomenon and spatially variable shape parameters in RBF interpolation*, Comput. Math. Appl., 54 (2007), pp. 379–398.

[37] R. Franke, *Locally determined smooth interpolation at irregularly spaced points in several variables*, J. Inst. Math. Appl., 19 (1977), pp. 471–482.

[38] R. Franke, *Scattered data interpolation: tests of some methods*, Math. Comp., 38 (1982), pp. 181–200.

[39] R. Franke, *Lecture notes on global basis function methods for scattered data*, International Symposium on Surface Approximation, University of Milano, Govgano, Italy, 1983.

[40] M. A. Golberg, C. S. Chen, and S. R. Karur, *Improved multiquadric approximation for partial differential equations*, Eng. Anal. with Bound. Elem., 18 (1996), pp. 9–17.

[41] G. H. Golub and J. M. Ortega, *Scientific computing and differential equations: An introduction to numerical methods*, Academic Press, Inc., Boston, MA, 1992.

[42] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving ordinary differential equations. I*, vol. 8 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, second ed., 1993.

[43] R. L. Hardy, *Multiquadric equations of topography and other irregular surfaces*, J. Geophys. Res., 76 (1971), pp. 1905–1915.

[44] ——, *Theory and applications of the multiquadric-biharmonic method. 20 years of discovery 1968–1988*, Comput. Math. Appl., 19 (1990), pp. 163–208.

[45] A. Heryudono and E. Larsson, *FEM-RBF: A geometrically flexible, efficient numerical solution technique for partial differential equations with mixed regularity.* Marie Curie FP7 Technical Report, 2012.

[46] Y. C. Hon and R. Schaback, *On unsymmetric collocation by radial basis functions*, Appl. Math. Comput., 119 (2001), pp. 177–186.

[47] A. Iske, *Optimal distribution of centers for radial basis function methods*, Report TUM M0004, Techn. Univ. München, Fak. f. Math., 2000.

[48] A. Iske, *Multiresolution methods in scattered data modelling*, vol. 37 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, Berlin, 2004.

[49] S. Janson and J. Tysk, *Feynman-Kac formulas for Black-Scholes-type operators*, Bull. London Math. Soc., 38 (2006), pp. 269–282.

[50] E. J. Kansa, *Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics. I. Surface approximations and partial derivative estimates*, Comput. Math. Appl., 19 (1990), pp. 127–145.

[51] ——, *Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics. II. Solutions to parabolic, hyperbolic and elliptic partial differential equations*, Comput. Math. Appl., 19 (1990), pp. 147–161.

[52] J. B. Keller and D. Givoli, *Exact nonreflecting boundary conditions*, J. Comput. Phys., 82 (1989), pp. 172–192.

[53] K. Kormann and E. Larsson, *A Galerkin radial basis function method for the Schrödinger equation*, SIAM J. Sci. Comput., 35 (2013), pp. A2832–A2855.

[54] K. Kormann, C. Lasser, and A. Yurova, *Stable Interpolation with Isotropic and Anisotropic Gaussians Using Hermite Generating Function*, SIAM J. Sci. Comput., 41 (2019), pp. A3839–A3859.

[55] E. Larsson, K. Åhlander, and A. Hall, *Multi-dimensional option pricing using radial basis functions and the generalized Fourier transform*, J. Comput. Appl. Math., 222 (2008), pp. 175–192.

[56] E. Larsson and B. Fornberg, *A numerical study of some radial basis function based solution methods for elliptic PDEs*, Comput. Math. Appl., 46 (2003), pp. 891–902.

[57] ——, *Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions*, Comput. Math. Appl., 49 (2005), pp. 103–130.

[58] E. Larsson, E. Lehto, A. Heryudono, and B. Fornberg, *Stable computation of differentiation matrices and scattered node stencils based on Gaussian radial basis functions*, SIAM J. Sci. Comput., 35 (2013), pp. A2096–A2119.

[59] E. Larsson, V. Shcherbakov, and A. Heryudono, *A least squares radial basis function partition of unity method for solving PDEs*, SIAM J. Sci. Comput., 39 (2017), pp. A2538–A2563.

[60] Y. J. Lee, C. A. Micchelli, and J. Yoon, *On convergence of flat multivariate interpolation by translation kernels with finite smoothness*, Constr. Approx., 40 (2014), pp. 37–60.

[61] ——, *A study on multivariate interpolation by increasingly flat kernel functions*, J. Math. Anal. Appl., 427 (2015), pp. 74–87.

[62] Y. J. Lee, G. J. Yoon, and J. Yoon, *Convergence of increasingly flat radial basis interpolants to polynomial interpolants*, SIAM J. Math. Anal., 39 (2007), pp. 537–553.

[63] L. Ling, *A fast block-greedy algorithm for quasi-optimal meshless trial subspace selection*, SIAM J. Sci. Comput., 38 (2016), pp. A1224–A1250.

[64] L. Ling, R. Opfer, and R. Schaback, *Results on meshless collocation techniques*, Eng. Anal. Boundary Elements, 30 (2006), pp. 247–253.

[65] L. Ling and R. Schaback, *Stable and convergent unsymmetric meshless collocation methods*, SIAM J. Numer. Anal., 46 (2008), pp. 1097–1115.

[66] ——, *An improved subspace selection algorithm for meshless collocation methods*, Internat. J. Numer. Methods Engrg., 80 (2009), pp. 1623–1639.

[67] W. R. Madych, *Error estimates for interpolation by generalized splines*, in Curves and surfaces (Chamonix-Mont-Blanc, 1990), Academic Press, Boston, MA, 1991, pp. 297–306.

[68] ——, *Miscellaneous error bounds for multiquadric and related interpolators*, Comput. Math. Appl., 24 (1992), pp. 121–138.

[69] ——, *An estimate for multivariate interpolation. II*, J. Approx. Theory, 142 (2006), pp. 116–128.

[70] W. R. Madych and S. A. Nelson, *Multivariate interpolation and conditionally positive definite functions*, Approx. Theory Appl., 4 (1988), pp. 77–89.

[71] ——, *Bounds on multivariate polynomials and exponential error estimates for multiquadric interpolation*, J. Approx. Theory, 70 (1992), pp. 94–114.

[72] J. M. Martel and R. B. Platte, *Stability of radial basis function methods for convection problems on the circle and sphere*, J. Sci. Comput., 69 (2016), pp. 487–505.

[73] J. M. Melenk and I. Babuška, *The partition of unity finite element method: basic theory and applications*, Comput. Methods Appl. Mech. Engrg., 139 (1996), pp. 289–314.

[74] C. A. Micchelli, *Interpolation of scattered data: distance matrices and conditionally positive definite functions*, Constr. Approx., 2 (1986), pp. 11–22.

[75] S. Müller and R. Schaback, *A Newton basis for kernel spaces*, J. Approx. Theory, 161 (2009), pp. 645–655.

[76] D. E. Myers, S. De Iaco, D. Posa, and L. De Cesare, *Space-time radial basis functions*, Comput. Math. Appl., 43 (2002), pp. 539–549.

[77] M. Pazouki and R. Schaback, *Bases for kernel-based spaces*, J. Comput. Appl. Math., 236 (2011), pp. 575–588.

[78] J. Persson and L. von Sydow, *Pricing European multi-asset options using a space-time adaptive FD-method*, Comput. Vis. Sci., 10 (2007), pp. 173–183.

[79] C. Piret, *A radial basis function based frames strategy for bypassing the Runge phenomenon*, SIAM J. Sci. Comput., 38 (2016), pp. A2262–A2282.

[80] R. B. Platte, *How fast do radial basis function interpolants of analytic functions converge?*, IMA J. Numer. Anal., 31 (2011), pp. 1578–1597.

[81] R. B. Platte and T. A. Driscoll, *Polynomials and potential theory for Gaussian radial basis function interpolation*, SIAM J. Numer. Anal., 43 (2005), pp. 750–766.

[82] R. B. Platte and T. A. Driscoll, *Eigenvalue stability of radial basis function discretizations for time-dependent problems*, Comput. Math. Appl., 51 (2006), pp. 1251–1268.

[83] R. B. Platte, L. N. Trefethen, and A. B. J. Kuijlaars, *Impossibility of fast stable approximation of analytic functions from equispaced samples*, SIAM Rev., 53 (2011), pp. 308–318.

[84] M. J. D. Powell, *Univariate multiquadric interpolation: some recent results*, in Curves and surfaces (Chamonix-Mont-Blanc, 1990), Academic Press, Boston, MA, 1991, pp. 371–382.

[85] C. Rieger and B. Zwicknagl, *Sampling inequalities for infinitely smooth functions, with applications to interpolation and machine learning*, Adv. Comput. Math., 32 (2010), pp. 103–129.

[86] S. Rippa, *An algorithm for selecting a good value for the parameter c in radial basis function interpolation*, Adv. Comput. Math., 11 (1999), pp. 193–210.

[87] A. Safdari-Vaighani, A. Heryudono, and E. Larsson, *A radial basis function partition of unity collocation method for convection-diffusion equations arising in financial applications*, J. Sci. Comput., 64 (2015), pp. 341–367.

[88] S. Sarra and E. Kansa, *Multiquadric radial basis function approximation methods for the numerical solution of partial differential equations*, Advances in Computational Mechanics, 2 (2009).

[89] R. Schaback, *Error estimates and condition numbers for radial basis function interpolation*, Adv. Comput. Math., 3 (1995), pp. 251–264.

[90] ——, *Multivariate interpolation by polynomials and radial basis functions*, Constr. Approx., 21 (2005), pp. 293–317.

[91] ——, *Convergence of unsymmetric kernel-based meshless collocation methods*, SIAM J. Numer. Anal., 45 (2007), pp. 333–351.

[92] ——, *Limit problems for interpolation by analytic radial basis functions*, J. Comput. Appl. Math., 212 (2008), pp. 127–149.

[93] ——, *All well-posed problems have uniformly stable and convergent discretizations*, Numer. Math., 132 (2016), pp. 597–630.

[94] R. Schaback and H. Wendland, *Kernel techniques: from machine learning to meshless methods*, Acta Numer., 15 (2006), pp. 543–639.

[95] V. Shankar and G. B. Wright, *Mesh-free semi-Lagrangian methods for transport on a sphere using radial basis functions*, J. Comput. Phys., 366 (2018), pp. 170–190.

[96] V. Shcherbakov, *Radial basis function partition of unity operator splitting method for pricing multi-asset American options*, BIT, 56 (2016), pp. 1401–1423.

[97] G. Song, J. Riddle, G. E. Fasshauer, and F. J. Hickernell, *Multivariate interpolation with increasingly flat radial basis functions of finite smoothness*, Adv. Comput. Math., 36 (2012), pp. 485–501.

[98] A. I. Tolstykh, *On using RBF-based differencing formulas for unstructured and mixed structured-unstructured grid calculations*, in Proceedings of the 16th IMACS World Congress on Scientific Computation, Applied Mathematics and Simulation, Lausanne, Switzerland, 2000, 6 pp.

[99] L. N. Trefethen and M. Embree, *Spectra and pseudospectra*, Princeton University Press, Princeton, NJ, 2005.

[100] H. Wendland, *Fast evaluation of radial basis functions: methods based on partition of unity*, in Approximation theory, X (St. Louis, MO, 2001), Innov. Appl. Math., Vanderbilt Univ. Press, Nashville, TN, 2002, pp. 473–483.

[101] ——, *Scattered data approximation*, vol. 17 of Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 2005.

[102] J. Wertz, E. J. Kansa, and L. Ling, *The role of the multiquadric shape parameters in solving elliptic partial differential equations*, Comput. Math. Appl., 51 (2006), pp. 1335–1348.

[103] G. B. Wright and B. Fornberg, *Stable computations with flat radial basis functions using vector-valued rational approximations*, J. Comput. Phys., 331 (2017), pp. 137–156.

[104] Z. M. Wu, *Hermite-Birkhoff interpolation of scattered data by radial basis functions*, Approx. Theory Appl., 8 (1992), pp. 1–10.

[105] Z. M. Wu and R. Schaback, *Local error estimates for radial basis function interpolation of scattered data*, IMA J. Numer. Anal., 13 (1993), pp. 13–27.

[106] F. Yang, L. Yan, and L. Ling, *Doubly stochastic radial basis function methods*, J. Comput. Phys., 363 (2018), pp. 87–97.

Paper I

Available online at www.sciencedirect.com
ScienceDirect

Journal of Computational and Applied Mathematics 222 (2008) 82–93

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.com/locate/cam

Improved radial basis function methods for multi-dimensional option pricing

Ulrika Pettersson[a], Elisabeth Larsson[a,*], Gunnar Marcusson[b], Jonas Persson[c]

[a] *Department of Information Technology, Uppsala University, Box 337, SE-751 05 Uppsala, Sweden*
[b] *Försäkringsmatematik, Box 5148, SE-102 43 Stockholm, Sweden*
[c] *SunGard Front Arena, Box 70351, SE-107 24 Stockholm, Sweden*

Received 29 April 2006; received in revised form 9 November 2006

Abstract

In this paper, we have derived a radial basis function (RBF) based method for the pricing of financial contracts by solving the Black–Scholes partial differential equation. As an example of a financial contract that can be priced with this method we have chosen the multi-dimensional European basket call option. We have shown numerically that our scheme is second-order accurate in time and spectrally accurate in space for constant shape parameter. For other non-optimal choices of shape parameter values, the resulting convergence rate is algebraic. We propose an adapted node point placement that improves the accuracy compared with a uniform distribution. Compared with an adaptive finite difference method, the RBF method is 20–40 times faster in one and two space dimensions and has approximately the same memory requirements.

MSC: 65M70

Keywords: Radial basis function; Black–Scholes equation; Multi-dimensional; Boundary condition

1. Introduction

The financial markets are becoming more and more complex, with trading not only of stocks, but also of numerous types of financial derivatives. The market requires updated information about the values of these derivatives every second of the day. This leads to a huge demand for fast and accurate computer simulations.

In this study we consider the problem of pricing financial contracts on several underlying assets. These contracts are receiving more and more interest as the demand for complex derivatives from the customers and the speed of computers have increased over the years. We have chosen to use a European basket option as an example. This is a rather simple contract but works well as an indicator of the usefulness of our method.

One way of pricing financial contracts is to solve the Black–Scholes equation [1], a partial differential equation (PDE) in which the number of spatial dimensions is determined by the number of underlying assets. When the number

* Corresponding author.
E-mail address: Elisabeth.Larsson@it.uu.se (E. Larsson).

0377-0427/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2007.10.038

of dimensions grows, this becomes computationally very demanding. Thus, it is necessary to use fast and memory efficient algorithms.

Other methods to price high-dimensional contracts are, e.g., Monte Carlo methods, "sparse grids", and finite difference methods. Monte Carlo methods have the advantage of scaling linearly with the number of dimensions, but have the drawback of converging very slowly. There are several ways of speeding-up the convergence, e.g., different variance reduction and quasi-random sequence techniques. A good reference for Monte Carlo algorithms and theory is [2]. Sparse grids is an approximation technique rediscovered in the 1990s. By combining several grids with different step sizes, a small number of grid points can be used to achieve an accurate approximation and keep the memory requirements low. Reference [3] gives an introduction to sparse grids with applications. Finite difference methods are generally well known, but for details about the method we have used here for comparisons, we refer to Section 5.

Here, we consider RBF approximation as a potentially effective approach for solving the multi-dimensional Black–Scholes equation. A typical RBF approximant has the form

$$u(\vec{x}) = \sum_{j=1}^{N} \lambda_j \phi(\varepsilon \|\vec{x} - \vec{x}_j\|),$$

where $\phi(r)$ is the RBF, $\vec{x}_j$, $j = 1, \ldots, N$ are center points, and ε is a shape parameter. A small value of ε leads to flatter RBFs. The shape parameter is an important method parameter, with a significant effect on the accuracy of the method. With infinitely smooth RBFs the method can be spectrally accurate [4,5], meaning that the required number of node points for a certain desired accuracy is potentially very small. Since the method only needs pairwise distances between points, it is meshfree. Therefore, it is easy to use in higher dimensions and it also allows for problem adapted node placement.

Option pricing using RBFs has been explored in one dimension for European and American options by Hon et al. [6,7] and in both one and two dimensions by Fasshauer et al. [8] and Marcozzi et al. [9] with promising results. Hon has also applied a quasi-radial basis function method to option pricing in one dimension [10].

The contribution of this paper is a thorough numerical study of the effects of the method parameters on the accuracy and performance of the method, providing some insights regarding the possibilities and limitations of RBF methods. We look at sample problems in one and two dimensions and we also compare the results of the RBF method with those of an adaptive finite difference scheme [11]. Furthermore, we discuss boundary conditions both from a theoretical and an implementational viewpoint.

The outline of the paper is as follows. In Section 2, we present the sample problems and boundary conditions. Then, in Section 3, we derive the space approximation and time discretization of the problem. Section 4 contains numerical experiments for the RBF method and Section 5 shows the results of the comparison with the adaptive finite difference method. Finally, Section 6 gives some conclusions.

2. The multi-dimensional Black–Scholes problem

2.1. The Black–Scholes equation

The Black–Scholes equation is a time-dependent linear PDE, in its original formulation posed as a final value problem. Here we use a transformed version of the PDE. Time is reversed to make standard texts on time-integration for PDEs applicable, and all variables have been scaled to be dimensionless. The details of the transformation can be found in [11]. The transformed problem reads

$$\begin{cases} \dfrac{\partial}{\partial \hat{t}} P(\hat{t}, \vec{x}) = \mathcal{L} P(\hat{t}, \vec{x}), & \hat{t} \in \mathbb{R}_+,\ \vec{x} \in \mathbb{R}_+^d, \\ P(0, \vec{x}) = \Phi(\vec{x}), & \vec{x} \in \mathbb{R}_+^d, \end{cases} \tag{1}$$

where

$$\mathcal{L} P = 2\bar{r} \sum_{i=1}^{d} x_i \frac{\partial P}{\partial x_i} + \sum_{i,j=1}^{d} [\bar{\sigma}\bar{\sigma}^{\mathrm{T}}]_{ij} x_i x_j \frac{\partial^2 P}{\partial x_i \partial x_j} - 2\bar{r} P, \tag{2}$$

where $P(\hat{t}, \vec{x})$ is the value of the option at time $\hat{t}$ when the underlying assets have the values given by $\vec{x}$. Furthermore, the coefficient $\bar{r}$ is the scaled short interest rate, $\bar{\sigma}$ is the scaled volatility and d denotes the number of underlying assets and thus the number of spatial dimensions of the problem. An example of a contract function for a European basket call option is the average option

$$\Phi(\vec{x}) = \max\left(\frac{1}{d}\sum_{i=1}^{d} x_i - \bar{K}, 0\right), \tag{3}$$

where the scaled strike price in our case is $\bar{K} = 1$. The weights could also be different from $1/d$, but that would just be another scaling of the variables. This type of contract function is considered in [12].

2.2. Boundary conditions for the finite difference method

As mentioned previously, we use the adaptive finite difference method derived in [11] for reference solutions and for comparisons. For a finite difference discretization of the Black–Scholes problem, (numerical) boundary conditions are needed at all parts of the boundary. This implementation employs

$$\frac{\partial^2 P(\vec{x}, \hat{t}\,)}{\partial n^2} = 0, \tag{4}$$

where $\partial/\partial n$ indicates differentiation in the direction normal to the boundary. This is an approximation discussed and used in [12]. It has also been successfully used in [11,13]. There are of course other possible boundary conditions, but this choice has proven to work very well for this problem. Condition (4) is approximated by a second-order discretization of the second derivative and can be considered as a linear extrapolation of the solution up to the boundary. For all interior grid points a second-order discretization of the PDE is used.

2.3. Boundary conditions for the RBF method

Condition (4) does not work well with an RBF approximation method. One reason is that it does not imply linearity in a region near the boundary, since the condition is enforced only at the boundary and the infinitely smooth RBFs that we use are not in themselves linear.

In [14], Janson and Tysk show that the problem we consider here is actually well posed without boundary conditions as long as the growth at infinity is restricted. Therefore, we only use near- and far-field boundary conditions. This means that no boundary conditions are employed at boundaries of the type $\Gamma_i = \{\vec{x} \mid \vec{x} \in \mathbb{R}^d_+, \vec{x} \neq \vec{0}, x_i = 0\}$, $i = 1, \ldots, d$.

The near-field boundary can be seen as the single point $\vec{x} = \vec{0}$, and there we enforce

$$P(\hat{t}, \vec{0}) = 0. \tag{5}$$

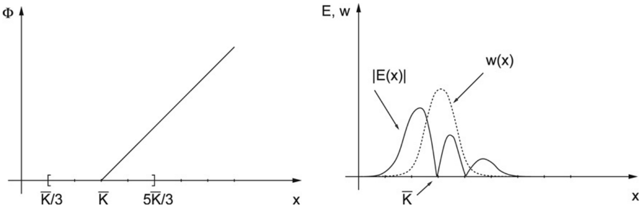
At the far-field boundary, which we have not yet defined, we use the asymptotic solution

$$P(\hat{t}, \vec{x}) \to \frac{1}{d}\sum_{i=1}^{d} x_i - \bar{K}\mathrm{e}^{-2\bar{r}\hat{t}}, \quad \|\vec{x}\| \to \infty. \tag{6}$$

A different approach to boundary conditions for the RBF method was used in [8]. There $(d-1)$-dimensional problems are solved at the parts of the boundary where we do not enforce any boundary conditions at all. Our arguments are (i) errors in the computations in the lower dimensions are transferred to and possibly enlarged in the higher dimensions, (ii) with the need to recursively solve PDEs in all dimensions up to d, it becomes more difficult to implement the algorithm, and (iii) since the PDE at the boundaries collapses into lower-dimensional versions, time-stepping the boundary points along with the rest should automatically provide the correct behavior.

2.4. Computational domain

The problem is defined on $\mathbb{R}^d_+$, but for computational reasons we need to restrict the problem to a finite domain. For the finite difference method the domain is $[0, a_1] \times [0, a_2] \times \cdots \times [0, a_d]$, in order to easily construct the structured grid



(a) The contract function and Ω_i.

(b) The weight function (dashed) and $|E(x)|$ (solid).

Fig. 1. Illustrations of the error norms in one dimension.

that is needed. However, the RBF method is meshfree, which gives us the opportunity to choose the artificial far-field boundary as we like. With the contract function (3), it makes sense to use a boundary surface of the type $\sum_{i=1}^{d} x_i = C$, where the constant C is chosen to bring the surface far enough from the origin for the far-field solution (6) to be an accurate approximation.

2.5. Measuring the error

When measuring the error in the approximate solutions it is important to remember the real-life background of the problems we are solving. Firstly, when we solve the Black–Scholes equation, we want to know the price ($\hat{t} = T$) of an option today with exercise time T years from today. That is, the error function is given by

$$E(\vec{x}) = P(T, \vec{x}) - u(T, \vec{x}). \tag{7}$$

Secondly, in option trading, the region of most interest is when the mean of the stock prices is close to the strike price. Typically, the probability for a stock to default or to be very far from the strike price is small. Based on actual trading data from the Stockholm stock exchange, we define the region of interest Ω_i to be all $\vec{x}$ for which

$$\frac{1}{d}\sum_{i=1}^{d} x_i \in \left[\frac{\bar{K}}{3}, \frac{5\bar{K}}{3}\right]$$

holds, and propose a financial error norm given by

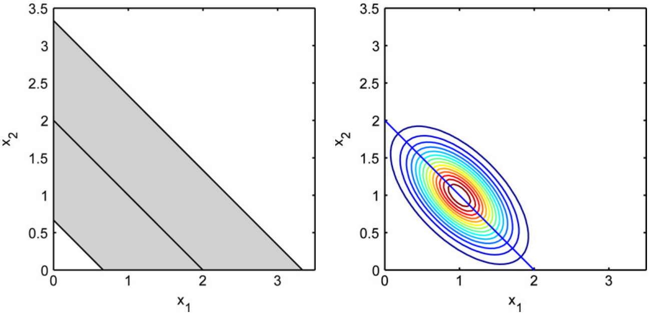
$$E_f = \max_{\vec{x}\in\Omega_i} |E(\vec{x})|. \tag{8}$$

The region of interest Ω_i is depicted in Fig. 1(a) for a one-dimensional problem and in Fig. 2(a) for a two-dimensional problem.

We have also used a weighted integral norm defined as

$$E_w = \int_{\Omega} w(\vec{x})|E(\vec{x})|\mathrm{d}\vec{x}, \tag{9}$$

where Ω is the whole computational domain. The weight function is chosen as a product of d Gaussian functions, centered in the region of interest and with $\int_{\Omega} w(\vec{x})\mathrm{d}\vec{x} = 1$. In one dimension, we use $w(\vec{x}) \propto \exp(-5(x - \bar{K})^2)$ and in two dimensions, we use $w(\vec{x}) \propto \exp(-4(x_1 + x_2 - 2\bar{K})^2)\exp(-(x_1 - x_2)^2)$. The weight functions in one and two dimensions are shown in Figs. 1(b) and 2(b), respectively. The idea can be extended to several dimensions and other contract functions by changing the function $w(\vec{x})$ accordingly. The main reason for using this norm is that it was the output of one of the adaptive finite difference codes that we wanted to compare with. However, it also makes sense to remove the influence of the larger errors at the boundary, where one stock is defaulted, since this case is of limited interest.



(a) The region Ω_i. (b) Contours of $w(\vec{x})$.

Fig. 2. Illustrations concerning the error norms in two dimensions. The line $s_1 + s_2 = 2$ in both the subfigures is where the contract function has a discontinuous first derivative.

3. RBF approximation and time-stepping

We approximate the solution of (1) with a time-dependent linear combination of RBFs centered at the node points $\vec{x}_k, k = 1, \ldots, N$,

$$u(\hat{t}, \vec{x}) = \sum_{k=1}^{N} \lambda_k(\hat{t}\,)\phi(\varepsilon\|\vec{x} - \vec{x}_k\|) = \sum_{k=1}^{N} \lambda_k(\hat{t}\,)\phi_k(\vec{x}), \tag{10}$$

where $\phi(r)$ is the radial basis function, ε is the shape parameter, and $\lambda_k(\hat{t}\,)$ are coefficients to be determined.

Our method of determining these coefficients is collocation at the node points. For interior node points $\vec{x}_k$, $k = 1, \ldots, N_i$ we use Eq. (1) and for node points at the near or far field boundaries, $\vec{x}_k$, $k = N_i + 1, \ldots, N$, we enforce (5) or (6), respectively. Let $\vec{u}_i(\hat{t}\,) = (u(\hat{t}, \vec{x}_1), \ldots, u(\hat{t}, \vec{x}_{N_i}))^{\mathrm{T}}$ and $\vec{u}_b(\hat{t}\,) = (u(\hat{t}, \vec{x}_{N_i+1}), \ldots, u(\hat{t}, \vec{x}_N))^{\mathrm{T}}$. Then from (10)

$$\begin{pmatrix} \vec{u}_i(\hat{t}\,) \\ \vec{u}_b(\hat{t}\,) \end{pmatrix} = \begin{pmatrix} A_{ii} & A_{ib} \\ A_{bi} & A_{bb} \end{pmatrix} \begin{pmatrix} \vec{\lambda}_i(\hat{t}\,) \\ \vec{\lambda}_b(\hat{t}\,) \end{pmatrix}, \tag{11}$$

where the total coefficient matrix A has elements $a_{jk} = \phi(\varepsilon\|\vec{x}_j - \vec{x}_k\|)$ and the indicated block structure is due to the decomposition of interior and boundary node points. Furthermore, A is non-singular for standard choices of RBFs [15], and

$$\begin{aligned} \mathcal{L}\vec{u}_i(\hat{t}\,) &= \begin{pmatrix} B_{ii} & B_{ib} \end{pmatrix} \begin{pmatrix} \vec{\lambda}_i(\hat{t}\,) \\ \vec{\lambda}_b(\hat{t}\,) \end{pmatrix} = \begin{pmatrix} B_{ii} & B_{ib} \end{pmatrix} A^{-1} \begin{pmatrix} \vec{u}_i(\hat{t}\,) \\ \vec{u}_b(\hat{t}\,) \end{pmatrix} \\ &\equiv \begin{pmatrix} C_{ii} & C_{ib} \end{pmatrix} \begin{pmatrix} \vec{u}_i(\hat{t}\,) \\ \vec{u}_b(\hat{t}\,) \end{pmatrix}, \end{aligned} \tag{12}$$

where the matrix elements of B are $b_{jk} = \mathcal{L}\phi(\varepsilon\|\vec{x}_j - \vec{x}_k\|)$, for $j = 1, \ldots, N_i$ and $k = 1, \ldots, N$.

The eigenvalues of C_{ii} determine the stability limits for the time-steps of different time advancing methods. For the problems we consider here, the range of size of the eigenvalues is quite large, but there are no eigenvalues with positive real part. Therefore, we have chosen to use the unconditionally stable BDF2 method [16] for the time evolution of the problem. We use a constant time-step k. Let $\hat{t}^{\,n} = kn$ and let $\vec{u}_i^n \approx \vec{u}_i(\hat{t}^{\,n})$. The time-stepping scheme applied to (1) yields

$$\vec{u}_i^n + \beta_1\vec{u}_i^{n-1} + \beta_2\vec{u}_i^{n-2} = k\beta_0\mathcal{L}\vec{u}_i^n, \tag{13}$$

where $\beta_0 = 1$, $\beta_1 = -1$, and $\beta_2 = 0$ for the first time-step and $\beta_0 = \frac{2}{3}$, $\beta_1 = -\frac{4}{3}$, and $\beta_2 = \frac{1}{3}$ for subsequent steps.

The boundary conditions are enforced at each new time level through

$$\vec{u}_b^n = \vec{g}_b^n, \tag{14}$$

where $\vec{g}_b^n = (g(\hat{t}^{\,n}, \vec{x}_{N_i+1}), \ldots, g(\hat{t}^{\,n}, \vec{x}_N))^{\mathrm{T}}$, and

$$g(\hat{t}, \vec{x}) = \begin{cases} 0, & \vec{x} = \vec{0} \\ d^{-1} \displaystyle\sum_{i=1}^{d} x_i - \bar{K} \mathrm{e}^{-2\bar{r}\hat{t}}, & \|\vec{x}\|_1 = C. \end{cases} \tag{15}$$

Combining (12)–(14) gives the overall scheme for advancing all unknowns one step in time,

$$\begin{pmatrix} I - k\beta_0 C_{ii} & -k\beta_0 C_{ib} \\ 0 & I \end{pmatrix} \begin{pmatrix} \vec{u}_i^n \\ \vec{u}_b^n \end{pmatrix} = \begin{pmatrix} -\beta_1 \vec{u}_i^{n-1} - \beta_2 \vec{u}_i^{n-2} \\ \vec{g}_b^n \end{pmatrix}. \tag{16}$$

The initial condition from (1) in discrete form is

$$\vec{u}_i^0 = \vec{f}_i = (\Phi(\vec{x}_1), \ldots, \Phi(\vec{x}_{N_i}))^{\mathrm{T}}. \tag{17}$$

Due to the change in β_0 between the first and second time-step, we need to factorize the matrix block $I - k\beta_0 C_{ii}$ twice. However, this can be avoided by choosing the time-step in a special way [17].

In, e.g., [6], the authors claim that the time-stepping is the major source of numerical errors. However, we suspect that this is related to how the boundary conditions are implemented. In our scheme (16) the boundary conditions are incorporated in a correct way, and we show in Section 4.4 that we get the expected second-order convergence in time. If instead the boundary unknowns are adjusted separately after each time-step, an error is introduced in the whole domain through the global coupling of the unknowns and time continuity is lost.

4. Numerical experiments

We have used multi-quadric RBFs in all the experiments, i.e., $\phi(r) = \sqrt{1 + r^2}$. The far-field boundary surface was given by all $\vec{x}$ for which $\frac{1}{d} \sum_{i=1}^{d} x_i = 4\bar{K}$. The problem parameters were set to $\bar{r} = 5/9$, corresponding to $r = 0.05$, and $\bar{\sigma} = 1$, corresponding to $\sigma = 0.3$, in one dimension. For the two-dimensional problem we used

$$\bar{\sigma} = \begin{pmatrix} 1 & 1/6 \\ 1/6 & 1 \end{pmatrix}, \quad \text{corresponding to} \quad \sigma = \begin{pmatrix} 0.30 & 0.05 \\ 0.05 & 0.30 \end{pmatrix}.$$

The number of time-steps M is in most cases chosen as the smallest M such that using $M + 1$ steps does not lead to a significant improvement of the accuracy. In cases where we are not looking at performance, M is just chosen large enough not to influence the accuracy. The exercise time used was $T = 0.045$, corresponding to 1 year.

The accuracy of the RBF method naturally depends on the number of node points N. However, the accuracy is also very much influenced by the choice of shape parameter and to a lesser degree by the distribution of the node points. In the following subsections, we first discuss how to make these choices, and then we look at space and time accuracy.

4.1. Node distribution

Since we are concerned with making the error small in the region of interest, we can adapt the node point distribution to reduce the financial error norms, while allowing a larger error in the far-field region.

We have not tried to optimize the node distribution, but we have tried some different approaches and found one that gives a clear improvement compared with a uniform distribution. Examples are shown in Fig. 3. In one dimension, the node points are placed in the following way. If $N = 3p + 2$, for some integer p, we distribute $p + 1$ points uniformly in the intervals $[0, \bar{K} - \delta]$ and $[\bar{K} + \delta, 2\bar{K}]$. Then we place the remaining p points in the last part of the computational domain. The small distance, δ, from $\bar{K}$ is chosen as $\delta = 1/(N - 1)$. The symmetric placement around $\bar{K}$ is motivated by numerical experiments showing that errors are reduced by this choice. In two dimensions a similar distribution is chosen in the diagonal direction. Fig. 4 shows the difference between using a uniform and non-uniform distribution in one dimension. Fig. 5 also shows an example of the error $E(x)$ for the two distributions. A comparison of the errors

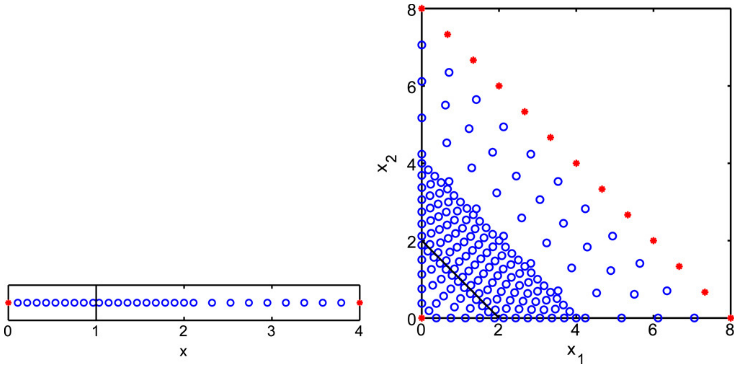


Fig. 3. Non-uniform node distributions in one dimension (left) and two dimensions (right).

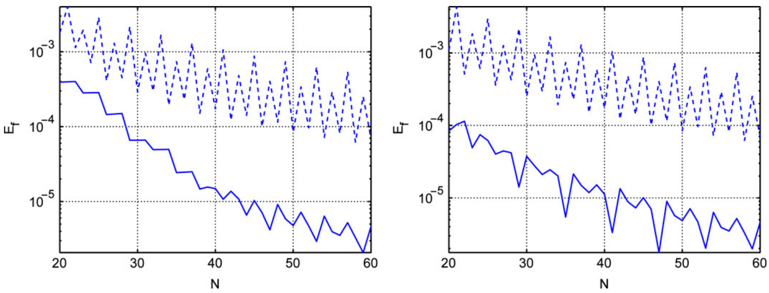


Fig. 4. Financial error norms for uniform (dashed) and non-uniform (solid) node distributions in one dimension with $\varepsilon = 4$ (left) and $\varepsilon = 1 + N/20$ (right).

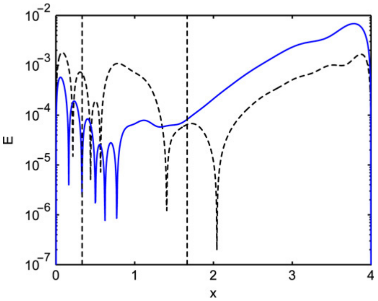


Fig. 5. The absolute value of the error $E(x)$ for $N = 20$ and $\varepsilon = 2$ for uniform (dashed) and non-uniform (solid) node distributions in one dimension.

for the two types of distributions in two dimensions is shown in Fig. 6. To compute these errors we used a reference solution computed by the finite difference method on a very fine grid. It should be noted that the price for the smaller errors with the non-uniform distribution is that the conditioning of the matrix A in (11) becomes worse.

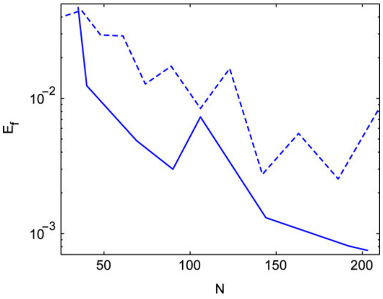


Fig. 6. Financial error norms for uniform (dashed) and non-uniform (solid) distributions in two dimensions with $\varepsilon = 1$.

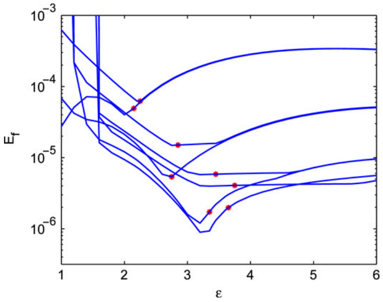


Fig. 7. The financial error norm as a function of ε for $N \in [20, 60]$. The stars show $\varepsilon = 1 + N/20$.

4.2. *Choosing the shape parameter value*

The best choice of shape parameter is problem dependent [18] and there is (currently) no easy way to determine it a priori. Furthermore, the RBF matrices become increasingly ill-conditioned when ε decreases, making it impossible to compute the approximation at small optimal shape parameters using standard methods. However, there are methods to get around this for moderate numbers of node points [19,20].

The best shape parameter value, for N ranging from 20 to 60 in the one-dimensional problem, can be reasonably well approximated by $\varepsilon = 1 + N/20$ for our particular choice of node point distribution. The difference between using a constant ε and the formula above is illustrated in Fig. 4. As can be seen in Fig. 7 the optima are not always well defined and we are very close to the ill-conditioned zone.

It is easy to believe that the formula that works well for small N is also a suitable choice for larger N. However, the asymptotic convergence rate can be very different from the initial behavior [21]. This is illustrated in the left part of Fig. 8, where the error is plotted for a larger range of N. The error is computed with the non-uniform distribution and plotted against the corresponding uniform step size $h = 4\bar{K}/(N - 1)$. The fitted slopes are 1.3 and 4.4 and indicate an algebraic rate of convergence in both regions. The right part of the figure shows that by letting ε grow slower with N, we improve the asymptotic rate of convergence. The slopes are 1.5, 1.9 and 2.4 respectively. The convergence rates could be improved even more by taking smaller ϵ, but the ill-conditioning prevents us from doing this.

4.3. *Accuracy in space*

One of the main advantages of the RBF method is that it can provide spectral accuracy. However, the experiments in the previous section only showed algebraic convergence. The reason is that ε was increased with N. The spectral

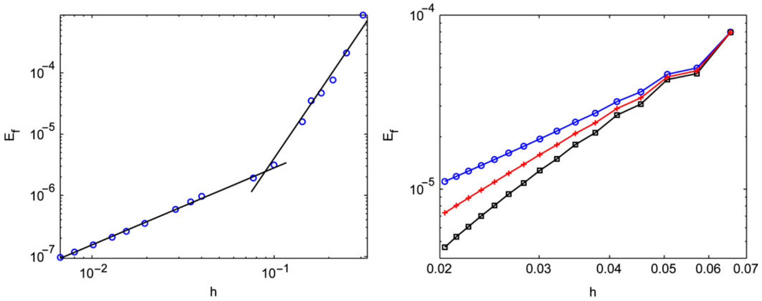


Fig. 8. Left: Financial error norm for the choice $\varepsilon = 1 + N/20$. Right: Financial error norm for $\varepsilon = 1 + N/20$ (○), $\varepsilon = a + bN^{3/4}$ (+) and $\varepsilon = c + dN^{1/2}$ (□), where a, b, c, and d are constants.

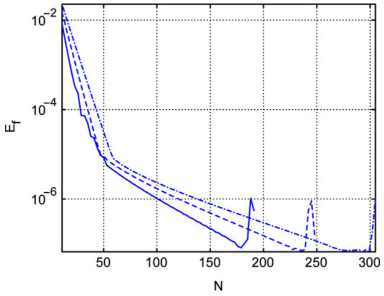


Fig. 9. The financial error norm as a function of N for constant $\varepsilon = 6$ (solid), $\varepsilon = 8$ (dashed), and $\varepsilon = 10$ (dash–dot).

accuracy holds for fixed ε and in Fig. 9 the spectral convergence rate can be observed. That is,

$$E_f = C\exp(-\alpha N).$$

Also here the asymptotic rate of convergence is different from that for small N. The value of α is approximately 0.2 for all three values in the small N region and in the large N part, $\alpha \approx 0.032$, 0.026, and 0.021. As can be seen, the spectral rate is higher for smaller ε. This was also observed in [21].

4.4. Accuracy in time

The accuracy in time, for the one-dimensional problem, was studied by fixing the spatial part of the approximation to $N = 98$ RBFs with shape parameter $\epsilon = 1 + 98/20 = 5.9$, and then varying the number of time-steps from $M = 2$ to $M = 10^4$. The results are displayed in Fig. 10.

The different curves correspond to different ways of measuring. For all errors we see that the expected order of accuracy 2 is realized. However, measuring the maximum error over the whole interval (+) includes large errors at the boundary that increasing M cannot remove. We can draw this conclusion since when measuring the maximum error in the interior of the domain, with the financial norm (solid line), it is possible to get smaller values of the error without increasing N or changing any other parameter. Using the weighted integral norm (○) it is possible to reduce the error even further. Studying the error locally at the strike price shows that we can get errors as low as 10^{-8} with 98 basis functions. For the last three ways of measuring the error it is very clear when the error from the space approximation takes over and starts to dominate. When this happens and on what level depends on where and how the error is measured.

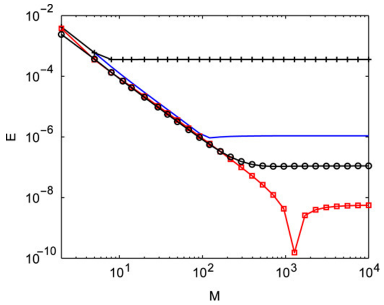


Fig. 10. The error E as a function of the number of time-steps M. Maximum error over the whole region (+), financial norm (solid), weighted integral norm (○) and error at $x = \bar{K}$ (□).

5. Efficiency of the RBF method compared with the finite difference method

In order to investigate the efficiency of the RBF method, we measured the execution time for the one-dimensional problem and the execution time and memory requirements for the two-dimensional problem, and compared the results with those of the finite difference method presented in [11]. Both implementations are in MATLAB and none of the codes are optimized. A brief description of the finite difference method and the results of the comparisons are given below.

5.1. Adaptive finite differences

The generalized Black–Scholes equation (1) can be solved by approximating the derivatives in space and time by finite difference, see e.g. [12]. In [11] centered second-order finite differences on a structured but not equidistant grid are used in space, and the second-order implicit, unconditionally stable BDF2 scheme [16] is used for the time discretization.

The adaptive algorithm in space automatically adjusts the discretization to achieve a predefined truncation error. This allows the user to choose the error level instead of the number of grid points as is standard in non-adaptive finite difference implementations. The adaptive method can alternatively be used to minimize the memory usage by restricting the number of grid points used in each dimension.

Time adaptivity is implemented through a variable step size BDF2 version combined with an explicit multi-step method used for estimating the local truncation error at each time-step. The time-step is then chosen so that the error is controlled.

The adaptive method has been successfully used for European basket options in [11] where the local truncation error is controlled and in [13] where a functional of the global error is estimated and controlled using a similar technique.

Since the time-stepping algorithm is implicit and the approximation of space derivatives is local, the solution of large, but very sparse, systems of equations is necessary. For this purpose, the iterative restarted GMRES method [22] has been used, together with a preconditioner (incomplete LU factorization) to speed-up the computations.

5.2. Results of the efficiency tests

The results of the tests can be seen in Figs. 11 and 12. In Fig. 11 the effect of the shape parameter choice can again be observed. The formula $\varepsilon = 1 + N/20$ was used, and the two different convergence rates are reflected by the time consumption of the RBF method. With a lower convergence rate, N must increase more to get to a desired tolerance, and hence the computational time grows faster. This illustrates that another choice of shape parameter values should preferably be made for larger N, i.e., when extremely high accuracy is desired.

For the two-dimensional experiments the choice of ε was not made in a rigorous way. The shape parameter value was optimized locally in a small interval, typically around $\varepsilon = [0.5, 3.5]$, close to the ill-conditioned zone for each

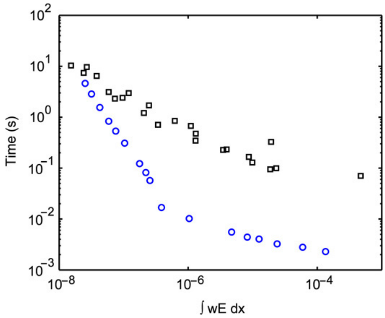


Fig. 11. Time efficiency comparison between the RBF method (circles) and the second-order accurate adaptive finite difference method (squares) for the one-dimensional problem.

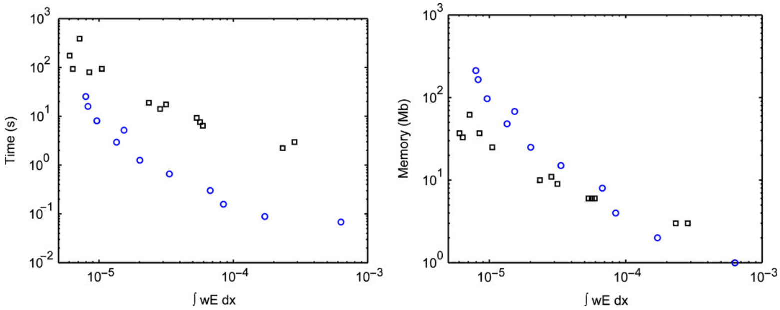


Fig. 12. Time and memory efficiency comparison between the RBF method (circles) and the second-order accurate adaptive finite difference method (squares) for the two-dimensional problem.

experiment. Again the results in Fig. 12 suggest that these choices of shape parameters are not optimal for larger N, corresponding with the left part of the figure.

Measured on the right-hand parts of the two figures, i.e., where the choices of ε are relatively good, the RBF method is approximately 20–40 times faster than the finite difference method. Although the speed-up is lower in two dimensions than in one dimension, it is still good enough in two dimensions to suggest that the RBF method can be more efficient than the finite difference method for (even) higher-dimensional problems. The memory requirements are rather similar for the two methods in two dimensions, which is a positive result considering that the RBF method works with dense matrices, whereas the finite difference method uses sparse matrices. There are also possibilities to improve the performance of the RBF methods further both with respect to memory and time usage [17].

6. Conclusions

In this work we have derived a streamlined RBF method for option pricing in several dimensions, including boundary conditions. We have shown that it is second order in time (due to the second-order time-stepping scheme) and spectrally accurate in space. We have also shown that it can be difficult to take full advantage of the spectral property due to the ill-conditioning of the RBF matrices for small shape parameter values.

Furthermore, we have shown how an adapted placement of the node points, instead of a standard uniform choice, can increase the accuracy by up to an order of magnitude. We believe that even better node distribution strategies can be found for problems in two or more dimensions. By exploiting the meshfree nature of RBF approximation, we can also reduce the size of the computational domain by $d!$ in d dimensions.

We have investigated how the convergence rate in space is affected by the choice of shape parameter and found that if the shape parameter is held constant the convergence rate is spectral. However, if the shape parameter is *increased* according to some formula $\varepsilon \propto N^q$, $q > 0$, the resulting convergence rate becomes algebraic and grows worse with increasing q. Unless special algorithms for small ε are employed [19], a general recommendation must be to use the smallest ε for which stable computation is possible.

The new RBF method has been compared with an existing second-order adaptive finite difference method and the experiments show that the RBF method is 20–40 times faster than the finite difference method in the low to intermediate accuracy range. The slower convergence rate in the region of high accuracy is an interesting phenomenon that we would like to study further. However, for this application, very high accuracy is not of practical interest, since the model itself is not that accurate. The memory requirements of the two methods are comparable for the problems considered here.

We conclude that overall, the RBF method performs well. There are further improvements to be made and we expect that RBF methods for option pricing will be competitive in higher dimensions also.

Acknowledgements

The first and fourth authors' parts of work were funded by FMB, the Swedish Graduate School in Mathematics and Computing. The second author's part of work was supported by a grant from the Swedish Research Council.

References

[1] F. Black, M. Scholes, The pricing of options and corporate liabilities, J. Polit. Econ. 81 (3) (1973) 637–654.

[2] P. Glasserman, Monte Carlo methods in financial engineering, in: Stochastic Modelling and Applied Probability, in: Applications of Mathematics, vol. 53, Springer-Verlag, New York, 2004.

[3] H.-J. Bungartz, M. Griebel, Sparse grids, Acta Numer. 13 (2004) 147–269.

[4] W.R. Madych, Miscellaneous error bounds for multiquadric and related interpolators, Comput. Math. Appl. 24 (12) (1992) 121–138.

[5] M. Buhmann, N. Dyn, Spectral convergence of multiquadric interpolation, Proc. Edinb. Math. Soc. 36 (2) (1993) 319–333.

[6] Y.-C. Hon, X.-Z. Mao, A radial basis function method for solving options pricing models, J. Financ. Eng. 8 (1999) 31–49.

[7] Z. Wu, Y.-C. Hon, Convergence error estimate in solving free boundary diffusion problem by radial basis functions method, Engrg. Anal. Bound. Elem. 27 (2003) 73–79.

[8] G.E. Fasshauer, A.Q.M. Khaliq, D.A. Voss, Using meshfree approximation for multi-asset American option problems, J. Chin. Inst. Eng. 27 (2004) 563–571.

[9] M.D. Marcozzi, S. Choi, C.S. Chen, On the use of boundary conditions for variational formulations arising in financial mathematics, Appl. Math. Comput. 124 (2001) 197–214.

[10] Y.C. Hon, A quasi-radial basis functions method for American options pricing, Comput. Math. Appl. 43 (3) (2002) 513–524.

[11] J. Persson, L. von Sydow, Pricing European multi-asset options using a space-time adaptive FD-method, Comput. Vis. Sci. 10 (2007) 173–183, Published electronically in July.

[12] D. Tavella, C. Randall, Pricing Financial Instruments: The Finite Difference Method, John Wiley & Sons, Inc., New York, 2000.

[13] P. Lötstedt, J. Persson, L. von Sydow, J. Tysk, Space-time adaptive finite difference method for European multi-asset options, Comput. Math. Appl. 53 (2007) 1159–1180, Available electronically in April.

[14] S. Janson, J. Tysk, Feynman–Kac formulas for Black–Scholes type operators, Bull. London Math. Soc. 38 (2006) 269–282.

[15] C.A. Micchelli, Interpolation of scattered data: Distance matrices and conditionally positive definite functions, Constr. Approx. 2 (1) (1986) 11–22.

[16] E. Hairer, S.P. Nørsett, G. Wanner, Solving ordinary differential equations. I, in: Nonstiff Problems, 2nd ed., in: Springer Series in Computational Mathematics, vol. 8, Springer-Verlag, Berlin, 1993.

[17] E. Larsson, K. Åhlander, A. Hall, Multi-dimensional option pricing using radial basis functions and the generalized Fourier transform, Tech. Rep. 2006-037, Dept. of Information Technology, Uppsala Univ., Uppsala, Sweden. Available at: http://www.it.uu.se/research/publications/reports/, 2006; J. Comput. Appl. Math. 222 (1) (2008) 175–192.

[18] E. Larsson, B. Fornberg, Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions, Comput. Math. Appl. 49 (2005) 103–130.

[19] B. Fornberg, G. Wright, Stable computation of multiquadric interpolants for all values of the shape parameter, Comput. Math. Appl. 48 (2004) 853–867.

[20] E. Larsson, B. Fornberg, A numerical study of some radial basis function based solution methods for elliptic PDEs, Comput. Math. Appl. 46 (2003) 891–902.

[21] E. Larsson, U. Pettersson, Fixed shape parameter radial basis function approximations for time-independent PDE problems, 2008 (manuscript in preparation).

[22] A. Greenbaum, Iterative methods for solving linear systems, in: Frontiers in Applied Mathematics, vol. 17, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

Paper II

AN INVESTIGATION OF GLOBAL RADIAL BASIS FUNCTION COLLOCATION METHODS APPLIED TO HELMHOLTZ PROBLEMS[*]

ELISABETH LARSSON[**] AND ULRIKA SUNDIN[**]

Abstract. Global radial basis function (RBF) collocation methods with inifintely smooth basis functions for partial differential equations (PDEs) work in general geometries, and can have exponential convergence properties for smooth solution functions. At the same time, the linear systems that arise are dense and severly ill-conditioned for large numbers of unknowns and small values of the shape parameter that determines how flat the basis functions are. We use Helmholtz equation as an application problem for the theoretical analysis and numerical experiments. We analyse and characterise the convergence properties as a function of the number of unknowns and for different shape parameter ranges. We provide theoretical results for the flat limit of the PDE solutions and investigate when the non-symmetric collocation matrices become singular. We also provide practical strategies for choosing the method parameters and evaluate the results on Helmholtz problems in a curved waveguide geometry.

Key words. Radial basis function, Helmholtz equation, shape parameter, flat limit, error estimate

AMS subject classifications.
65N35, 65D15, 41A30

1. Introduction.

We started writing this paper in 2004. Some of the results can be found in the MSc thesis of the second author [30]. At that time, the first paper on the flat radial basis function (RBF) interpolation limit [5] had just been published, and most of the work on the paper about multivariate flat RBF limits [20] was done, but the paper was not published yet. The focus of research in RBF-based methods for partial differential equations (PDEs) was on global collocation methods, and we were interested in the limit behavior for RBF approximations to PDEs. Then the manuscript ended up 'in a drawer' due to various circumstances, and we came to pick it up again 15 years later. The current research focus has shifted to localized RBF-methods such as RBF-generated finite difference methods (RBF-FD) [10] and RBF partition of unity methods (RBF-PUM) [22]. However, we think that the results in this paper, even though they are on global RBF methods, provide insights that are generally useful also today. The objectives of the work are

- to investigate the approximation errors theoretically and numerically to gain understanding both about the flat limit, the convergence properties, and the dependence on the shape parameter,
- to identify the gaps between theoretical results and numerical behavior,
- to provide practically useful strategies for choosing the method parameters and assessing the results.

The outline of the paper is as follows: In Section 2, we define three different Helmholtz test problems that are used throughout the paper. In Section 3 we derive the systems of equations for non-symmetric and symmetric collocation. Section 4 is devoted to cases where the non-symmetric collocation matrix is singular, and in Section 5, we discuss the limit properties. How to prove these properties is sketched in Appendix A. Section 6 contains a combination of theoretical error estimates, and more heuristic

[*] The first author was supported by a grant from The Swedish Research Council. The second author was funded by the graduate school in Mathematics and Scientific Computing.

[**] Scientific Computing, Department of Information Technology, Uppsala University, Box 337, SE-751 05 Uppsala, Sweden (elisabeth.larsson@it.uu.se, ulrika.sundin@it.uu.se).

error approximations. Then in Section 7, we provide numerical results as well as practical strategies for method parameter selection. The paper ends with a discussion of the results in Section 8.

2. Generic and specific model problems. Throughout the paper, we consider time-independent, linear, partial differential equations (PDEs). We assume that the PDE equation(s), together with the different boundary equations can be summarized as

$$\mathcal{L}^i u(\underline{x}) = f^i(\underline{x}), \quad \underline{x} \in \Omega^i, \quad i = 1, \ldots, N_{\text{op}}, \tag{2.1}$$

where $\mathcal{L}^i$ is a linear operator, u is the solution function, f^i is a given function, $\underline{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$, and $\Omega^i \subseteq \bar{\Omega}$ is a region in the computational domain or a boundary segment.

To give examples and illustrate specific properties, we use a series of Helmholtz problems of increasing complexity. The Helmholtz equation models time-harmonic wave propagation, and in all cases, we consider wave guide problems with a wave originating from a source at the left boundary and propagating to the right. We allow reflected waves from the interior of the domain to propagate back to the left and out through the left boundary, but no waves may enter from outside the right boundary. The main reasons for our choice of model problems are the following:

- There is one problem parameter, the wavenumber κ, that can be varied to study its relation to the RBF method parameters.
- A Helmholtz problem is generally more difficult to solve than a Laplace or Poisson problem, especially for large wavenumbers, due to the indefiniteness of the operator, the wave nature of the solution, and the typically more complicated boundary conditions.

The Helmholtz PDE is in all examples given by

$$\mathcal{L}^1 u(\underline{x}) = -\Delta u(\underline{x}) - \kappa^2 u(\underline{x}) = 0, \quad \underline{x} \in \Omega^1 = \Omega. \tag{2.2}$$

The first and simplest model problem is one-dimensional, with $\Omega = (0, 1)$. The non-reflecting (or radiation) boundary conditions are given by

$$\mathcal{L}^2 u(x) = -\frac{du}{dx}(x) - i\kappa u(x) = -2i\kappa, \quad x = 0, \tag{2.3}$$

$$\mathcal{L}^3 u(x) = \frac{du}{dx}(x) - i\kappa u(x) = 0, \qquad x = 1, \tag{2.4}$$

and the analytical solution is $u(x) = \exp(i\kappa x)$, if κ is constant.

The second problem is two-dimensional with a rectangular domain $\Omega = (0, L_1) \times (0, 1)$. At the top and bottom boundaries, we use the Dirichlet boundary condition

$$\mathcal{L}^4 u(\underline{x}) = u(\underline{x}) = 0, \quad \underline{x} = (0, x_2) \text{ or } \underline{x} = (L_1, x_2), \tag{2.5}$$

indicating that we consider a waveguide type of problem. The conditions at the left and right boundaries are

$$\mathcal{L}^2 u(\underline{x}) = -\frac{\partial u}{\partial x_2}(\underline{x}) - i\beta_m u(\underline{x}) = -2i\beta_m \sin(\alpha_m x_1), \quad \underline{x} = (x_1, 0), \tag{2.6}$$

$$\mathcal{L}^3 u(\underline{x}) = \frac{\partial u}{\partial x_2}(\underline{x}) - i\beta_m u(\underline{x}) = 0, \qquad \underline{x} = (x_1, 1), \tag{2.7}$$

where $\alpha_m = \frac{m\pi}{L_1}$, $\beta_m = \sqrt{\kappa^2 - \alpha_m^2}$, and $m \geq 1$ is an integer. These conditions allow for just one propagating mode in the solution, which is given by $u(\underline{x}) = \exp(i\beta_m x_2)\sin(\alpha_m x_1)$, assuming a constant κ. It should be noted that if κ and m are chosen such that $\beta_m = 0$, the problem is not well-defined, and we avoid such combinations in the experiments.

The third and final problem is also two-dimensional, but the domain Ω is now enclosed between two curves $\gamma_1(x_2) < x_1 < \gamma_2(x_2)$, $x_2 \in (0,1)$, see Figure 2.1. The Dirichlet condition (2.5) is modified to hold at γ_1 and γ_2.

$$\mathcal{L}^4 u(\underline{x}) = u(\underline{x}) = 0, \quad \underline{x} = (\gamma_j(x_2), x_2), \quad j = 1, 2. \tag{2.8}$$

At the left and right boundary, we use so called Dirichlet–to–Neumann map (DtN) radiation boundary conditions [18]

$$\begin{aligned}
\mathcal{L}^2 u(\underline{x}) &= -\frac{\partial u}{\partial x_2} - i\sum_{m=1}^{\infty} \beta_m \langle u(\cdot,0), \psi_m^0\rangle \psi_m^0(x_1) \\
&= -2i\sum_{m=1}^{\infty} A_m \beta_m \psi_m^0(x_1), \qquad\qquad x_2 = 0, \qquad (2.9)\\
\mathcal{L}^3 u(\underline{x}) &= \frac{\partial u}{\partial x_2} - i\sum_{m=1}^{\infty} \beta_m \langle u(\cdot,1), \psi_m^1\rangle \psi_m^1(x_1) = 0, \quad x_2 = 1,
\end{aligned}$$

where, for a fixed x_2, the modes $\psi_m^{x_2} = \sqrt{2}\sin(\alpha_m(x_1 - \gamma_1(x_2)))$, with $\alpha_m = \frac{m\pi}{\gamma_2(x_2)-\gamma_1(x_2)}$. The inner product is given by

$$\langle u(\cdot,x_2), \psi_m^{x_2}\rangle = \int_{\gamma_1(x_2)}^{\gamma_2(x_2)} u(x_1,x_2)\psi_m^{x_2}(x_1)\,dx_1, \tag{2.10}$$

and the amplitudes $A_m = \psi_m^0(x_s)$, where x_s is the position of the source in the vertical coordinate. The amplitudes are chosen to emulate a point source. The DtN conditions allow for any combination of modes to move transparently through the vertical boundaries. For practical and computational reasons, the infinite sum is truncated at $\mu_{x_2} = \lfloor \frac{\kappa(\gamma_2(x_2)-\gamma_1(x_2))}{\pi} \rfloor$. For a discussion of the assumptions behind this truncation and these particular DtN conditions, see [29].

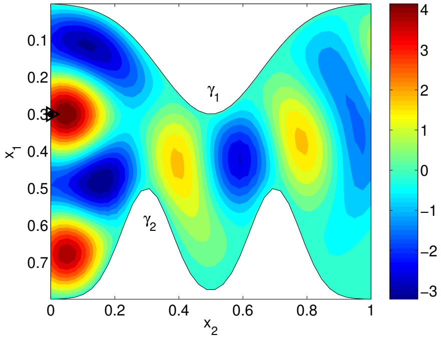


FIG. 2.1. *Wave propagation in an M-shaped duct. The source position is indicated by the marker at the left boundary and the wave number is $\kappa = 6\pi$. The real part of the solution is displayed.*

3. The RBF approximations. In this section, we first describe Kansa's non-symmetric collocation method [17] for our model problems. The main advantage of the non-symmetric collocation method is its simplicity. This is also why we use this method for both numerical and theoretical studies throughout this paper. However, an argument against using non-symmetric collocation is that the RBF approximation matrix, in rare cases [16], can become singular. This is discussed further in Section 4. To avoid singularity, symmetric collocation [43, 6, 14] can instead be employed. This is slightly more involved, especially with non-trivial operators, which is why we include an example of how to do this for the one-dimensional model problem.

3.1. Non-symmetric collocation. When we use non-symmetric collocation to discretize the problem (2.1), the RBF approximant is given by

$$s(\underline{x}) = \sum_{j=1}^{N} \lambda_j \phi(\varepsilon \| \underline{x} - \underline{x}_j \|) = \sum_{j=1}^{N} \lambda_j \phi_j(\underline{x}), \tag{3.1}$$

where $\underline{x}_j$, $j = 1, \ldots, N$ are the RBF center points and ε is the shape parameter. The collocation conditions are imposed at the N center points. Let $\underline{x}_j^k$, $j = 1, \ldots, N_k$ be the subset of center points that belong to the region or section Ω^k. The corresponding operator is used for collocation, and we get the equations

$$\mathcal{L}^k s(\underline{x}_i^k) = \sum_{j=1}^{N} \lambda_j \mathcal{L}^k \phi_j(\underline{x}_i^k) = f^k(\underline{x}_i^k), \quad i = 1, \ldots, N_k, \quad k = 1, \ldots, N_{\text{op}}.$$

If the points are ordered according to the set affiliation, we get a system of equations, $M\underline{\lambda} = \underline{f}$, with the following general block structure

$$\begin{pmatrix} \mathcal{L}^1 \phi \\ \vdots \\ \mathcal{L}^{N_{\text{op}}} \phi \end{pmatrix} \begin{pmatrix} \underline{\lambda} \end{pmatrix} = \begin{pmatrix} \underline{f}^1 \\ \vdots \\ \underline{f}^{N_{\text{op}}} \end{pmatrix}, \tag{3.2}$$

where the block $\mathcal{L}^k \phi$ is of size $(N_k \times N)$.

Applying the operators in the specific model problems to the RBFs is straightforward, except for the DtN operators in (2.9). The left boundary condition applied to one of the RBFs and evaluated at the point $\underline{x} = (x_1, 0)$ takes the form

$$\mathcal{L}^2 \phi_j(\underline{x}) = -\frac{\partial \phi_j}{\partial x_2}(\underline{x}) - i \sum_{m=1}^{\mu_0} \beta_m \langle \phi_j(\cdot, 0), \psi_m^0 \rangle \, \psi_m^0(x_1).$$

To form the whole block $\mathcal{L}^2 \phi$, we need to evaluate $\mu_0 \cdot N$ inner products. This cannot in general be done analytically for infinitely smooth RBFs such as multiquadrics, inverse quadratics, or Gaussians.

One of our aims with choosing the Helmholtz model problems was to see if using RBFs would make it difficult to implement non-trivial boundary conditions. There are no fundamental issues preventing implementation of boundary conditions involving linear functionals applied to the basis functions. A practical issue is that the computational cost for the quadrature is quite large, although linear in N. In Section 7, we investigate how accurately we need to compute the inner products to not destroy the overall accuracy of the solution. The experiments show that we need to compute the inner products more accurately than the overall error tolerance, which increases the cost further.

3.2. Symmetric collocation. Non-singularity of the RBF approximation matrix can be ensured through symmetric collocation [43, 6, 14]. The idea is to view the RBF $\phi(\varepsilon\|\underline{x}-\underline{\xi}\|)$ as a function of two variables $\psi(\underline{x},\underline{\xi})$. Then in the ansatz for the RBF approximation, for each basis function, the operator corresponding to its center location is applied to the second argument of the basis function. Since we consider complex operators, we also need to conjugate the operators in order to get a Hermitian matrix in the end. The approximation then takes the form

$$s(\underline{x}) = \sum_{k=1}^{N_{\text{op}}}\sum_{j=1}^{N_k} \lambda_j^k \overline{\mathcal{L}_\xi^k}\psi(\underline{x},\underline{x}_j^k).$$

For the one-dimensional Helmholtz problem, collocation with this ansatz leads to a system of equations with the following structure

$$\begin{pmatrix} \mathcal{L}_x^1\overline{\mathcal{L}_\xi^1}\psi & \mathcal{L}_x^1\overline{\mathcal{L}_\xi^2}\psi & \mathcal{L}_x^1\overline{\mathcal{L}_\xi^3}\psi \\ \mathcal{L}_x^2\overline{\mathcal{L}_\xi^1}\psi & \mathcal{L}_x^2\overline{\mathcal{L}_\xi^2}\psi & \mathcal{L}_x^2\overline{\mathcal{L}_\xi^3}\psi \\ \mathcal{L}_x^3\overline{\mathcal{L}_\xi^1}\psi & \mathcal{L}_x^3\overline{\mathcal{L}_\xi^2}\psi & \mathcal{L}_x^3\overline{\mathcal{L}_\xi^3}\psi \end{pmatrix} \begin{pmatrix} \underline{\lambda}^1 \\ \underline{\lambda}^2 \\ \underline{\lambda}^3 \end{pmatrix} = \begin{pmatrix} \underline{0} \\ -2i\kappa \\ 0 \end{pmatrix},$$

where the block $\mathcal{L}_x^j\overline{\mathcal{L}_\xi^k}\psi$ is of size $(N_j \times N_k)$. To see that the coefficient matrix M really is Hermitian, we can use the following differentiation rules for the RBFs

$$\frac{\partial^n}{\partial \xi^n}\psi(\underline{x}_j,\underline{x}_k) = (-1)^n\frac{\partial^n}{\partial x^n}\psi(\underline{x}_j,\underline{x}_k), \tag{3.3}$$

$$\frac{\partial^n}{\partial x^n}\psi(\underline{x}_k,\underline{x}_j) = (-1)^n\frac{\partial^n}{\partial x^n}\psi(\underline{x}_j,\underline{x}_k). \tag{3.4}$$

We can then show for the different blocks in the matrix that the matrix elements satisfy $m_{jk} = \overline{m}_{kj}$. As an example, for elements in the first two off-diagonal blocks we get

$$\begin{aligned} \mathcal{L}_x^1\overline{\mathcal{L}_\xi^2}\psi(\underline{x}_j,\underline{x}_k) &= (-\frac{\partial^2}{\partial x^2}-\kappa^2)(-\frac{\partial}{\partial \xi}+i\bar{\kappa})\psi(\underline{x}_j,\underline{x}_k) = (-\frac{\partial^2}{\partial x^2}-\kappa^2)(\frac{\partial}{\partial x}+i\bar{\kappa})\psi(\underline{x}_j,\underline{x}_k), \\ \overline{\mathcal{L}_x^2\overline{\mathcal{L}_\xi^1}}\psi(\underline{x}_k,\underline{x}_j) &= (-\frac{\partial}{\partial x}+i\bar{\kappa})(-\frac{\partial^2}{\partial \xi^2}-\kappa^2)\psi(\underline{x}_k,\underline{x}_j) = (\frac{\partial}{\partial x}+i\bar{\kappa})(-\frac{\partial^2}{\partial x^2}-\kappa^2)\psi(\underline{x}_j,\underline{x}_k). \end{aligned}$$

Apart from the important non-singularity property, limited numerical experiments also show that the conditioning is slightly better (one order of magnitude) than for the non-symmetric method. However, the error curves, as functions of both x and ε, are close to identical.

It would be complicated to implement the symmetric collocation method for the two-dimensional problem with DtN boundary conditions. It would also be even more costly than for the non-symmetric case, because of the increased number of integrals to compute. As mentioned for example in [16], when using non-symmetric collocation, singular matrices are hardly ever observed. Due to its simplicity, non-symmetric collocation is more widely used than symmetric collocation. In the following, we choose to study the properties of the non-symmetric collocation method.

4. Singularity of the RBF collocation matrix. As already stated, the RBF collocation matrix may become singular with the non-symmetric collocation approach. This becomes particularly clear for problems with a parameter that can be varied freely as for our Helmholtz examples. For the one-dimensional Helmholtz model problem, we can in fact show that for any given node distribution (with distinct nodes) there are always wavenumbers κ that lead to a singular collocation matrix.

To get the equations in an appropriate form for eigenvalue analysis, we multiply the PDE (2.2) with -1 and the boundary conditions (2.3) and (2.4) with $i\kappa$. After collocation with the PDE at the interior points x_j^1, and the boundary conditions at the boundary points, we get a collocation matrix M with elements

$$m_{jk} = \left\{ \begin{array}{llll} \kappa^2\phi_k(x_j^1) & + & \phi_k''(x_j^1), & j = 1,\dots,N-2, \quad k = 1,\dots,N, \\ \kappa^2\phi_k(0) & - & i\kappa\phi_k'(0) \quad , & j = N-1, \quad k = 1,\dots,N, \\ \kappa^2\phi_k(1) & + & i\kappa\phi_k'(1) \quad , & j = N, \quad k = 1,\dots,N. \end{array} \right.$$

We can express M as a matrix polynomial in κ,

$$M = \kappa^2 A + \kappa i B + C,$$

where A, B, and C are real matrices. Furthermore, A is the usual RBF interpolation matrix. The question of singularity of M can be posed as a quadratic eigenproblem

$$(\kappa^2 A + \kappa i B + C)\underline{v} = \underline{0}. \tag{4.1}$$

For standard RBFs and distinct points, A is non-singular. By introducing $\underline{w} = \kappa\underline{v}$ we can then reformulate (4.1) as a standard eigenvalue problem

$$\left(\begin{array}{cc} 0 & I \\ -A^{-1}C & -iA^{-1}B \end{array}\right)\left(\begin{array}{c} \underline{v} \\ \underline{w} \end{array}\right) = \kappa \left(\begin{array}{c} \underline{v} \\ \underline{w} \end{array}\right).$$

Solving this problem leads to $2N$ eigenvalues. That is, values of κ for which the collocation matrix M is singular. Two of the eigenvalues have to be $\kappa = 0$ because of the scaling of the boundary conditions. By conjugating equation (4.1), we get

$$(\bar{\kappa}^2 A - \bar{\kappa} i B + C)\underline{\bar{v}} = ((-\bar{\kappa})^2 A + (-\bar{\kappa}) i B + C)\underline{\bar{v}} = \underline{0}.$$

That is, if $(\kappa, \underline{v})$ is an eigenvalue–eigenvector pair, then $(-\bar{\kappa}, \underline{\bar{v}})$ also is. Hence, all eigenvalues with $\mathrm{Re}(\kappa) \neq 0$ must come in pairs $(\kappa, -\bar{\kappa})$. Then, there may also be a number of eigenvalues on the imaginary axis. The κ that are of interest in the Helmholtz problem are such that $\mathrm{Re}(\kappa) > 0$. We are then left with a maximum of $N-1$ potentially interesting wavenumbers that lead to a singular problem.

In Figure 4.1, the eigenvalues that lead to a singular system are computed for different problem sizes using multiquadric and Gaussian RBFs. For multiquadrics, there are no eigenvalues in the region of interest, that is, real wavenumbers with solutions that are well resolved. The eigenvalues with the largest real part are closest to the real axis. These problems are resolved to $2\pi N/\kappa \approx 2$ points per wavelength, which is the theoretical lowest possible resolution to use for a wave propagation problem. The Gaussian approximation produces eigenvalues that are closer to the real axis, but also here the eigenvalues with large real part correspond to badly resolved problems. It should be noted that complex wavenumbers, typically with a significantly smaller imaginary part than real part, are used in practical applications to model damping within the media that the waves propagate through.

5. The flat RBF limit for PDE problems. The limits of multivariate RBF interpolants as the shape parameter ε goes to zero were analyzed in [20, 37, 25, 39, 24]. The same type of limits for finitely smooth RBFs where also studied in [42, 23]. It

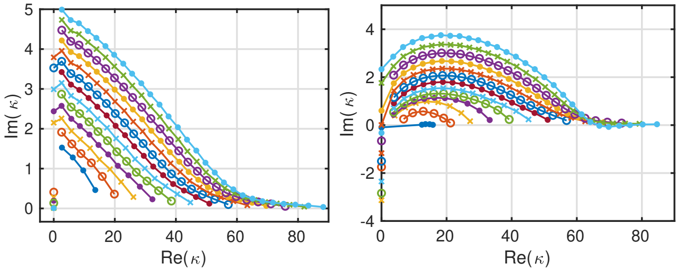


Fig. 4.1. *The wavenumbers that lead to a singular system for the one-dimensional problem using* $N = 6, 8, \dots, 30$ *from bottom to top, for multiquadric RBFs with* $\varepsilon = 5$ *(left) and Gaussian RBFs with* $\varepsilon = 10$ *(right).*

was shown that the limit behavior is closely related to polynomial unisolvency [2] on the set of node points. We define

$$N_{K,d} = \begin{pmatrix} K+d \\ K \end{pmatrix}, \tag{5.1}$$

which is the dimension of the space of polynomials of degree K in $\mathbb{R}^d$. If $N = N_{K,d}$, and the node set is unisolvent, then the (infinitely smooth) flat limit RBF interpolant reproduces the multivariate polynomial interpolant of degree K on these nodes.

When we apply the non-symmetric collocation method to a PDE problem, the RBF approximant has the same general form (3.1), and we can derive corresponding results for the limit.

In order to express the conditions for different limit results, we need to define two matrices, P and Q, from which we can determine polynomial unisolvency and unisolvency of the discrete PDE problem. Let $\{p_j(\underline{x})\}_{j=1}^N$ be N linearly independent monomials of minimal degree in d dimensions. For example, for $N = 7$ and $d = 2$, we can choose $\{1, x, y, x^2, xy, y^2, x^3\}$. If $N_{K-1,d} < N \leq N_{K,d}$, then the degree of $p_N(\underline{x})$ is K.

The set of node points $\{x_i\}_{i=1}^N$ satisfies polynomial unisolvency if there, for any given data at the node points, is a unique linear combination $\sum_{j=1}^N \beta_j p_j(\underline{x})$ that interpolates the data. This is equivalent to non-singularity of the matrix

$$P = \begin{pmatrix} p_1(\underline{x}_1) & p_2(\underline{x}_1) & \cdots & p_N(\underline{x}_1) \\ p_1(\underline{x}_2) & p_2(\underline{x}_2) & \cdots & p_N(\underline{x}_2) \\ \vdots & \vdots & & \vdots \\ p_1(\underline{x}_N) & p_2(\underline{x}_N) & \cdots & p_N(\underline{x}_N) \end{pmatrix}. \tag{5.2}$$

In cases where P is singular, we instead construct a minimal non-degenerate basis [20]. Such a basis can be constructed by choosing N monomials of smallest possible degree under the constraint that they give linearly independent columns in the matrix P. The highest selected monomial degree M is then also the degree of $p_N(\underline{x})$. As an example, for $N = 5$ node points all located on the line $x = y$, a minimial non-degenerate basis is $\{1, x, x^2, x^3, x^4\}$ and $M = 4$.

Unisolvency of the discrete PDE problem on the set of node points $\{x_i\}_{i=1}^N$ with respect to $\{p_j(\underline{x})\}_{j=1}^N$ requires a unique linear combination $\sum_{j=1}^N \beta_j p_j(\underline{x})$ that satisfy the collocation conditions

$$\sum_{j=1}^{N} \beta_j \mathcal{L}^k p_j(\underline{x}_i^k) = f^k(\underline{x}_i^k), \quad i = 1, \ldots, N_k, \quad k = 1, \ldots, N_{\text{op}}.$$

This is equivalent to non-singularity of the matrix

$$Q = \begin{pmatrix} \mathcal{L}^1 p_1(\underline{x}_1^1) & \cdots & \mathcal{L}^1 p_N(\underline{x}_1^1) \\ \vdots & & \vdots \\ \mathcal{L}^{N_{\text{op}}} p_1(\underline{x}_{N_{N_{\text{op}}}}^{N_{\text{op}}}) & \cdots & \mathcal{L}^{N_{\text{op}}} p_N(\underline{x}_{N_{N_{\text{op}}}}^{N_{\text{op}}}) \end{pmatrix}. \tag{5.3}$$

As in [20], we need the RBFs to fulfill three conditions in order to get the results in the theorem given below. We repeat the conditions and discuss their validity briefly here, but for a full explanation, we refer the reader to [20].

(I) The RBF $\phi(r)$ can be Taylor expanded as $\phi(r) = \sum_{j=0}^{\infty} a_j r^{2j}$.

(II) The PDE collocation matrix M in system (3.2) is non-singular in the interval $0 < \varepsilon \leq R$, for some $R > 0$.

(III) Certain matrices $A_{p,J}$, built from the coefficents a_j in the Taylor expansion of $\phi(r)$, are non-singular for $0 \leq p \leq d$ and $0 \leq J \leq K$.

Condition (I) is true for all infinitely smooth RBFs that are commonly used. Condition (II) is likely to hold for some value of R, but the previous section shows that M can become singular at any ε, given a specific combination of PDE problem and node points. Condition (III) was shown to hold for these RBFs in [25].

The following theorem gives the different possibilities for the limiting RBF approximant as the shape parameter $\varepsilon \to 0$.

Theorem 5.1. *Assume that the RBF $\phi(r)$ fulfills conditions (I)–(III) and that the number of node points satisfy $N_{K-1,d} < N \leq N_{K,d}$. The degree of a minimal non-degenerate basis for the point set is either K for a unisolvent set or M for a non-unisolvent set. If*

(i) P and Q are non-singular, the limit exists and is a polynomial of deg K. If $N = N_{K,d}$ it is the unique degree K polynomial solution to the discrete PDE problem, otherwise the final polynomial depends on the choice of RBF.

(ii) P is singular, but Q is non-singular, the limit exists and is an RBF-dependent polynomial of degree M.

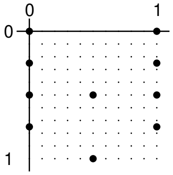
(iii) P is non-singular, but Q is singular, divergence will occur unless the right hand side $\underline{f}$ of system (3.2) happens to be in the range of Q. If there is just a single null-space polynomial $n(\underline{x})$ of degree K, the divergent term is proportional to $\varepsilon^{-2} n(\underline{x})$.

(iv) P has a nullspace of dimension $m > 0$ and Q has a nullspace of dimension $p > 0$, then if $m \geq p$ the limit is likely, but not certain to exist. If it exists it is of degree M. If $m < p$ divergence is likely, but not certain.

The proof builds on the results for RBF interpolation in [20]. The key arguments and differences are pointed out in Appendix A.

The uncomplicated case (i) is of course the most common and the other types are deviations stemming from degenerate node point configurations or specific combinations of PDE problem parameters and node points that lead to degeneracy. Below, we give an example of each type of degeneracy for the two-dimensional Helmholtz problem given by (2.2) and (2.5)–(2.7) with $m = 1$.

Example (ii): The node set is not polynomially unisolvent.

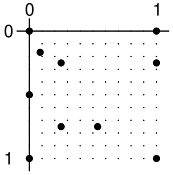


The $N = 10$ points are $(1/2, 1/2)$, $(1, 1/2)$, and $(k/4, 0)$, $(k/4, 1)$, $k = 0, \ldots, 3$. The matrix P has a nullspace defined by $n(\underline{x}) = x_2(x_2 - \frac{1}{2})(x_2 - 1)$. A non-degenerate basis is given by $\{1,\, x_1,\, x_2,\, x_1^2,\, x_1x_2,\, x_2^2,\, x_1^3,\, x_1^2x_2,\, x_1x_2^2,\, x_1^4\}$ with $M = 4$. The limit is hence a polynomial of degree four whose coefficients depend on the choice of RBF. To illustrate what this dependence can look like, we give the general form of the limit polynomial $p(\underline{x})$.

$$\begin{aligned}
p(\underline{x}) &= \beta_0 + \beta_1 x_2 + \beta_2 x_1 + \beta_3 x_2^2 + \beta_4 x_2 x_1 + \beta_5 x_1^2 \\
&\quad + \beta_6 \left[-12a_3(x_2^3 + 3x_2x_1^2) + \frac{8a_2^2}{a_1}(x_2^3 + x_2x_1^2) \right] \\
&\quad + \beta_7 \left[-12a_3(x_1^3 + 3x_2^2x_1) + \frac{8a_2^2}{a_1}(x_1^3 + x_2^2x_1) \right] \\
&\quad + \beta_8 \left[-4a_3(5x_1^3 + 3x_2^2x_1) + \frac{8a_2^2}{a_1}(x_1^3 + x_2^2x_1) \right] \\
&\quad + \beta_9 \Bigg[\quad - 4a_4(9x_2^3 + 36x_2^2x_1 + 45x_2x_1^2 + 20x_1^3 - 24x_2^3x_1 - 40x_2x_1^3) \\
&\quad + \frac{6a_2a_3}{a_1}(3x_2^3 + 4x_2^2x_1 + 3x_2x_1^2 + 4x_1^3) - \frac{72a_3^2}{a_2}(x_2^3x_1 + x_2x_1^3) \Bigg],
\end{aligned}$$

where a_j are the Taylor expansion coefficients of the RBF, and β_j are the ten unknown coefficients that are determined by the ten discrete PDE collocation conditions. The result is κ-dependent as well as RBF-dependent.

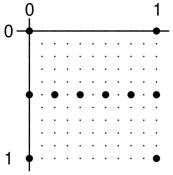
Example (iii): The node set is not PDE-unisolvent.



The $N = 10$ points are $(0, 0)$, $(1/2, 0)$, $(1, 0)$, $(0, 1)$, $(1/4, 1)$, $(1, 1)$, $(1/6, (2545 - 23\sqrt{9233})/3936)$, $(1/4, 1/4)$, $(3/4, 1/4)$, and $(3/4, 969/1804)$. For $\kappa = 4\sqrt{246}/9$ the matrix Q has a nullspace defined by $q(\underline{x}) = -\frac{5}{32}x_2(x_2+1) + \frac{x_1}{16}(8 - 24x_1 + 3x_2 + 16x_1^2 + 4x_1x_2 - 7x_2^2)$.

In this case, we get divergence of order ε^{-2} as $\varepsilon \to 0$ for all RBFs that obey conditions (I)–(III). This can be observed not only in exact arithmetic, but also in for example a double precision numerical simulation. However, if we move just one of the points or change κ slightly, there is no longer a nullspace. This kind of degeneracy is very rare, since it requires very special combinations of wavenumber and node points.

Example (iv): Both P and Q have a nullspace.



The $N = 10$ points are $(0, 0)$, $(0, 1)$, $(1, 0)$, $(1, 1)$ and $(1/2, k/5)$, $k = 0, \ldots, 5$. The matrices P and Q have a common nullspace of dimension two defined by $q_1(\underline{x}) = x_1(x_1 - \frac{1}{2})(x_1 - 1)$ and $q_2(\underline{x}) = x_2(x_1 - \frac{1}{2})(x_2 - 1)$. The limit exists and is an RBF- and κ-dependent polynomial of degree $M = 5$.

A practical implication of this result is that if we use node sets that are not unisolvent, e.g, Cartesian nodes, both PDE approximation and interpolation are expected to behave poorly in the small shape parameter regime. The condition number of the linear system is larger than in the unisolvent case, and the result contains a term that diverges as $\varepsilon \to 0$.

An important property of interpolation with Gaussian RBFs is that it never diverges [37]. In the PDE case, this property also holds as long as Q is non-singular. However, it can still be difficult to compute the limit numerically using a stable evaluation method. The RBF-QR method derived in [13, 11], and further explored for solving PDEs [21] uses pivoting to handle non-unisolvent cases. This means that a limit can always be computed, but it may be different from the Gaussian limit. The RBF-GA method [12] always reproduces the correct Gaussian limit, but instead cannot handle large values of N.

6. Convergence properties and error estimates.

In this section, we investigate errors and convergence properties from different perspectives, as well as quantify how the choice of shape parameter affects the results. We start by formulating general residual-based error estimates in the following subsection.

6.1. General error estimates using Green's functions.

We define the error function as the difference between the RBF approximant and the exact solution to the PDE problem (2.1)

$$e(\underline{x}) = s(\underline{x}) - u(\underline{x}). \tag{6.1}$$

In the interpolation case, the error and the residual are the same, and if the function $u(\underline{x})$ is known at a point, the corresponding error can be explicitly computed. In the PDE case, we can compute the residual for each operator, while the error is governed by the same type of PDE as the solution

$$\mathcal{L}^i e(\underline{x}) = \mathcal{L}^i s(\underline{x}) - f^i(\underline{x}) \equiv r^i(\underline{x}), \quad \underline{x} \in \Omega^i, \quad i = 1, \ldots, N_{\text{op}}, \tag{6.2}$$

where r^i are residuals. One way to find the error is to solve the above PDE problem. However, because the residuals are zero at the collocation points, they are highly oscillatory and more node points are required to approximate the error than to solve the original PDE.

Instead of solving the error equation, we can formulate *a posteriori* error estimates in terms of the residual. Writing out the error PDE for the one-dimensional problem we get

$$\left\{ \begin{array}{rcl} -\Delta e(x) - \kappa^2 e(x) & = & r(x), \\ -e'(0) - i\kappa e(0) & = & 0, \\ e'(1) - i\kappa e(1) & = & 0. \end{array} \right. \tag{6.3}$$

A Green's function satisfying the boundary conditions is given by

$$G(x,\xi) = \frac{i}{2\kappa} e^{i\kappa|x-\xi|}, \tag{6.4}$$

with

$$\frac{\partial G}{\partial \xi} = \left\{ \begin{array}{ll} \frac{1}{2} e^{i\kappa|x-\xi|}, & x \geq \xi, \\ -\frac{1}{2} e^{i\kappa|x-\xi|}, & x < \xi, \end{array} \right. \quad \text{and } \Delta_\xi G = -\frac{i\kappa}{2} e^{i\kappa|x-\xi|} - \delta(x), \tag{6.5}$$

such that $-\Delta_\xi G - \kappa^2 G = \delta(x)$, which allows us express the error as

$$e(\xi) = \int_0^1 G(x,\xi) r(x)\, dx. \tag{6.6}$$

We can use this form of the error to formulate an error estimate as

$$\|e\|_\infty \le \int_0^1 |G(x,\xi)||r(x)|\,dx = \frac{1}{2\kappa}\int_0^1 |r(x)|dx \le \frac{1}{2\kappa}\|r\|_\infty. \tag{6.7}$$

For the two-dimensional Helmholtz problem in a rectangular domain, the corresponding Green's function satisfying the boundary conditions is given by

$$G(\underline{x},\underline{\xi}) = \sum_{m=1}^{\infty} \frac{i}{2\beta_m} e^{i\beta_m|x_2-\xi_2|}\psi_m(x_1)\psi_m(\xi_1), \tag{6.8}$$

with $-\Delta_\xi G - \kappa^2 G = \delta(x_2)\sum_{m=1}^{\infty}\psi_m(x_1)\psi_m(\xi_1)$. Similarly as for the one-dimensional problem we define the error as

$$e(\underline{\xi}) = \int_0^1\int_0^{L_1} G(\underline{x},\underline{\xi})r(\underline{x})\,dx_1\,dx_2. \tag{6.9}$$

To see how this works, we note that the vertical eigenfunctions form an orthonormal basis, and we can express the residual as

$$r(x_1,x_2) = \sum_{m=1}^{\infty}\langle r(\cdot,x_2),\psi_m^{x_2}\rangle\psi_m^{x_2} \equiv \sum_{m=1}^{\infty} r_m(x_2)\psi_m(x_1). \tag{6.10}$$

This allows us to simplify the error expression

$$\begin{aligned} e(\underline{\xi}) &= \sum_{m=1}^{\infty}\frac{i}{2\beta_m}\int_0^1 e^{i\beta_m|x_2-\xi_2|}\int_0^{L_1}\psi_m(x_1)\psi_m(\xi_1)\sum_{n=1}^{\infty} r_n(x_2)\psi_n(x_1)\,dx_1\,dx_2 \\ &= \sum_{m=1}^{\infty}\frac{i}{2\beta_m}\psi_m(\xi_1)\int_0^1 e^{i\beta_m|x_2-\xi_2|}r_m(x_2)\,dx_2 \end{aligned} \tag{6.11}$$

To convert this into error estimate, we first note that for $m \le \mu_0 = \lfloor \kappa L_1/\pi \rfloor$, the horizontal wavenumber β_m is real, and $|e^{i\beta_m|x_2-\xi_2|}| = 1$, while for $m > \mu_0$, β_m is purely imaginary and $|\int_0^1 e^{i\beta_m|x_2-\xi_2|}\,dx_2| \le |\int_0^1 e^{i\beta_m|x_2-\frac{1}{2}|}\,dx_2| = \frac{2}{|\beta_m|}(1-e^{-\frac{1}{2}|\beta_m|})$. Then we get

$$\begin{aligned} \|e\|_\infty &\le \sum_{m=1}^{\mu_0}\frac{1}{\sqrt{2}\beta_m}\int_0^1 |r_m(x_2)|\,dx_2 + \sum_{m=\mu_0+1}^{\infty}\frac{\sqrt{2}}{|\beta_m|^2}(1-e^{-\frac{1}{2}|\beta_m|})\int_0^1 |r_m(x_2)|\,dx_2 \\ &\le \sum_{m=1}^{\mu_0}\frac{1}{\sqrt{2}\beta_m}\|r_m\|_\infty + \sum_{m=\mu_0+1}^{\infty}\frac{\sqrt{2}}{|\beta_m|^2}(1-e^{-\frac{1}{2}|\beta_m|})\|r_m\|_\infty. \end{aligned} \tag{6.12}$$

For the two-dimensional problem in a domain with curved boundaries, we cannot provide an explicit Green's function. If we think about the curved domain as a sequence of thin almost rectangular domains, we can modify the previous estimate to get a heuristic approximation of the error

$$\begin{aligned} \|e\|_\infty \approx &\sum_{m=1}^{\infty}\int_{\Re e(\beta_m)>0}\frac{|r_m(x_2)|}{\sqrt{2}\beta_m}\,dx_2 \\ &+ \sum_{m=1}^{\infty}\int_{\Im m(\beta_m)>0}\frac{\sqrt{2}}{|\beta_m|^2}(1-e^{-\frac{1}{2}|\beta_m|})|r_m(x_2)|\,dx_2. \end{aligned} \tag{6.13}$$

We evaluate this error approximation numerically in Section 7 and find that we get surprisingly good results.

6.2. Convergence properties for small ε. As discussed in the previous section, we approach the polynomial limit, $s(\underline{x}) = p(\underline{x})$, as $\varepsilon \to 0$. For polynomial *interpolation* in one dimension, the interpolation error $e_I(x)$ takes the form

$$e_I(x) = u(x) - p(x) = \frac{\prod_{j=1}^{N}(x - x_j)}{N!} u^{(N)}(\xi),$$

where $\xi \in (x_1, x_N)$. For equispaced points, $x_{j+1} - x_j = h$, this can be estimated by

$$|e_I(x)| \le \frac{h^N}{N} \max_{\xi \in (x_1, x_N)} |u^{(N)}(\xi)|,$$

see [15, pp. 39–40]. In the PDE case, the residual plays the role of the error. By following the steps for the proof of the polynomial error [15, pp. 43–44], we can get a similar estimate for the residual.

Theorem 6.1. *For a one-dimensional linear PDE problem*

$$\begin{cases} \mathcal{L}^1 u(x) &=& f^1(x), & x_1 < x < x_N, \\ \mathcal{L}^2 u(x) &=& f^2(x), & x = x_1, \\ \mathcal{L}^3 u(x) &=& f^3(x), & x = x_N, \end{cases}$$

with a polynomial solution $p(x)$ determined through collocation at the nodes x_i, $i = 1, \ldots, N$ the residual $r(x) = \mathcal{L}^1 p(x) - f(x)$ has the form

$$r(x) = \frac{\prod_{j=2}^{N-1}(x - x_j)}{(N-2)!} r^{(N-2)}(\xi),$$

where $\xi \in (x_1, x_N)$. For equispaced points, $x_{j+1} - x_j = h$, this can be estimated by

$$|r(x)| \le \frac{h^{N-2}}{N-2} \max_{\xi \in (x_1, x_N)} |r^{(N-2)}(\xi)|.$$

Proof. Let $\Psi(s) = r(s) - \frac{r(x)}{\chi(x)}\chi(s)$, where $\chi(x) = \prod_{j=2}^{N-2}(x - x_j)$. Then $\Psi(x) = 0$ and $\Psi(x_j) = 0$, $j = 2, \ldots, N-2$, since $r(x_j) = 0$ at all interior node points where the equation is enforced. This means that $\Psi(s)$ has at least $N-1$ zeros. By repeated application of Rolle's theorem, we find that $\Psi^{(N-2)}(s)$ has at least one zero. That is,

$$0 = \Psi^{(N-2)}(\xi) = r^{(N-2)}(\xi) - \frac{r(x)}{\chi(x)}(N-2)!.$$

Rearranging gives the expression for $r(x)$. □

To see how this can help us in understanding the behavior of the error for small ε, we insert the residual estimate in the error estimate (6.7) for the one-dimensional Helmholtz problem to get

$$\|e\|_\infty \le \frac{1}{2\kappa} \frac{h^{N-2}}{N-2} \|r^{(N-2)}\|_\infty. \tag{6.14}$$

In the flat limit, the residual is $r(x) = -p''(x) - \kappa^2 p(x)$, where $p(x)$ is the limit polynomial of degree $N-1$. Then $r^{(N-2)}(x) = -\kappa^2 p^{(N-2)}(x)$. We know that $p(x) \approx u(x) = \exp(i\kappa x)$, but even if $p(x)$ is a very good approximation of $u(x)$, $p^{(N-2)}(x)$ (which is a line) is a rather poor approximation of $u^{(N-2)}(x)$. However, numerical tests indicate that the order of magnitude is right. That is, $|p^{(N-2)}| \approx |\frac{d^{N-2}\exp(i\kappa x)}{dx^{N-2}}| = \kappa^{N-2}$. We cannot use this as a bound, but we get an approximate expression for the error in the limit

$$\|e\|_\infty \approx \frac{1}{2\kappa}\frac{h^{N-2}}{N-2}\kappa^2\kappa^{N-2} = \frac{\kappa(\kappa h)^{N-2}}{2(N-2)} \approx \frac{1}{2}(\kappa h)^{N-1}. \tag{6.15}$$

Note that the quantity κh is small only if the problem is adequately resolved.

For the two-dimensional problem, the limit polynomial has degree K if $N_{K-1,d} < N \le N_{K,d}$ and it is zero at the interior node points. To get an estimate for the residual in terms of its derivatives, we could potentially use a sampling inequality such as [27, Theorem 3.5], which says that for all $h \le h_0$

$$\|r\|_\infty \le C_k h^k \sum_{|\sigma|=k} \|D^\sigma r\|_\infty, \tag{6.16}$$

where h_0 depends on the geometry of Ω. For the unit square, which we are using here, $h_0 = \frac{1}{2kc_2}$ with $c_2 = 12$. In the discretizations that we use $h = 1/(\sqrt{N}-1)$. Requiring $h \le h_0$ leads to the following condition $k \le (\sqrt{N}-1)/24$. We want to use the theorem for $k = K-1$, where inverting the expression for $N_{K,d}$ yields that $K = \lfloor \sqrt{2}\sqrt{N+1/8} - 1.5 \rfloor$. That is, the theorem does not hold in this case. From practical experience, the result holds also for larger k (larger h), and we will therefore use it to approximate the residual.

In this case, using that $r(\underline{x}) = -\Delta p(\underline{x}) - \kappa^2 p(\underline{x})$, and, for $|\sigma| = K-1$, $D^\sigma r(\underline{x}) = -\kappa^2 D^\sigma p(\underline{x}) \approx -\kappa^2 D^\sigma u(\underline{x})$, we get

$$\|r\|_\infty \le C_{K-1} h^{K-1}\kappa^2 \sum_{|\sigma|=K-1} \beta_1^{\sigma_2}\alpha_1^{\sigma_1} \le C_{K-1}\kappa^2 K(\kappa h)^{K-1},$$

Combining the approximate expression for the residual with the error estimate (6.12) restricted to the first mode (scaled by $1/\sqrt{2}$) gives

$$\|e\|_\infty \approx \frac{1}{2|\beta_1|} C_{K-1}\kappa^2 K(\kappa h)^{K-1}$$

Numerical experiments show that $KC_{K-1} = C/(K-1)$, provides the appropriate behavior with respect to N (both K and h are coupled with N). This leads to

$$\|e\|_\infty \approx \frac{C\kappa^2(\kappa h)^{K-1}}{2|\beta_1|(K-1)} \approx \tilde{C}(\kappa h)^K,$$

where the final expression is just to show that the dimension is similar to that of the one-dimensional error approximation.

Figure 6.1 shows the computed errors of the one-dimensional and two-dimensional problems for small values of the shape parameter. The error behavior agrees well with the derived error approximations. For the two-dimensional problem, we also show that the error expression can be multiplied by a constant to get a very good fit to the actual error. This means that we can use $\|e\|_\infty \approx C(\kappa h)^K$ *a priori* with $C = 1$ to determine the necessary resolution for a given tolerance. Given at least two numerical solutions, we can also estimate the constant C. The error approximation is most likely to be valid for problems that are almost rectangular or with mildly varying coefficients, but only for small shape parameter values.

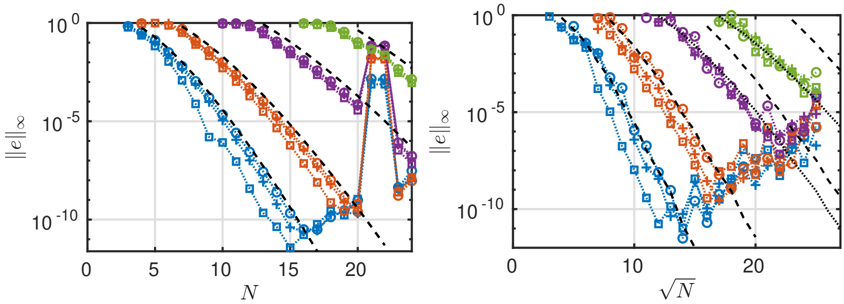


FIG. 6.1. *The computed errors using the Gaussian RBF and the RBF-QR method for* $\varepsilon = 0.5$ *(□),* $\varepsilon = 0.25$ *(+), and* $\varepsilon = 0.01$ *(○) for* $\kappa = \pi, 2\pi, 4\pi, 6\pi$*, from left to right, together with the approximation* $\|e\|_\infty \approx \frac{1}{2}(\kappa h)^{N-1}$ *for the one-dimensional problem (left), and for* $\kappa = 1.2\pi, 2.4\pi, 4.8\pi, 7.2\pi$*, from left to right, together with the approximation* $\|e\|_\infty \approx (\kappa h)^K$ *for the two-dimensional problem (right) (dashed curves). For* $\kappa = 4.8\pi$ *and* 7.2π *in the two-dimensional case, we also show the error approximation using* $C = 1/40$ *and* $C = 1/800$*, respectively (dotted lines).*

6.3. Convergence properties for larger ε. As shown in [38, 40, 22], the convergence of a PDE approximation can be expressed in terms of the approximation properties of the interpolant (consistency error) and a stability term. The consistency error of the PDE operator can for example be expressed as

$$\mathcal{E}_\mathcal{L} = \mathcal{L}(I_h(u) - u),$$

where $I_h(u)$ interpolates u using a node set with fill distance h. Several authors have derived exponentially converging error results for RBF interpolation [33, 28, 26, 3, 44, 34]. The first papers are focused on interpolation errors, while [34] also provides estimates for derivatives of functions with many zeros, such as the interpolation error. We use the optimality property of RBF interpolants in the native space (reproducing kernel Hilbert space) [7]

$$\|I_h(u)\|_{\mathcal{N}(\Omega)} \le \|u\|_{\mathcal{N}(\Omega)},$$

to replace the interpolation error norm with the function norm, since $\|I_h(u)-u\|_{\mathcal{N}(\Omega)} = \|\mathcal{E}_\mathcal{I}\|_{\mathcal{N}(\Omega)} \le 2\|u\|_{\mathcal{N}(\Omega)}$. We get the following estimates for RBF interpolants in compact cube domains using [34, Corollary 5.1] for Gaussians

$$\|\mathcal{E}_\mathcal{I}\|_\infty \le e^{C_G \log(h)/h}\|\mathcal{E}_\mathcal{I}\|_{\mathcal{N}_\mathcal{G}(\Omega)} \le 2e^{C_G \log(h)/h}\|u\|_{\mathcal{N}_\mathcal{G}(\Omega)},$$

where $C_G > 0$, and for inverse multiquadrics

$$\|\mathcal{E}_\mathcal{I}\|_\infty \le e^{-C_Q/h}\|\mathcal{E}_\mathcal{I}\|_{\mathcal{N}_\mathcal{Q}(\Omega)} \le 2e^{-C_Q/h}\|u\|_{\mathcal{N}_\mathcal{Q}(\Omega)},$$

when $h \le h_0$, and with $C_Q > 0$. This is the same h_0 as in the sampling inequality (6.16), which means that the condition is restrictive. The constants C_G and C_Q depend on the number of dimensions d and properties of the domain Ω.

The results for derivatives of the interpolation error are given for Lipschitz domains, which are more general than compact cubes, but the results are instead weaker in terms of the convergence rate. From [34, Theorem 3.5], we get

$$\|\mathcal{E}_\mathcal{L}\|_\infty \le 2e^{\tilde{C}_G \log(h)/\sqrt{h}}\|u\|_{\mathcal{N}_\mathcal{G}(\Omega)},$$

$$\|\mathcal{E}_{\mathcal{L}}\|_\infty \le 2e^{-\tilde{C}_Q/\sqrt{h}}\|u\|_{\mathcal{N}_{\mathcal{Q}}(\Omega)},$$

for Gaussians and inverse multiquadrics respectively. The higher rate of Gaussian RBFs is related to the behavior of embedding constants for the native space in relation to Sobolev spaces of increasing order. The constants $\tilde{C}_G$ and $\tilde{C}_Q$ depend on properties of the domain Ω, and $\tilde{C}_Q$ also depends on $\mathcal{L}$ and d. In [35], it is shown that the better convergence rates are obtained also for derivatives of the interpolation error if the nodes are clustered in a layer close to the boundary.

In order to investigate numerically what the actual behavior of the error is for the Helmholtz problem, we solve the one-dimensional problem for a range of shape parameter values and different numbers of node points. In this test, we have used multiquadric RBFs. We assume that the error for multiquadric RBFs has the form

$$\|e\|_\infty = A_M \exp(-C_M f(h)),$$

where $C_M > 0$, $f(h) = 1/h$ or $f(h) = 1/\sqrt{h}$, and the native space norm has been absorbed into the constant. Then a plot of the logarithm of the error against $f(h)$ should result in a straight line. From Figure 6.2, it is clear that $f(h) = 1/h$ is a better fit. The dashed lines correspond to a fit of the model with $f(h) = 1/h$ to the actual errors, where the results suffering from ill-conditioning effects have been ignored.

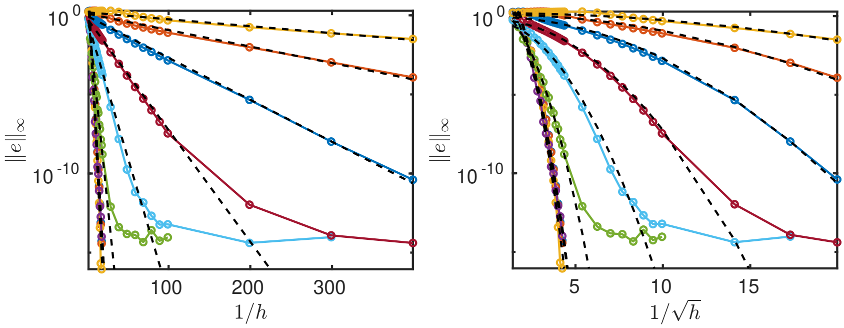


FIG. 6.2. *The error in the one-dimensional Helmholtz solution when multiquadric RBFs are used as a function of* $1/h$ *(left) and* $1/\sqrt{h}$ *(right) for shape parameters* $\varepsilon = 10^{-2+\frac{4}{9}q}$, $q = 1, \dots, 9$ *(left to right). The dashed black lines/curves correspond to a fit of* $\|e\|_\infty = A_M \exp(-C_M/h)$ *to the error data (in both cases).*

Figure 6.3 shows the fitted model parameters A_M and C_M for different shape parameter values. The different curves correspond to different wavenumbers, and it should be noted that the exponential rate C_M becomes independent of the wave number when $\varepsilon \gtrapprox 0.5$. The rate also decreases with increasing shape parameter values. The optimal rate is attained for a small positive shape parameter value, and for even smaller shape parameters, the asymptotic (polynomial) rate is dominating. The coefficient A_M instead seems to be largest where the rate is optimal, and smallest where the rate is lowest, which makes it harder to determine the best shape parameter value. We discuss this further in the following subsection.

6.4. Convergence as a function of the shape parameter. Dependence on the shape parameter is not discussed in [34], and the results reported in the previous subsection hold for a fixed value of ε. However, using a shape parameter $\varepsilon_0 \neq 1$ for an

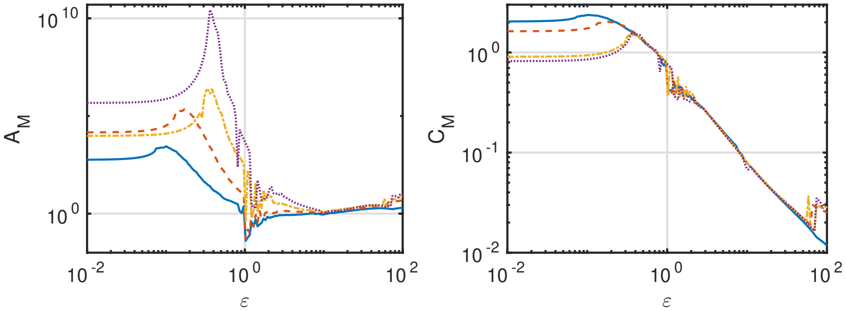


Fig. 6.3. *The result of fitting the model parameters A_M and C_M to the computed errors for the one-dimensional Helmholtz problem using multiquadric RBFs and different values of the shape parameter ε, and for $\kappa = \pi$ (solid line), $\kappa = 2\pi$ (dashed line), $\kappa = 4\pi$ (dash-dot line), and $\kappa = 6\pi$ (dotted line).*

interpolation problem defined in the domain Ω with fill distance h is equivalent to using a shape parameter $\varepsilon_1 = 1$ for a problem in the scaled domain $\varepsilon_0\Omega$ with fill distance $\varepsilon_0 h$. This can be understood by noting that $\phi(\varepsilon_0\|\underline{x}_i - \underline{x}_j\|) = \phi(1 \cdot \|\varepsilon_0\underline{x}_i - \varepsilon_0\underline{x}_j\|)$. Hence, the native space norm is the same in both cases, and the errors are the same in both cases.

If we let the constants A_M and C_M in the error estimate for a specific domain Ω and shape parameter ε be denoted by $A_M(\Omega, \varepsilon)$ and $C_M(\Omega, \varepsilon)$, this means that

$$A_M(\Omega, \varepsilon)e^{-C_M(\Omega,\varepsilon)/h} = A_M(\varepsilon\Omega, 1)e^{-C_M(\varepsilon\Omega,1)/(\varepsilon h)}. \tag{6.17}$$

That is, the convergence rate for a fixed value of ε is increasing for smaller values of ε. This can also be seen in Figure 6.3, where the slope in the logarithmic plot of C_M against ε is approximately equal to -1. It should be stressed that this does not hold in the flat limit regime, only for $\varepsilon \gtrapprox 0.5$ (in our case). This also corresponds to the theoretical result given in [26], where an explicit constraint on the smallest shape parameter for which the results hold is given as $\varepsilon \geq 1/D$, where for a cube domain, D is the side. This coincides well with the numerical results. However, there is also an upper bound $\varepsilon \leq 1$, which is harder to reconciliate with what we observe.

Figures 6.4 and 6.5 show the error as a function of ε for two one-dimensional problems, and one two-dimensional problem, respectively. The error curves represent a common behavior for smooth solution functions. Starting from a large shape parameter and moving towards smaller values, the error first decreases rapidly then reaches an optimal region, and finally levels out at the polynomial approximation error, see [20] for a more detailed discussion about the error curve and the optimal shape parameter.

Due to the conditioning problems for decreasing values of ε and increasing values of N, a common approach in the literature is to scale the shape parameter such that, e.g., $\varepsilon h = C$, which is called stationary interpolation. A problem is that stationary interpolation does not converge as h goes to zero. This can be understood by again looking at analogous problems. I we start from a problem on the domain Ω with shape parameter ε and fill distance h, and we refine to get fill distance h/q and shape parameter $q\varepsilon$, then the equivalent problem is $(q\Omega, \varepsilon, h)$. That is, the refinement corresponds to stretching out the domain, while keeping the fill distance and shape parameter constant. This makes the apparent solution function become increasingly smooth, and approaching a constant. Since constants are only reproduced for $\varepsilon = 0$ for the commonly used infinitely smooth RBFs, there is no convergence for a fixed non-zero ε. By augmenting the RBF approximation with polynomial terms, convergence corresponding to the polynomial order can be recovered also in the stationary case [9].

The convergence curves when choosing the shape parameter as $\varepsilon = Ch^{\beta}$ for different exponents β are also shown as dashed lines in Figures 6.4 and 6.5. As expected, the stationary choice, $\beta = -1$ levels out as N increases. For $\beta > -1$ we get convergence along different paths. The choice $\beta = 0$ corresponds to the exponential convergence case for fixed shape parameter values. For these Helmholtz problems, the curve with $\varepsilon = Ch^{3/2}$ captures the optimal shape parameter values well. For other types of problems, the relation would look different.

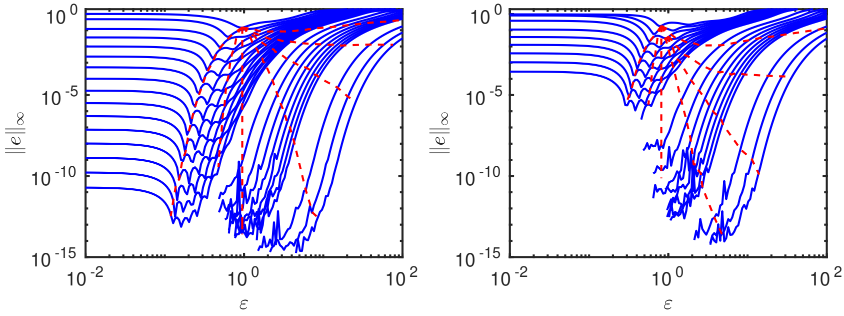


FIG. 6.4. *The maximum error as a function of ε for $\kappa = 2\pi$ (left) and $\kappa = 4\pi$ (right) using multiquadric RBFs. The number of node points is from top to bottom $N = 6, 7, \ldots, 21, 30, 40, \ldots, 100, 200, 300, 400$ in the left subfigure, and $N = 10, 11, \ldots, 20, 30, \ldots, 100, 200, 300, 400$ in the right subfigure. The dashed lines show how the error curves are traversed if the shape parameter is chosen as $\varepsilon = Ch^{\beta}$, with $\beta = \frac{3}{2}, \frac{1}{2}, 0, -\frac{1}{2}, -\frac{3}{4}, -1, -\frac{3}{2}$ from left to right.*

For the two-dimensional problem, the curves are more irregular due to several interacting terms in the error [20]. However, the overall behavior for the different ways to choose the shape parameter is very similar to the one-dimensional case.

Assuming that $C_M(\varepsilon\Omega, 1)$ in (6.17) does not vary strongly with ε, something that can be verified by noting that the slope the line in Figure 6.3 for C_M is approximately equal to -1, we can finally provide a convergence rate for the scaled ε convergence case. If we have exponential convergence as $1/\varepsilon h$ and $\varepsilon = Ch^{\beta}$ we end up with

$$\|e\|_\infty = A_M^{\varepsilon} e^{-C_M^{\varepsilon}/h^{\beta+1}}, \quad -1 < \beta \leq 0, \tag{6.18}$$

where $C_M^{\varepsilon} > 0$ and the superscript indicates the potential ε-dependence. If $\beta > 0$, the convergence curves may enter the polynomial region, and we cannot in general get increasing convergence rates for increasing β. The validity of this is expression is further investigated numerically in Section 7.

7. Numerical experiments. In this section, we focus on the third test problem with curved boundaries, see Figure 2.1. We look at how to choose the method parameters and how we can use the theoretical estimates to interpret the results. Unless otherwise mentioned, the problem parameters are given by wavenumber $\kappa = 6\pi$,

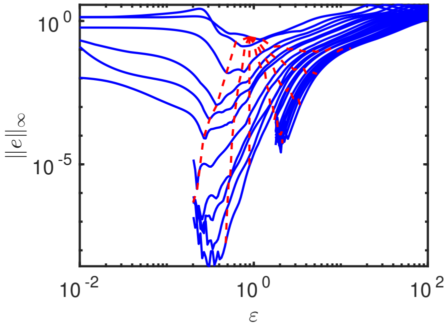


FIG. 6.5. *The maximum error as a function of ε for $\kappa = 2.2\pi$ for the two-dimensional problem using multiquadric RBFs. The number of node points is from top to bottom $N \approx n^2$, for $n = 3, \dots, 25$. The dashed lines show how the error curves are traversed if the shape parameter is chosen as $\varepsilon = Ch^\beta$, with $\beta = \frac{3}{2}, \frac{1}{2}, 0, -\frac{1}{2}, -\frac{3}{4}, -1, -\frac{3}{2}$ from left to right.*

source location $x_s = 0.3$, and boundary curves

$$\begin{aligned}
\gamma_1 &= 0.3\exp(-20(x_2-0.5)^2),\\
\gamma_2 &= 0.8 - 0.3\left(\exp(-80(x_2-0.3)^2) + \exp(-80(x_2-0.7)^2)\right).
\end{aligned}$$

For global RBF approximations and shape parameters that are not in the flat limit a uniform node spacing is in general recommended [31]. However, when the problem size is large enough, there can instead be problems at the boundaries unless the nodes are clustered towards the boundaries [32, 11]. In our experiments, we do not reach the regime where this is an issue. Therefore, we use quasi uniform nodes. The nodes are constructed from the input parameters n_1 and n_2, that specify the number of nodes in the vertical direction at the left boundary, and the number of nodes in the horizontal direction. We define the step sizes $h_1 = 0.8/(n_1 - 1)$ and $h_2 = 1/(n_2 - 1)$. Based on these step sizes, the nodes are then placed uniformly along vertical lines with as similar node distance as possible. The nodes at the top and bottom boundaries are placed with uniform arc length. If the nodes are too regular, they are not unisolvent, and the conditioning gets higher at least for shape parameters that are small [20]. Therefore, we add a random perturbation to each node. In all experiments performed here, the size of the random perturbation is $0.25(h_1, h_2)$ for the interior nodes, while boundary nodes are only perturbed along the boundary. The solution, residual, and errors are evaluated on a grid. An example of both nodes and evaluation grid is given in Figure 7.1. The resulting numbers of node points for the grids we have used in the experiments are shown in Table 7.1.

Errors are measured against a reference solution computed using the largest node set with $n_1 \times n_2 = 100 \times 125$. This is the largest problem size that fits in the memory of the Dell Latitude E6230 laptop with an i5-3360 dual core CPU running at 2.8 GHz that was used for the experiments. When we refer to the maximum norm of the numerical errors or the solution, we evaluate them on the 60×60 evaluation grid, except for the solutions with higher wavenumbers, where we use 100×100 grid points. We use multiquadric RBFs in all numerical experiments. The MATLAB implementations of the solvers that were used in the experiments are available at the first authors software page.

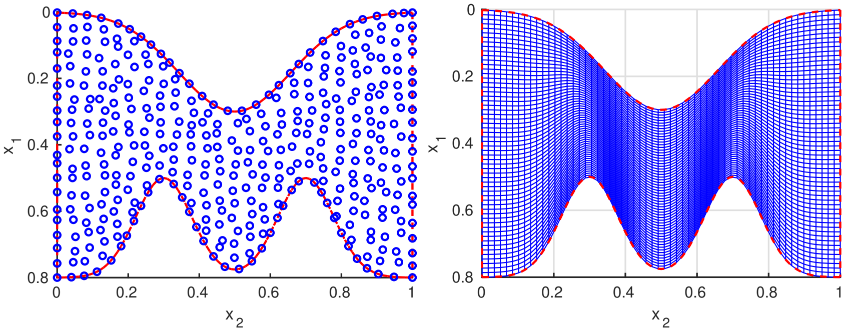


FIG. 7.1. *Node points with $n_1 = 20$ and $n_2 = 25$ (left) and the evaluation grid with 60×60 points used for the convergence experiments (right).*

TABLE 7.1
The size N of the different node sets that are used in the experiments. The parameters n_1 and n_2 are chosen to make h_1 and h_2 as equal as possible.

$n_1 \times n_2$	10×12	11×14	12×15	13×16	14×17	15×19	16×20
N	104	131	152	174	194	235	261

$n_1 \times n_2$	17×21	18×22	19×24	20×25	22×27	24×30	26×32
N	287	317	362	396	462	563	639

$n_1 \times n_2$	28×35	30×37	32×40	34×42	36×45	38×47	40×50
N	747	844	971	1079	1219	1341	1493

$n_1 \times n_2$	50×62	60×75	70×87	80×100	90×112	100×125	
N	2294	3306	4434	5813	7300	9029	

7.1. Selecting a tolerance for constructing the DtN boundary conditions. As mentioned in Section 3.1, we need to compute N inner products with each vertical eigenmode ψ_m present in the problem at the two vertical boundaries. Accurate numerical computation of these integrals is a significant computational cost, e.g, up to $n_f = 1700$ function evaluations per integral are needed for tolerance $1e - 15$. The question is which tolerance to choose.

The sensitivity of the problem (ill-conditioning) depends strongly on the shape parameter ε with an exponentially increasing condition number as the shape parameter goes to zero. By using a stable evaluation method such as RBF-QR for Gaussian RBFs, the sensitivity is removed and the tolerance for the integrals does not need to be smaller than the desired error in the solution. However, for the test problem considered here, too small values of ε, leading to a global polynomial approximation is not an appropriate choice, and we are not able to use RBF-QR.

Table 7.2 shows the average number of function evaluations needed by MATLAB's `quadl` to approximate one integral to a prescribed absolute tolerance for different values of the shape parameter ε for a node set with $n_1 \times n_2 = 30 \times 38$. The bold faced entries in the table show the largest tolerance that can be used before the approximation changes significantly. The tolerance is much smaller than the absolute error in the solution, which is about 0.5 compared with the reference solution. The condition numbers computed by MATLAB are between $1 \cdot 10^{17}$ for $\varepsilon = 5$ and $1 \cdot 10^{11}$ for $\varepsilon = 12$. For larger N, the ill-conditioning also increases, so we expect that even smaller tolerances are needed in this case.

Table 7.2

The average number of function evaluations for approximating one integral of the type in (2.9) using `quadl` *for multiquadric RBFs. Bold faced numbers show the largest tolerance that does not significantly alter the result. The relative error against the reference solution is also given. A* × *indicates that the approximation had an error of the same order as the size of the solution.*

Tolerance	1e−4		1e−6		1e−8		1e−10	
$\varepsilon = 5$	33	×	52	×	97	×	**164**	2.5e−1
$\varepsilon = 6$	34	×	54	×	**102**	1.4e−1	168	1.4e−1
$\varepsilon = 7$	35	×	56	6.3e−1	**106**	6.1e−2	173	5.9e−2
$\varepsilon = 8$	36	×	58	6.6e−2	**109**	5.2e−2	180	5.2e−2
$\varepsilon = 9$	36	×	**59**	6.0e−2	112	5.0e−2	187	5.0e−2
$\varepsilon = 10$	36	×	**60**	5.0e−2	114	5.0e−2	193	5.0e−2
$\varepsilon = 11$	37	3.5e−1	**62**	5.1e−2	115	5.1e−2	198	5.1e−2
$\varepsilon = 12$	**37**	7.9e−2	63	5.2e−2	117	5.2e−2	202	5.2e−2

7.2. Choosing the starting value for the shape parameter. To solve a large scale problem efficiently it pays off to choose the shape parameter carefully, since it does not affect the cost, only the accuracy. As was discussed in Section 6.4, a practical way to achieve convergence in spite of the ill-conditioning is to choose the shape parameter as $\varepsilon = Ch^{\beta}$, with $\beta > -1$. We are going to use $\beta = -1/2$, which provides a trade-off between convergence rate and conditioning problems. Then we need to decide which C to use.

Compared with the full solution, it is not so expensive to solve a much less resolved problem a few times for different shape parameters. We want to test if the residual-based error estimate (6.13) can help us find the best shape parameter value for such a problem, and from there the C to use. We also try the ℓ_2-norm of the residual as an indicator, since the residual should be small when the error is small. The maximum norm of the residual was also tested, but did not correlate strongly with the error. Figure 7.2 shows the relative error estimate as well as the relative ℓ_2-norm of the residual together with the actual error against the reference solution. In the first example, the shape parameter values corresponding to the smallest error estimate, ε^{est}, and the smallest residual norm, ε^{res}, are both close to the actual minimum ε^{*}. In the second example, the minimum for the error estimate is a bit higher than the true value.

Table 7.3 gives the minimal shape parameter values for ten different (small to medium) problem sizes. In most of the cases the error estimate, the residual estimate, or both are close to the true value. We have also computed the C-values corresponding to the average of the two estimates. If we had solved only the first problem, we would have chosen $C = 1.5$. This is what we have used for the convergence experiments in the following subsection. We also tried $C = 1$, but then the ill-conditioning prevented us from solving the largest problems.

An alternative method to find a good shape parameter value is to use the leave-one-out cross validation method. It was first introduced for RBF interpolation methods [36], and a cost effective version of the method was derived in [45]. It was suggested to use LOOCV for PDE problems using the residual as error indicator in [4], and this

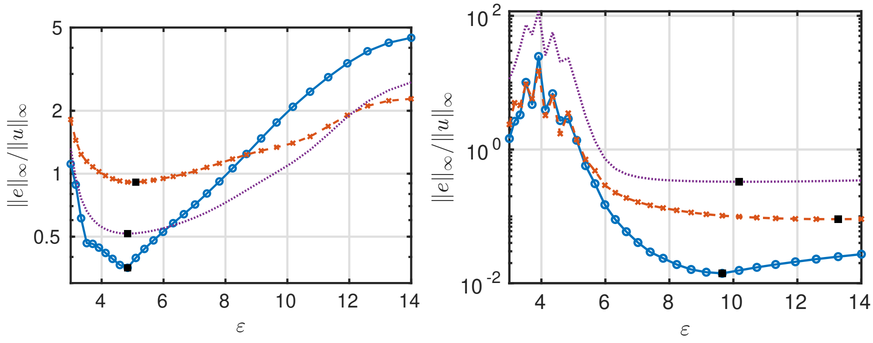


Fig. 7.2. *The error estimate* (6.13) *(×), the ℓ_2-norm of the residual (dotted line) and the error against a highly resolved reference solution (○) for the* 10×12 *(left) and* 40×50 *(right) node sets. The minima are indicated by black squares.*

Table 7.3

The optimal shape parameter for the error against the reference solution ε^*, the error estimate ε^{est}, and the ℓ_2-norm of the residual ε^{res} and the constant $\tilde{C}$ implied by the average of the two estimates for different problem sizes.

$n_1 \times n_2$	10 × 12	11 × 14	12 × 15	13 × 16	14 × 17
ε^*	4.8	4.6	5.7	4.6	7.0
ε^{est}	5.1	6.7	8.7	6.7	3.2
ε^{res}	4.8	5.4	7.4	5.4	3.7
$\tilde{C}$	1.5	1.7	2.2	1.6	0.9
$n_1 \times n_2$	15 × 19	16 × 20	20 × 25	30 × 37	40 × 50
ε^*	7.4	4.8	9.7	9.2	9.7
ε^{est}	7.8	6.0	9.2	13.3	13.3
ε^{res}	6.3	5.4	7.0	9.7	10.2
$\tilde{C}$	1.7	1.3	1.7	1.9	1.7

was implemented in [8]. We tried to use residual-based LOOCV on the Helmholtz problems in this paper, but the preliminary results were not close enough to the optimal values, and we therefore decided to use the error approximation instead.

7.3. Convergence experiments. Here, we use the relation $\varepsilon = C/\sqrt{h} = 1.5/\sqrt{h}$ to run a convergence experiment. We solve the test problem for different problem sizes and compute the error estimate and the error against the reference solution. According to equation (6.18), with this choice of shape parameter scaling, the error should be of the form

$$\|e\|_\infty = A_M \exp(-C_M/\sqrt{h}).$$

In Figure 7.3, we plot the relative error and the relative error estimate (6.13) against $1/\sqrt{h}$. A line has been fitted to the data set, and it is clear from the picture that it is a good fit of the convergence trend. The slopes C_M are 0.78 for the error and 0.75 for the error estimate, which means that the error estimate gives very good results for the ratio of errors at different resolutions, even if the constant is not precise. The constant A_M is 3.0 times larger for the error estimate than for the error. Based on the curves in Figure 7.2, we expect A_M to be problem and/or parameter dependent.

If we compare the error reduction from the smallest to the largest problem size with what we would get with an algebraically converging method where the error is $\mathcal{O}(h^p)$, a reduction in error with a factor 242 for a step size reduction of 10 corresponds to $p = 2.4$. That is, even if we have exponential convergence, the overall error reduction is not that impressive. However, the small numbers of points we can use, while still getting reasonable results are impressive. The smallest problem has 12 points in the horizontal direction, which corresponds to 4 points per wavelength. A rule of thumb for a finite difference method is that at least 15 points per wavelength, that is 45 for this problem, are needed for geometric resolution. For the largest problems, the tolerance for the quadrature had to be lowered. The small perturbations introduced by the inexact quadrature with tolerance $1 \cdot 10^{-10}$ are enough to prevent the convergence curve from following the straight line, and the convergence rate then seems to decrease. These experiments were run using tolerance $1 \cdot 10^{-14}$.

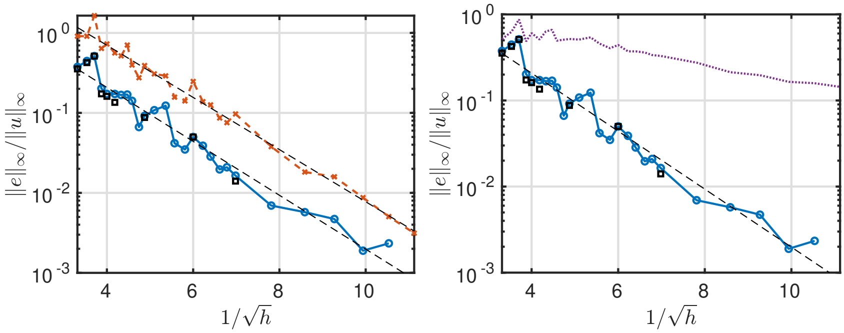


FIG. 7.3. *The relative error estimate* (6.13) *(×) and the error against the reference solution as a function of* $1/\sqrt{h}$ *are shown in the left subfigure. The* ℓ_2*-norm of the residual is shown together with the same error curve in the right subfigure. The black squares are the results for the optimal shape parameter values. The dashed lines represent lines fitted to the data points.*

In the right subfigure of Figure 7.3, the ℓ_2-norm of the residual is plotted together with the same relative error results. Even though the residual norm gives reasonable estimates for the optimal shape parameter, it is clear that we cannot use it to follow the error trend.

7.4. Experiments with larger wave numbers. We have also solved problems with larger wavenumbers as this usually is a challenge for wave propagation problems. For these problems $\kappa = 12\pi$ and 24π, corresponding to 6 and 12 wavelengths along the duct. The solution functions are shown in Figure 7.4.

These solutions have 9 and 19 propagating modes at the left boundary, respectively. A problem here was to compute the inner products with the eigenmodes to high enough accuracy. The accuracy of the boundary conditions is crucial to get the correct wave pattern. We were not able to run the simulations for $\kappa = 24\pi$ for a larger problem size than 50×62 (with good results). The same shape parameter scheme as for the convergence experiment was used.

For each problem, we ran three different problem sizes in order to get an estimate of the errors in the solutions. Then we computed the relative errors of the coarser solutions with respect to the finest solution. The computed errors are compared with the error estimate (6.13) to find the approximate ratio between real errors and estimate. Then we use the worst case ratio to project an error estimate also for

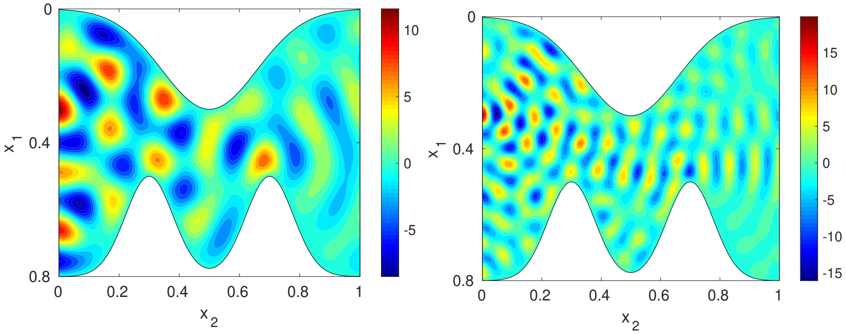


Fig. 7.4. *The solution function for $\kappa = 12\pi$ (left) and $\kappa = 24\pi$ (right). The solution is computed using nodes with $n_1 \times n_2 = 50 \times 62$. The source is located at the left boundary at $x_s = 0.3$.*

Table 7.4

The relative error in relation to the finest solution, the relative error estimate, the ratio, the adjusted error estimate, and the local slope for the problem with $\kappa = 12\pi$.

$n_1 \times n_2$	$\|e\|_\infty/\|u\|_\infty$	$\|\tilde{e}\|_\infty/\|u\|_\infty$	$\|\tilde{e}\|_\infty/\|e\|_\infty$	$\frac{\|\tilde{e}\|_\infty}{\|u\|_\infty} / \min \frac{\|\tilde{e}\|_\infty}{\|e\|_\infty}$	C_M
40 × 50	0.0435	0.3083	7.1	0.0435	–
50 × 62	0.0240	0.1781	7.4	0.0251	0.66
60 × 75	–	0.0699	–	0.0099	1.2

the finest solution. The results are shown in Tables 7.4 and 7.5, indicating around 1% error for $\kappa = 12\pi$ and around 12% error for $\kappa = 24\pi$. The numbers of points per wavelength are 75/6=12.5 and $62/12 \approx 5.2$, respectively. For the problem with $\kappa = 6\pi$ and the same two node sets, we had 0.6–0.7% error and 21–25 points per wavelength. If we look at the error for 5 points per wavelength for $\kappa = 6\pi$, it is around 20%. That is, it seems that we do not need to resolve more with increasing frequency. For finite difference and finite element methods the error in a waveguide Helmholtz problem is typically proportional to $h^p\kappa^{p+1}$ [1, 29, 19]. This effect diminishes as the order of the method increases, and for a spectral method it disappears. This is consistent with the results for small ε in Section 6.2, where the error approximations are proportional to $(\kappa h)^K$.

8. Discussion. The main benefits with using global RBF methods for solving Helmholtz-type problems are that very few points per wavelength are needed to obtain a qualitatively correct solution, and that the number of points per wavelength does not need to increase with κ (the number of wavelengths). It is also relevant that non-trivial waveguide geometries can be managed easily, since there is no need for an orthogonal or even a structured grid. In [29, 19], we used orthogonal grids, which limits how much the boundaries can vary. It should be mentioned that the DtN boundary conditions assume a smooth continuation with horizontal boundaries outside of the domain. In our example, the derivative of the boundary curves is non-zero at $x_2 = 0, 1$, which introduces an error. However, since we got optimal convergence rates in the experiments, these errors are not large enough to influence the results at the level of errors that we could reach.

The main challenge of using a global RBF method for a PDE problem is the computational cost. In Helmholtz applications it is of interest to solve problems that are

Table 7.5

The relative error in relation to the finest solution, the relative error estimate, the ratio, the adjusted error estimate, and the local slope for the problem with $\kappa = 24\pi$.

$n_1 \times n_2$	$\|e\|_\infty/\|u\|_\infty$	$\|\tilde{e}\|_\infty/\|u\|_\infty$	$\|\tilde{e}\|_\infty/\|e\|_\infty$	$\frac{\|\tilde{e}\|_\infty}{\|u\|_\infty} / \min \frac{\|\tilde{e}\|_\infty}{\|e\|_\infty}$	C_M
30 × 37	0.3842	1.4909	3.9	0.3842	–
40 × 50	0.1292	0.7941	6.1	0.2047	0.64
50 × 62	–	0.4756	–	0.1226	0.62

large in terms of wavelengths, and therefore require a certain resolution. With a dense linear system, both the storage requirements and the computational cost for a direct solver quickly become difficult to manage at least without using distributed computing. On top of that, the severe ill-conditioning of the linear systems makes them sensitive to numerical errors in the quadrature employed in DtN conditions as well as to rounding errors. An attractive alternative to using global RBF collocation methods is to use localized methods such as RBF-generated finite differences (RBF-FD) [10] and RBF partition of unity methods (RBF-PUM) [22]. In [41] it was shown that for an option pricing application, there was no significant difference in accuracy between the global method and RBF-PUM for a given problem size, while the computational cost is significantly lower for RBF-PUM due to sparsity of the linear systems.

We compared the non-symmetric and symmetric collocation approaches and found that the symmetric method, even though elegant, becomes cumbersome especially for non-trivial operators. The main benefit of the symmetric collocation is the guaranteed non-singularity of the interpolation matrix. However, for the non-symmetric method, singularity only occurred for wavenumbers that were physically uninteresting or for problems that were numerically unresolved. It seems reasonable that if the continuous problem is well-posed and the discrete problem is resolved enough to be close to the continuous problem, singularity is unlikely, see also [16, 40].

We have also investigated the error behavior as a function of N and ε from different perspectives. Some of this can be explained by the limit behavior. We studied this for interpolation in [20], but here we looked at what is different for PDE problems. If the node set is unisolvent and PDE unisolvent, the RBF approximant has the form $s(\underline{x}) = P_K(\underline{x}) + \varepsilon^2 P_{K+2}(\underline{x}) + \ldots$, where $P_K(\underline{x})$ is the unique polynomial solution of degree K to the PDE problem, and P_{K+2j} have zero PDE residual at the node points. When ε is small, $P_K(\underline{x}) - u(\underline{x})$ dominates the error. This is the flat region in the error as a function of ε, see Figures 6.4 and 6.5. Then as ε starts to grow, there may be an optimal ε-range where the additional terms improve on the polynomial error, but eventually, the ε-terms dominate the error, and the exponential convergence rate depends mainly on ε and not on the problem, see Figure 6.3.

A contribution that we think is novel and of practical interest is the discussion about convergence for scaled shape parameters. We provide arguments for why $\varepsilon = Ch^\beta$ should lead to a convergence rate of the form $e^{C_M/h^{\beta+1}}$, and show that this is what we also get numerically for $\beta = -1/2$.

Another practical contribution is that we have shown that given a reasonable error estimate, we can decide on a good choice for the shape parameter based on a small test problem. Then using a converging shape parameter strategy, we can solve the real problem, and also based on a comparison of error estimates and errors against the finest solution, we can get an improved error estimate for the solution of the most resolved problem.

Even though global collocation methods are not really practical for large scale problems, many of the things we have learned can be transferred also to localized methods, as these are based on 'local global collocation'.

Appendix A. Proof sketches. In order to save space and not repeat already published material, we do not give the full proof for Theorem 5.1 here, instead we give instructions how to carry out the proof using the machinery laid down in [20, pp. 122–127]. Because the RBF approximant in the PDE case has exactly the same form as the usual RBF interpolant, we get the exact same expansion [20, Eq. (28)] of the solution for small ε

$$s(\underline{x},\varepsilon)=\varepsilon^{-2K}(\varepsilon^{-2q}P_{-q}(\underline{x})+\cdots+\varepsilon^{2K}P_K(\underline{x})+\cdots). \tag{A.1}$$

What differs from the interpolation case is the conditions that the polynomials must fulfill. In the PDE case we have that

$$\begin{array}{ll} P_K & \text{satisfies the inhomogeneous PDE and} \\ & \text{boundary conditions at the } N \text{ node points,} \\ P_j, \quad j\neq K & \text{satisfy the homogeneous PDE and} \\ & \text{boundary conditions at the } N \text{ node points.} \end{array} \tag{A.2}$$

The proof of part (i) is completely analogous to the proofs of Theorems 4.1 and 4.2 in [20]. For part (iii), we follow the steps in the proof of Theorem 4.1. For simplicity, we first assume that the nullspace $n(\underline{x})$ of the matrix Q defined in (5.3) is of degree K. The steps are identical until the point were we are considering the conditions for P_{-q+K}. There are three possibilities

- If $q=0$, then the polynomial is P_K and must satisfy the PDE. However, since the matrix Q is singular, this can only happen in the (unlikely) case that the right hand side $\underline{f}$ is in the range of Q.
- If $q>0$ and P_{-q+K} is identically zero, then the moment vector $\underline{\sigma}_{-q}$ is zero, leading to $\underline{\lambda}_{-q}$, because of the non-singularity of P. This means that we could have omitted the $-q$ term in the expansion and we must have $q=0$. This is in conflict with the previous case.
- Then we must have $q>0$ and P_{-q+K} must contain a nullspace component $\alpha n(\underline{x})$. This means that we have at least one divergent term in the expansion of the solution.

If there is just a single nullspace component of degree K, and extending Q with an appropriate monomial of degree $K+1$ leads to $\text{rank}(Q)=N$, then at the next step looking at P_{-q+1+K} we get the two possibilities $\alpha=0$, which has been ruled out, or $P_{-q+1+K}=P_K$. Hence, we must have $q=1$ and divergence of order ε^2.

If the nullspace is of lower degree than K, we will also get divergence, but the negative power of ε could be higher.

The argument behind part (ii) is that we need to go to the polynomial P_{-q+M} before we have enough degrees of freedom to satisfy the discrete PDE problem. Therefore, the limit must have degree M. However, because Q is non-singular, all previous polynomials must be identically zero and accordingly there can be no divergence. Compare with the proofs of Theorems 4.2 and 4.3.

For part (iv) of the proof, we follow the proof of Theorem 4.3. The important difference is that the relation between the moments is determined by the nullspace of P, but the possible nullspace parts in the polynomials P_{-q+J} is determined by the nullspace of Q. In [20], we arrive at an equation $C^TB^{-1}C\underline{\alpha}=\underline{0}$. The corresponding equation here becomes

$$C^T B^{-1} D\underline{\alpha} = \underline{0}, \tag{A.3}$$

where C is of size $n \times m$ and D has dimensions $n \times p$. To be precise, at step J of the proof, m is the dimension of the J-degree part of the nullspace of P and p is the corresponding dimension for Q.

If $m = p$, the system (A.3) is square, but non-singularity cannot be guaranteed when C and D are different. If $m > p$, the system is over-determined and it is likely that the only solution is $\underline{\alpha} = 0$. If on the other hand, $m < p$ the system is under-determined, allowing for non-zero nullspace components in the expansion polynomials.

If $n(x)$ defines a nullspace component for P, then $p(\underline{x})n(\underline{x})$ defines a higher degree nullspace component using any polynomial $p(\underline{x})$. Therefore, the dimension m typically grows with J. However, there is no similar mechanism for the nullspace of Q (since $Ln(\underline{x}) = 0$ does not generally imply $L(p(\underline{x})n(\underline{x})) = 0$). Accordingly, the dimension p is likely to stay the same or decrease with J.

These facts taken together lead to the statements in part (iv). We use the formulation *likely*, since it should be theoretically possible to construct counter examples in both the convergent and the divergent case.

REFERENCES

[1] A. Bayliss, C. Goldstein, and E. Turkel, *The numerical solution of the Helmholtz equation for wave propagation problems in underwater acoustics*, Comput. Math. Applic., 11 (1985), pp. 655–665. Special Issue Computational Ocean Acoustics.

[2] L. Bos, *On certain configurations of points in $\mathbb{R}^n$ which are unisolvent for polynomial interpolation*, Journal of approximation theory, 64 (1991), pp. 271–280.

[3] M. Buhmann and N. Dyn, *Spectral convergence of multiquadric interpolation*, Proc. Edinburgh Math. Soc. (2), 36 (1993), pp. 319–333.

[4] A. H.-D. Cheng, M. A. Golberg, E. J. Kansa, and G. Zammito, *Exponential convergence and h-c multiquadric collocation method for partial differential equations*, Numer. Methods Partial Differential Equations, 19 (2003), pp. 571–594.

[5] T. A. Driscoll and B. Fornberg, *Interpolation in the limit of increasingly flat radial basis functions*, Comput. Math. Appl., 43 (2002), pp. 413–422. Radial basis functions and partial differential equations.

[6] G. Fasshauer, *Solving partial differential equations by collocation with radial basis functions*, in Surface Fitting and Multiresolution Methods, Volume 2 of the Proceedings of the 3rd International Conference on Curves and Surfaces, Chamonix-Mont-Blanc, A. LeMéhauté, C. Rabut, and L. Schumaker, eds., Nashville, TN, 1997, Vanderbilt University Press, pp. 131–138.

[7] G. E. Fasshauer, *Meshfree approximation methods with MATLAB*, vol. 6 of Interdisciplinary Mathematical Sciences, World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2007. With 1 CD-ROM (Windows, Macintosh and UNIX).

[8] A. J. M. Ferreira, C. M. C. Roque, R. M. N. Jorge, G. Fasshauer, and R. Batra, *Analysis of functionally graded plates by a robust meshless method*, J. Mech. Adv. Mater. Struct., 14 (2007), pp. 577–587.

[9] N. Flyer, B. Fornberg, V. Bayona, and G. A. Barnett, *On the role of polynomials in RBF-FD approximations: I. Interpolation and accuracy*, J. Comput. Phys., 321 (2016), pp. 21–38.

[10] B. Fornberg and N. Flyer, *A primer on radial basis functions with applications to the geosciences*, vol. 87 of CBMS-NSF Regional Conference Series in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2015.

[11] B. Fornberg, E. Larsson, and N. Flyer, *Stable computations with Gaussian radial basis functions*, SIAM J. Sci. Comput., 33 (2011), pp. 869–892.

[12] B. Fornberg, E. Lehto, and C. Powell, *Stable calculation of Gaussian-based RBF-FD stencils*, Comput. Math. Appl., 65 (2013), pp. 627–637.

[13] B. Fornberg and C. Piret, *A stable algorithm for flat radial basis functions on a sphere*, SIAM J. Sci. Comput., 30 (2007), pp. 60–80.

[14] C. Franke and R. Schaback, *Solving partial differential equations by collocation using radial basis functions*, Appl. Math. Comput., 93 (1998), pp. 73–82.

[15] G. H. Golub and J. M. Ortega, *Scientific computing and differential equations*, Academic Press, Inc., Boston, MA, 1992. An introduction to numerical methods.

[16] Y. C. Hon and R. Schaback, *On unsymmetric collocation by radial basis functions*, Appl. Math. Comput., 119 (2001), pp. 177–186.

[17] E. J. Kansa, *Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics. II. Solutions to parabolic, hyperbolic and elliptic partial differential equations*, Comput. Math. Appl., 19 (1990), pp. 147–161.

[18] J. B. Keller and D. Givoli, *Exact non-reflecting boundary conditions*, J. Comp. Phys., 82 (1989), pp. 172–192.

[19] E. Larsson, *A domain decomposition method for the Helmholtz equation in a multilayer domain*, SIAM J. Sci. Comp., 20 (1999), pp. 1713–1731.

[20] E. Larsson and B. Fornberg, *Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions*, Comput. Math. Appl., 49 (2005), pp. 103–130.

[21] E. Larsson, E. Lehto, A. Heryudono, and B. Fornberg, *Stable computation of differentiation matrices and scattered node stencils based on Gaussian radial basis functions*, SIAM J. Sci. Comput., 35 (2013), pp. A2096–A2119.

[22] E. Larsson, V. Shcherbakov, and A. Heryudono, *A least squares radial basis function partition of unity method for solving PDEs*, SIAM J. Sci. Comput., 39 (2017), pp. A2538–A2563.

[23] Y. J. Lee, C. A. Micchelli, and J. Yoon, *On convergence of flat multivariate interpolation by translation kernels with finite smoothness*, Constr. Approx., 40 (2014), pp. 37–60.

[24] ———, *A study on multivariate interpolation by increasingly flat kernel functions*, J. Math. Anal. Appl., 427 (2015), pp. 74–87.

[25] Y. J. Lee, G. J. Yoon, and J. Yoon, *Convergence of increasingly flat radial basis interpolants to polynomial interpolants*, SIAM J. Math. Anal., 39 (2007), pp. 537–553.

[26] W. R. Madych, *Miscellaneous error bounds for multiquadric and related interpolators*, Comput. Math. Appl., 24 (1992), pp. 121–138. Advances in the theory and applications of radial basis functions.

[27] ———, *An estimate for multivariate interpolation. II*, J. Approx. Theory, 142 (2006), pp. 116–128.

[28] W. R. Madych and S. A. Nelson, *Bounds on multivariate polynomials and exponential error estimates for multiquadric interpolation*, J. Approx. Theory, 70 (1992), pp. 94–114.

[29] K. Otto and E. Larsson, *Iterative solution of the Helmholtz equation by a second-order method*, SIAM J. Matrix. Anal. Appl., 21 (1999), pp. 209–229.

[30] U. Pettersson, *Radial basis function approximations for the Helmholtz equation*, M.Sc. thesis, UPTEC Report F 03 082, School of Engineering, Uppsala Univ., Uppsala, Sweden, 2003.

[31] R. B. Platte and T. A. Driscoll, *Polynomials and potential theory for Gaussian radial basis function interpolation*, SIAM J. Numer. Anal., 43 (2005), pp. 750–766.

[32] R. B. Platte, L. N. Trefethen, and A. B. J. Kuijlaars, *Impossibility of fast stable approximation of analytic functions from equispaced samples*, SIAM Rev., 53 (2011), pp. 308–318.

[33] M. J. D. Powell, *Univariate multiquadric interpolation: some recent results*, in Curves and surfaces (Chamonix-Mont-Blanc, 1990), Academic Press, Boston, MA, 1991, pp. 371–382.

[34] C. Rieger and B. Zwicknagl, *Sampling inequalities for infinitely smooth functions, with applications to interpolation and machine learning*, Adv. Comput. Math., 32 (2010), pp. 103–129.

[35] ———, *Improved exponential convergence rates by oversampling near the boundary*, Constr. Approx., 39 (2014), pp. 323–341.

[36] S. Rippa, *An algorithm for selecting a good value for the parameter c in radial basis function interpolation*, Adv. Comput. Math., 11 (1999), pp. 193–210. Radial basis functions and their applications.

[37] R. Schaback, *Multivariate interpolation by polynomials and radial basis functions*, Constr. Approx., 21 (2005), pp. 293–317.

[38] ———, *Convergence of unsymmetric kernel-based meshless collocation methods*, SIAM J. Numer. Anal., 45 (2007), pp. 333–351.

[39] ———, *Limit problems for interpolation by analytic radial basis functions*, J. Comput. Appl. Math., 212 (2008), pp. 127–149.

[40] ———, *All well-posed problems have uniformly stable and convergent discretizations*, Numer.

Math., 132 (2016), pp. 597–630.

[41] V. Shcherbakov and E. Larsson, *Radial basis function partition of unity methods for pricing vanilla basket options*, Comput. Math. Appl., 71 (2016), pp. 185–200.

[42] G. Song, J. Riddle, G. E. Fasshauer, and F. J. Hickernell, *Multivariate interpolation with increasingly flat radial basis functions of finite smoothness*, Adv. Comput. Math., 36 (2012), pp. 485–501.

[43] Z. M. Wu, *Hermite-Birkhoff interpolation of scattered data by radial basis functions*, Approx. Theory Appl., 8 (1992), pp. 1–10.

[44] Z. M. Wu and R. Schaback, *Local error estimates for radial basis function interpolation of scattered data*, IMA J. Numer. Anal., 13 (1993), pp. 13–27.

[45] F. Yang, L. Yan, and L. Ling, *Doubly stochastic radial basis function methods*, J. Comput. Phys., 363 (2018), pp. 87–97.

Recent licentiate theses from the Department of Information Technology

2020-001	Huu-Phuc Vo: *Towards Machine-Assisted Reformulation for MiniZinc*
2019-007	Carl Andersson: *Deep Learning Applied to System Identification: A Probabilistic Approach*
2019-006	Kristiina Ausmees: *Efficient Computational Methods for Applications in Genomics*
2019-005	Carl Jidling: *Tailoring Gaussian Processes for Tomographic Reconstruction*
2019-004	Amin Kaveh: *Local Measures for Probabilistic Networks*
2019-003	Viktor Bro: *Volterra Modeling of the Human Smooth Pursuit System in Health and Disease*
2019-002	Anton G. Artemov: *Inverse Factorization in Electronic Structure Theory: Analysis and Parallelization*
2019-001	Diane Golay: *An Invisible Burden: An Experience-Based Approach to Nurses' Daily Work Life with Healthcare Information Technology*
2018-004	Charalampos Orfanidis: *Robustness in Low Power Wide Area Networks*
2018-003	Fredrik Olsson: *Modeling and Assessment of Human Balance and Movement Disorders Using Inertial Sensors*
2018-002	Tatiana Chistiakova: *Ammonium Based Aeration Control in Wastewater Treatment Plants - Modelling and Controller Design*
2018-001	Kim-Anh Tran: *Static Instruction Scheduling for High Performance on Energy-Efficient Processors*

UPPSALA
UNIVERSITET

Department of Information Technology, Uppsala University, Sweden