Multiquadric Radial Basis Function Approximation Methods for the Numerical Solution of Partial Differential Equations

Scott A. Sarra
Marshall University
and
Edward J. Kansa
University of California, Davis

June 30, 2009

Preface

Radial Basis Function (RBF) methods have become the primary tool for interpolating multidimensional scattered data. RBF methods also have become important tools for solving Partial Differential Equations (PDEs) in complexly shaped domains.

Classical methods for the numerical solution of PDEs (finite difference, finite element, finite volume, and pseudospectral methods) are based on polynomial interpolation. Local polynomial based methods (finite difference, finite element, and finite volume) are limited by their algebraic convergence rates. Numerical studies, such as the comparison of the MQ collocation method with the finite element method in [136], have been done that illustrate the superior accuracy of the MQ method when compared to local polynomial methods. Global polynomial methods, such as spectral methods, have exponential convergence rates but are limited by being tied to a fixed grid. RBF methods are not tied to a grid and in turn belong to a category of methods called ***meshless methods***. The large number of recent books, which include [4, 3, 42, 71, 92, 93, 94, 95, 133, 149, 150, 196], on meshfree methods illustrates the popularity that the methods have recently enjoyed. The global, non-polynomial, RBF methods may be successfully applied to achieve exponential accuracy where traditional methods either have difficulties or fail. An example is in multidimensional problems in non-rectangular domains. RBF methods succeed in very general settings by composing a univariate function with the Euclidean norm which turns a multidimensional problem into one that is virtually one dimensional.

RBF methods are a generalization of the Multiquadric (MQ) RBF method which utilizes one particular RBF. The MQ RBF method has a rich history of theoretical development and applications. The subject of this monograph is the MQ RBF approximation method with a particular emphasis on using the method to numerically solve partial differential equations. This monograph differs from other recent books [31, 63, 179, 209] on meshless methods in that it focuses only on the MQ RBF while others have focused on meshless methods in general. It is hoped that this refined focus will result in a clear and concise exposition of the area.

Matlab code that illustrates key ideas about the implementation of the MQ method has been included in the text of the manuscript. The included code, as well as additional Matlab code used to produce many of the numerical examples, can be found on the web at `http://www.scottsarra.org/math/math.html`

Contents

Listings

Chapter 1

Development and Overview of the MQ RBF Method

The Multiquadric (MQ) Radial Basis Function (RBF) interpolation method was developed in 1968 by Iowa State University Geodesist Roland Hardy who described and named the method in a paper [97] that appeared in 1971. Hardy's discovery was motivated by a problem from cartography which he described as [97]

> ... given a set of discrete data on a topographic surface, reduce it to a satisfactory continuous function representing the topic surface ...

Hardy's early attempts at solving the problem involved the use of Trigonometric and Algebraic interpolation methods, both of which were found to be unsatisfactory. This is not surprising as it has been known since the mid 1950's [162] that interpolation in more than one dimension with basis functions ψ_j that are independent of data locations is not a well-posed problem. There are an infinite number of data locations that lead to a problem with no solution. Hardy bypassed this problem with a new approach in which the interpolant is constructed from linear combinations of a single basis function that is radially symmetric about its center, $x \in \mathcal{R}^d$, and whose argument, $r = \| x \|_2$, is dependent on the node locations. The basis functions used by Hardy were the quadric surfaces

$$\phi(r; c) = \sqrt{c^2 + r^2} \tag{1.1}$$

where c is a ***shape parameter*** that affects the shape of the surface. The RBF (1.1) is called the ***multiquadric*** or MQ RBF. The plot of the MQ in figure 1.1 reveals the radial symmetry of the basis functions. In reference [99], Hardy reviews the development of the MQ RBF method over the time period 1968 to 1988.

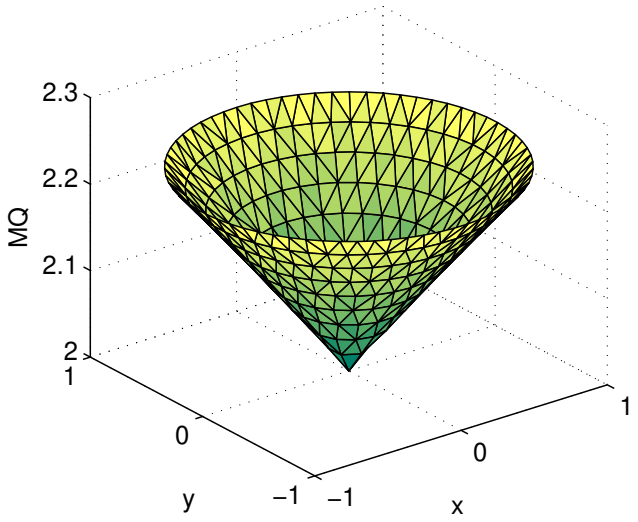

Figure 1.1: A MQ RBF (1.1) with shape parameter $c = 2$ on the unit circle.

Hardy's MQ interpolation scheme went largely unnoticed until 1979. Then in, a study [84] done at the Naval Postgraduate School in 1979 and summarized in [85], Mathematician Richard Franke compared various methods to solve the scattered data interpolation problem. He concluded that Hardy's MQ method was the best. Additionally, he conjectured that the system matrix of the method was invertible and that the method was well-posed. Franke's numerical results indicated that the method deserved attention and exposed the method to a broader audience. However, a theoretical underpinning was still absent. Then in 1986, the theory of the MQ method began to develop when Charles Micchelli, a Mathematician with IBM, proved that the system matrix for the MQ (as well as for many other RBFs) method was invertible [166]. Results [156] on the spectral convergence rate of MQ interpolation followed from Madych and Nelson in 1992. The first use of the MQ method to solve Differential Equations [117, 118] was by Physicist Edward Kansa in 1990. After the MQ method was first used to solve PDEs,

the popularity of the method continued to grow rapidly and a large number of applications of the method appeared.

Recent results ([52], [190], [132], [128]) have shown, what was at first a somewhat surprising result. As the parameter c goes to infinity, the MQ interpolation method is equivalent to global polynomial interpolation. Thus, if the MQ method is implemented on the structured grids that are required by global polynomial methods, they are equivalent to the polynomial based pseudospectral methods. In this way, the MQ method can be considered as a generalization of the pseudospectral methods to unstructured grids and complex domains.

1.1 An Example

Popular methods for the numerical solution of PDEs include finite difference, finite volume, and finite elements methods. The methods are based on local polynomial interpolation and have algebraic convergence rates. For instance, a local finite difference method with a five point stencil has a fourth-order algebraic convergence rate. The computational cost of approximating a derivative with a finite difference method is $\mathcal{O}(N)$.

Pseudospectral methods, based on global polynomial interpolation, are also widely popular due to their spectral convergence rates. In order to achieve spectral accuracy, global polynomial methods must be implemented on special collocations sites (or a smooth mapping of the grid). For example, a popular choice of collocation points (other Gaussian quadrature points may be used as well) are the Chebyshev-Gauss-Lobatto (CGL) quadrature points

$$x_j = -cos\left(\frac{\pi j}{N}\right), \qquad j = 0, 1, ..., N \tag{1.2}$$

on the interval $[-1, 1]$. The CGL points cluster densely around the endpoints of the interval. In higher dimensions, pseudospectral methods must be implemented on a tensor product grid that is based on the one-dimensional grid. Being tied to a fixed grid limits the applicability of pseudospectral methods to problems that have "nicely" shaped domains. The computational cost of evaluating a pseudospectral derivative is $\mathcal{O}(N^2)$ using matrix multiplication or $\mathcal{O}(N \log N)$ using fast transforms. More details on pseudospectral methods can be found in [100].

The non-polynomial MQ RBF method is theoretically spectral accurate (section 2.4) and is completely grid free. A RBF derivative can be evaluated by a matrix multiplication in $\mathcal{O}(N^2)$ flops. For large N, a MQ derivative can be evaluated to within accuracy ϵ_m in $\mathcal{O}\left(N \log N \left|\log \epsilon_m\right|^2\right)$ flops by fast multipole methods [10].

Next we give an example in order to illustrate some features of the MQ RBF methods and demonstrate how it compares to competing methods. The example approximates the derivative of function

$$f(x) = e^{\sin(\pi x)} \tag{1.3}$$

on the interval $[-1, 1]$ on grids of size ranging from $N = 10$ to $N = 60$ grid points and in increments of five. On each grid the derivative is approximated with the Chebyshev pseudospectral method, the MQ RBF method, and a fourth-order, five point, finite difference (FD) method. The finite difference method is implemented on a uniform grid as is typical (but not required) for low order finite difference methods. The Chebyshev pseudospectral (CPS) method is implemented on the CGL grid (1.2). The MQ RBF method is implemented on a grid that also clusters collocation points around the boundaries, but not as densely as the CGL grid. As N is increased, the shape parameter in the MQ method is adjusted so the the system matrix has a condition number in the range $5e15 \leq \kappa(B) \leq 5e16$ (an explanation for this is given in later sections).

The accuracy of each method at each N is illustrated in figure 1.2. Key aspects of this numerical example that are typical of the three methods are:

1. The FD method requires large N in order to be very accurate.

2. The MQ method is more accurate than the CPS method for small N ($N \leq 25$ in this case).

3. With larger N (actually its the smaller minimum distance between centers that is the cause, section 2.5), the MQ shape parameter must be adjusted so that the condition number of the associated linear system remains reasonable. The shape parameter adjustment decreases the accuracy of the method. If it is adjusted too much, the MQ method will have accuracy comparable to a local FD method (as it does in this example for $N = 60$.)

4. The CPS method is able to bypass solving ill-conditioned linear systems and there exist formulas for the elements of the CPS differentiation matrix as well as fast transforms to evaluate derivatives that avoid ill-conditioning and allow the CPS method to realize its theoretical spectral accuracy.

5. In problems with domains that are simply shaped (squares, rectangles, cubes, etc.) and that can accept the serve restriction on the location of collocation points, pseudospectral methods are usually the method of choice. However, in applied problems, domains are often not simply shaped and we may be unable to use structured grids. This is when RBF methods become the method of choice.

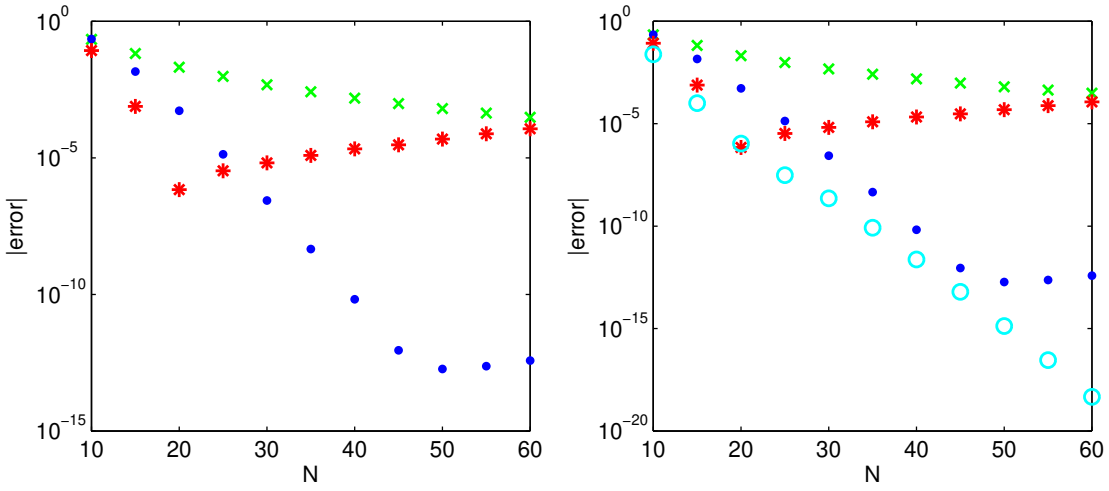

Figure 1.2: Left: The MQ method (red asterisk) versus Finite Difference (green x) and Chebyshev Pseudospectral methods (blue dot). Right: The MQ method computed with 80 digits of precision (cyan circles) has been added to the left image for comparison.

The MQ method can be thought of as a generalized, non-polynomial based, pseudospectral method. Theoretically, the MQ method has a spectral convergence rate and should have an error curve that continues down as the error plot for the CPS method does. The implementation of the MQ method in the above example is the most basic implementation possible. It does not feature any of the techniques discussed in the monograph that can be used

to improve upon the accuracy in this example. Throughout the monograph, progress in the development of the MQ method is discussed that has made the MQ method closer to being a true generalized pseudospectral method that can realize its theoretical spectral accuracy. In the right image of figure 1.2, the results of the MQ approximation being recomputed with extended precision (section 4.1) are shown. This time, the shape parameter is held constant at $\varepsilon = 1$ while N is increased and the MQ method exhibits spectral convergence. In section 4.2 an algorithm is discussed, for small N (such as in this example), that can stably evaluate the MQ interpolant at all values of the shape parameter without using extended precision.

The outline of the remainder of the book is as follows. In chapter 2 the solution of the scattered data interpolation problem is described and analyzed. The numerical solution of differential equations by the MQ collocation method, which is based on the MQ interpolant, is then discussed in chapter 3. Next, chapter 4 looks at algorithms and techniques that can be used to more accurately and efficiently setup and evaluate RBF approximation methods. Then in chapter 5, shape parameter selection, variations of the MQ method, as well as connections that the MQ method has to other methods, are covered. Finally, in chapter 6, some recent applications of the MQ collocation method are described.

Chapter 2

Scattered Data Approximation

Common choices of RBFs $\phi(r)$ fall into three main categories: compactly supported and finitely smooth; global and finitely smooth; global, infinitely differentiable, containing a free parameter. References [31, 63, 209] can be consulted for more information regarding the large number of basis function that can be used in RBF approximation. The MQ RBF is the focus of this monograph due to its popularity in applications and its good approximation properties. The MQ is representative of the class of RBFs that are global, infinitely differentiable, and that contain a shape parameter. Most results in this work apply to other members of this class as well.

2.1 Alternate Definition of the MQ

It has become common to redefine the MQ (1.1) by first letting $c = 1/\varepsilon$ which results in

$$\phi(r, \varepsilon) = \varepsilon\sqrt{1 + \varepsilon^2 r^2},$$

and then ignoring the scaling factor ε. The redefined MQ is

$$\phi(r, \varepsilon) = \sqrt{1 + \varepsilon^2 r^2}. \tag{2.1}$$

The new definition makes the MQ behave as a function of the shape parameter, ε, as do other infinitely differentiable RBFs containing a shape parameter. Throughout we use c to represent the shape parameter when using the MQ defined as (1.1) and ε when using the MQ defined as (2.1). Figure 2.1 plots the MQ with three different values of the shape parameter and illustrates how the function becomes increasing flat as ε approaches zero.

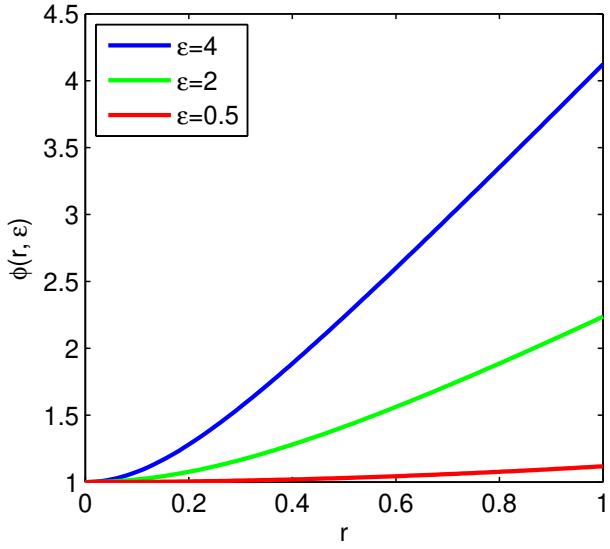

Figure 2.1: The MQ plotted with three different shape parameters.

2.2 RBF Interpolation

RBF interpolation methods use linear combinations of translates of one function $\phi(r)$ of a single real variable. Given a set of ***centers*** $\mathbf{x}_1^c, \ldots, \mathbf{x}_N^c$ in $\mathbb{R}^d$, the RBF interpolant takes the form

$$s(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j \phi(\left\| \mathbf{x} - \mathbf{x}_j^c \right\|_2, \varepsilon) \tag{2.2}$$

where

$$r = \| \mathbf{x} \|_2 = \sqrt{x_1^2 + \cdots + x_d^2}.$$

The coefficients, α, are chosen by enforcing the interpolation condition

$$s(\mathbf{x}_i) = f(\mathbf{x}_i) \tag{2.3}$$

at a set of nodes that typically coincide with the centers. Enforcing the interpolation conditions at N centers results in a $N \times N$ linear system

$$B\alpha = f \tag{2.4}$$

to be solved for the MQ expansion coefficients α. The matrix B with entries

$$b_{ij} = \phi(\left\| \mathbf{x}_i^c - \mathbf{x}_j^c \right\|_2), \qquad i, j = 1, \ldots, N \tag{2.5}$$

is called the ***interpolation matrix*** or the ***system matrix*** and consists of the functions serving as the basis of the approximation space. To evaluate the interpolant at M points $\mathbf{x}_i$ using (2.29), the $M \times N$ ***evaluation matrix*** H is formed with entries

$$h_{ij} = \phi(\left\| \mathbf{x}_i - \mathbf{x}_j^c \right\|_2), \qquad i = 1, \ldots, M \text{ and } j = 1, \ldots, N. \tag{2.6}$$

Then the interpolant is evaluated at the M points by the matrix multiplication

$$f_a = H\alpha. \tag{2.7}$$

A Matlab function that forms the MQ system matrix in two dimension is in listing 2.1. A Matlab function that forms the MQ evaluation matrix in two dimension is in listing 2.2. Both matrices are efficiently formed in Matlab without using for loops.

```
%  SYSTEMMATRIX2D
%
% Output
%    B    the N x N MQ RBF system matrix
% Input
%    xc   N distinct centers

function B = systemMatrixMQ2d(xc,shape)

    [M,N] = size(xc);
    if N>M, xc = xc'; end

     x = xc(:,1);    y = xc(:,2);
     o = ones(1,length(x));

     r = sqrt(  (x*o - (x*o)').^2 + (y*o - (y*o)').^2 );
     B = mq(r,shape);      %  N x N
```

Listing 2.1: systemMatrixMQ2d.m

To illustrate the MQ interpolation method, we use the function

$$\begin{aligned} f(x,y) &= \frac{3}{4}e^{\left[\frac{-1}{4}(9x-2)^2 - \frac{1}{4}(9y-2)^2\right]} + \frac{3}{4}e^{\left[\frac{-1}{49}(9x+1)^2 - \frac{1}{10}(9y+1)^2\right]} \\ &+ \frac{1}{2}e^{\left[\frac{-1}{4}(9x-7)^2 - \frac{-1}{4}(9y-3)^2\right]} - \frac{1}{5}e^{\left[-(9x-4)^2 - (9y-7)^2\right]} \end{aligned} \tag{2.8}$$

that Franke [85] considered in his 1982 test of scattered data approximation methods in which he concluded that the MQ RBF method was the best method among those surveyed. A circular wedge that is a fourth of a circle of radius $\sqrt{2}$ is used as the domain. The left image of figure 2.2 shows the 618 scattered centers used in the problem and the right image of figure 2.2

```
%  EVALUATIONMATRIX2D - for MQ interpolation in 2d
%
% Output
%    H   the M x N MQ RBF evaluation matrix
% Input
%    xc  N  distinct centers
%     x  M  points at which to evaluate the RBF interpolant

function H = evaluationMatrixMQ2d(xc,x,shape)

     [m,n] = size(xc);  if n>m, xc = xc'; end
     [m,n] = size(x);   if n>m,  x = x';  end

      xx = x(:,1);   xy = x(:,2);
     xcx = xc(:,1); xcy = xc(:,2);

      N = length(xcx);  M = length(xx);

     r = sqrt( (xx*ones(1,N) - ones(M,1)*xcx').^2 + ...
               (xy*ones(1,N) - ones(M,1)*xcy').^2 );

     H = mq(r,shape);      %  M x N
```

Listing 2.2: evaluationMatrixMQ2d.m

shows 930 uniformly spaced evaluation points. The approximate function is shown in figure 2.3.

A shape parameter of $\varepsilon = 3$ was used which resulted in a system matrix with a very large condition number (2.12) of $\kappa(B) \approx 10e16$. Dealing with the typically ill-conditioned matrices associated with RBF methods is a major issue that will be discussed at length in chapter 4. Despite the poorly conditioned system matrix, the approximation has a relatively small maximum error of 9.23e-6. The combination of poor conditioning, but good accuracy, is inherent to RBF methods and is explained by the RBF uncertainty principle in section 2.5. The Matlab program that carries out the example is in code listing 2.3.

2.3 Invertibility of the Interpolation Matrix

It is obvious from equation (2.4) that the solution of the MQ interpolation problem will exist and be unique if and only if B is invertible. To establish the invertibility of the MQ interpolation matrix, the definition of a completely monotone function is needed. A function ψ is ***completely monotone*** on $[0, \infty)$ if

1. $\psi \in C[0, \infty)$
2. $\psi \in C^{\infty}(0, \infty)$

```
xc = dlmread('frankeProblemCenters.txt',' ');           % centers
 x = dlmread('frankeProblemEvaluationPoints.txt',' ');  % evaluation points

 N = length(xc(:,1))
 M = length(x(:,1))

     f = frankesFunction(xc(:,1),xc(:,2));
fExact = frankesFunction(x(:,1),x(:,2));
 shape = 3;

    B = systemMatrixMQ2d(xc,shape);
kappa = cond(B)
alpha = B\f;

      H = evaluationMatrixMQ2d(xc,x,shape);
fApprox = H*alpha;

format long
maxError = norm(fApprox - fExact,inf)
format

t = delaunay(x(:,1),x(:,2));
trisurf(t,x(:,1),x(:,2),fApprox)
colormap('Summer')
```

Listing 2.3: interpolationFrankeExample.m

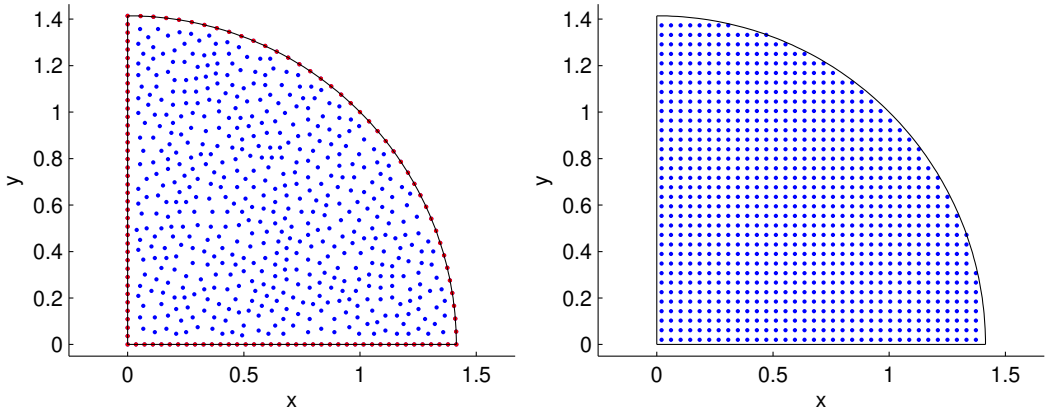

Figure 2.2: Left: 618 centers for the Franke function (2.8) example consisting on 118 boundary centers and 500 interior centers. (2.8). Right: evaluation points for example (2.8).

3. $(-1)^{\ell}\,\psi^{(\ell)}(r) \geq 0$ where $r > 0$ and $\ell = 0, 1, \ldots$

A simplified version of a theorem of Micchelli [166] gives the invertibility of the MQ interpolation matrix.

Theorem 1 *Let $\psi(r) = \phi(\sqrt{r}) \in C[0,\infty)$ and $\psi(r) > 0$ for $r > 0$. Let $\psi'(r)$ be completely monotone and nonconstant on $(0,\infty)$. Then for any set of N distinct centers $\{\mathbf{x}_j^c\}_{j=1}^N$, the $N \times N$ matrix B with entries $b_{jk} =$*

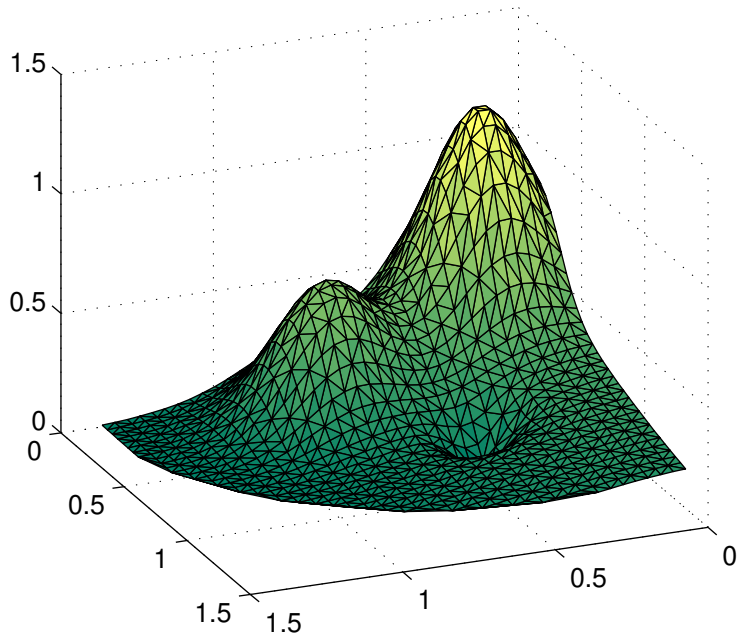

Figure 2.3: MQ approximation of the Franke function (2.8).

$\phi(\left\| \mathbf{x}_j^c - \mathbf{x}_k^c \right\|_2)$ *is invertible. Such a function is said to be* ***conditionally positive definite*** *of order one.*

For the MQ we have

$$\psi(r) = \phi(\sqrt{r}) = \sqrt{1+\varepsilon^2 r}$$

and

$$\begin{aligned}
\psi'(r) &= \frac{\varepsilon^2}{2\sqrt{1+\varepsilon^2 r}} \\
\psi''(r) &= \frac{-\varepsilon^4}{4(1+\varepsilon^2 r)^{3/2}} \\
\psi^{(3)}(r) &= \frac{3\varepsilon^6}{8(1+\varepsilon^2 r)^{5/2}} \\
\psi^{(4)}(r) &= \frac{-15\varepsilon^8}{16(1+\varepsilon^2 r)^{7/2}} \\
\vdots &= \vdots
\end{aligned}$$

Thus

$$(-1)^{\ell}\,\psi^{(\ell)}(r) \geq 0$$

and $\psi'(r)$ is completely monotone and the invertibility of the MQ system matrix B is established.

2.4 Convergence Rates

For sufficiently smooth functions, the MQ RBF method is ***exponentially*** or ***spectrally accurate***. These terms are used to describe a method that has an error that decays at the rate $\mathcal{O}\left(\eta^N\right)$ where $0 < \eta < 1$. This is in contrast to local methods such as finite difference or finite element methods that have algebraic convergence rates where the error decays as $\mathcal{O}\left(N^{-m}\right)$ for some constant m.

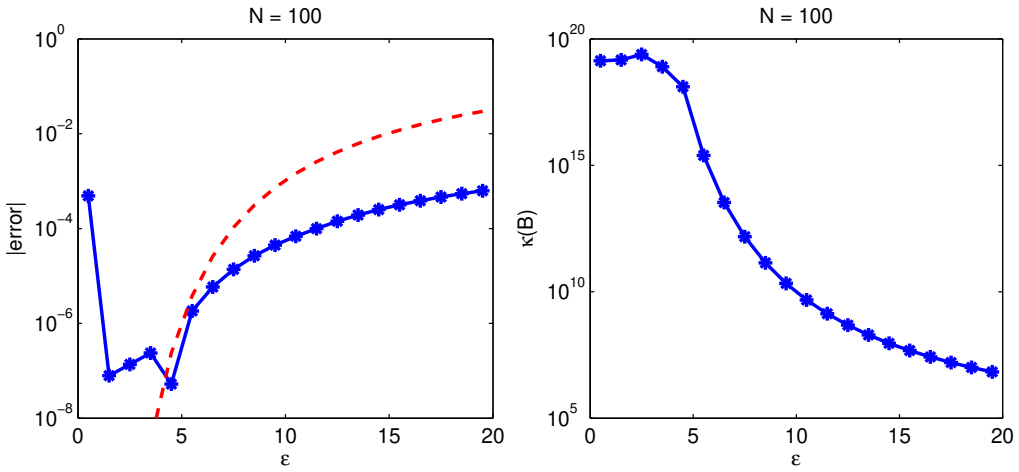

Figure 2.4: Stationary interpolation. N is fixed at $N = 100$. Left: maximum error (solid blue) versus the shape parameter. An error bound of the form (2.11) is plotted in red dashed. Right: the condition number, $\kappa(B)$, versus the shape parameter.

The convergence of RBF methods can be discussed in terms of two different types of approximation - stationary and non-stationary. In ***stationary approximation***, the number of centers N is fixed and the shape parameter ε is refined towards zero. This type of convergence is unique to RBF methods as a parallel does not exist in polynomial based methods. ***Non-stationary***

approximation fixes the value of the shape parameter and N is increased, as would be done in examining the convergence of polynomial based methods.

Scattered data RBF error estimates involve a quantity called the ***fill distance***,

$$h = h_{\Xi,\Omega} = \sup_{\mathbf{x}\in\Omega} \min_{\mathbf{x}_j^c\in\Xi} \left\| \mathbf{x} - \mathbf{x}_j^c \right\|_2 . \tag{2.9}$$

The fill distance indicates how well the set of centers, Ξ, fills out the domain, Ω. Geometrically, the fill distance is the radius of the largest possible empty ball that can be placed among the centers in the domain.

For a fixed shape parameter ε, the MQ interpolant converges to a sufficiently smooth underlying function at a spectral rate as the as the fill distance decreases (which usually corresponds to an increase in N). That is, the error behaves as [28, 154]

$$| f(\mathbf{x}) - s(\mathbf{x})| \leq e^{\frac{-K(\varepsilon)}{h}}. \tag{2.10}$$

In the estimate (2.10), $K(\varepsilon)$ is a constant that depends on the value of the shape parameter. Estimate (2.10) is useful for non-stationary interpolation, but it is unsure of how K varies with ε, which diminishes its utility in quantifying the converge of stationary interpolation.

Another error estimate [157] is

$$| f(\mathbf{x}) - s(\mathbf{x})| \leq K\eta^{1/(\varepsilon h)}, \tag{2.11}$$

where $0 < \eta < 1$. The estimate (2.11) is useful for both stationary and non-stationary interpolation as it shows that spectral convergence results as either the fill distance or shape parameter go to zero.

The theoretical convergence rates may be difficult to achieve computationally due to the condition number of the system matrix growing with both decreasing fill distance and decreasing shape parameter ε. As an example to illustrate stationary and non-stationary convergence, we interpolate the smooth function (1.3) using equally spaced centers on the interval $[-1, 1]$. The stationary interpolation results (***stationaryInterpolation.m***) are in figure 2.4. An error bound of the form (2.11) of $\left(\frac{1}{4}\right)^{1/(\varepsilon h)}$ is plotted in the left figure of the image along with the actual error. The error estimate holds and the expected convergence rate is achieved for $\varepsilon \geq 4.5$. However, below this threshold, the condition number of the system matrix is such that $\kappa(B) > \mathcal{O}\left(10^{18}\right)$ and standard algorithms for solving system (2.4) in 32-bit double precision floating point arithmetic are unable to deliver the predicted accuracy. As N increase in the non-stationary interpolation of this problem,

a similar result (***nonStationaryInterpolation.m***) is observed in figure 2.5 as $\kappa(B)$ becomes large.

In the stationary interpolation example, the minimum error was achieved with a shape parameter of $\varepsilon = 4.5$. Often this value referred to as the "optimal" value of the shape parameter, despite the fact that smaller errors can be realized with smaller values of the shape parameter using extended precision (4.1) and with algorithms that by bypass solving the linear system (2.4). In this case, the term optimal is used the mean the value of the shape parameter that produces the smallest error when solving (2.4) by standard algorithms using standard precision computer arithmetic.

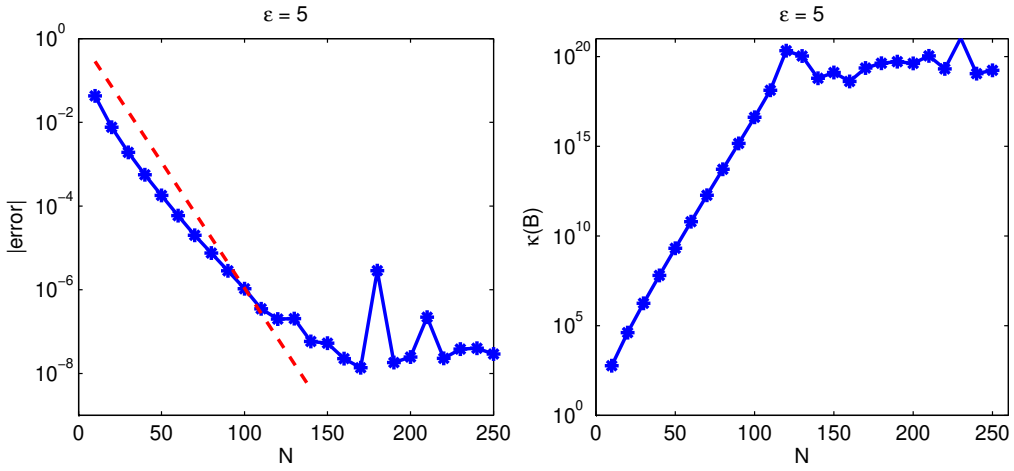

Figure 2.5: Non-stationary interpolation. The shape parameter is fixed at $\varepsilon = 5$. Left: maximum error (solid blue) versus N. An error bound of the form (2.11) is plotted in red dashed. Right: the condition number, $\kappa(B)$, versus N.

When either N or ε is held fixed and the other is refined, theoretical error bounds exist for stationary and non-stationary approximation. However, both N and ε may be allowed to vary and this is the approach that is most often taken in applications. For example, we again interpolate function (1.3), but this time let N vary from 10 to 100 and adjust the shape parameter so that the condition number of the system matrix is in the range $1e15 \leq \kappa(B) \leq 1e17$. The rationale for keeping $\kappa(B)$ in this range is based on the previous two examples which produced the most accurate results when the system matrix had such a condition number. The reasoning will be made

more formal in the next section. We say that the MQ method is ***critically conditioned*** when the system matrix (or the evaluation matrix when solving steady PDE problems [section 3.1]) has a condition number in a range where the MQ method produces accurate results, but the accuracy would deteriorate if the condition number were to increase any further. The results are illustrated in figure 2.6. An exponential convergence rate is observed from $N = 10$ to $N = 40$. The minimum error is with $N = 40$. With larger N the shape parameter must be increased in order to have a system matrix with a desirable condition number. Using more centers with a larger shape parameter is unable to be improved on the accuracy achieved with smaller N and smaller shape. While the particular N and shape parameter are problem dependent, this result is typical. Since the centers are uniformly spaced, the shape parameter increases at a simple linear rate with N in order to maintain the desired conditioning. With scattered centers, such a simple relationship would not exist.

The Matlab code that produces the example is in listing 2.4. Lines 18 to 25 calculate the condition number of the system matrix and then appropriately increase or decrease the shape parameter to lower or raise the condition number. This approach would be very computationally expensive in a problem with large N. Instead, the shape could be specified as being proportional to the minimum separation distance (2.14). If a variable shape parameter (section 5.2) is used, a formula for ε_j could depend on the distance from center $\mathbf{x}_j^c$ to its nearest neighbor. All such strategies have the goal of achieving good accuracy while keeping the conditioning of the problem under control.

2.5 The Uncertainty Principle

A condition number is used to quantify the sensitivity to perturbations of a linear system, such as (2.4), and to estimate the accuracy of a computed solution [203]. Using the 2 norm, the matrix ***condition number*** is

$$\kappa(B) = \| B\|_2 \left\| B^{-1}\right\|_2 = \frac{\sigma_{\max}}{\sigma_{\min}} \tag{2.12}$$

where σ are the singular values of B. A well-conditioned matrix will have a small condition number $\kappa(B) \geq 1$, while and ill-conditioned matrix will have a large condition number. In general, as the condition number increases by a factor of 10, it is likely that one less digit of accuracy will be obtained in a

```
 shape = 5;
     M = 298;
     x = linspace(-1,1,M)';
fExact = exp(sin(pi*x));
    dc = 0.025;
  minK = 1e15;
  maxK = 1e17;
     n = 10:5:100;
     warning off

 for k = 1:length(n);

     N = n(k);
    xc = linspace(-1,1,N)';
     f = exp(sin(pi*xc));

     K = 1;
     while (K<minK | K>maxK)
         B = systemMatrixMQ(xc,shape);
         K = cond(B);
         if K<minK
             shape = shape - dc;
         elseif K>maxK
             shape = shape + dc;
         end
      end

              H = evaluationMatrixMQ(xc,x,shape);
          alpha = B\f;
        fApprox = H*alpha;

    maxError(k) = norm(fApprox - fExact,inf);
    shapeP(k) = shape;

 end

 semilogy(n,maxError,'b*-')
 xlabel 'N', ylabel '|error|'
 figure
 plot(n,shapeP);
 xlabel 'N', ylabel 'shape, \epsilon'
 warning on
```

Listing 2.4: hybridInterpolation.m

computed solution. The MQ system matrix B is symmetric, thus its singular values are equal to the absolute value of its eigenvalues, λ, and the condition number can be restated as

$$\kappa(B) = \frac{|\lambda|_{\max}}{|\lambda|_{\min}}. \tag{2.13}$$

Section 2.6 discusses another condition number that may be a better indicator of the accuracy that can be obtained from a RBF approximation.

In section 2.4, the interpolation error was quantified in terms of the fill distance. This seems appropriate for the analysis of convergence rates because the fill distance measures how well the centers cover the region Ω. However, the fill distance is not a good measure of the conditioning of the problem. The reason is that a set of centers may have a fairly large fill distance but the system matrix may be very ill-conditioned if only two of the

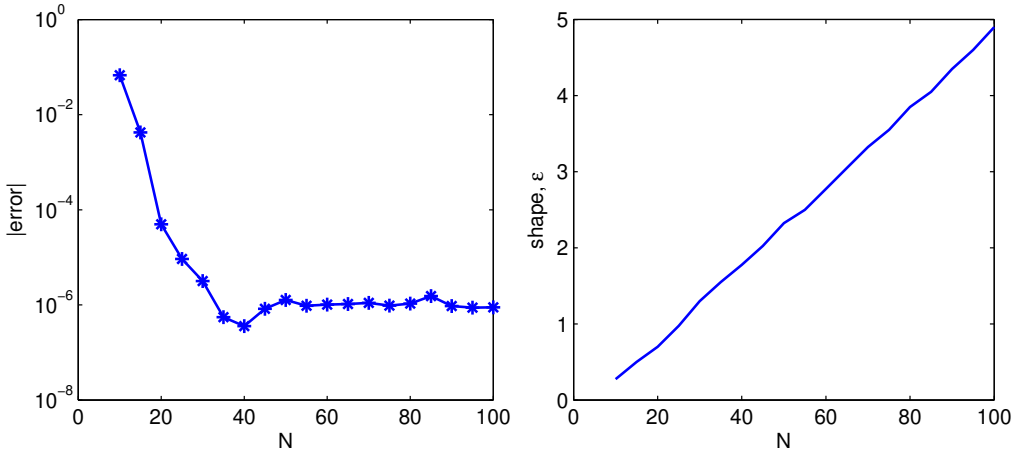

Figure 2.6: Interpolating function (1.3) while varying both N and ε. Left: Maximum error versus N. Right: Shape versus N.

centers are very close, which results in two rows of the system matrix being nearly identical. Stability is better quantified in terms of the ***minimum separation distance***

$$q_\Xi = \frac{1}{2} \min_{i \neq j} \left\| \mathbf{x}_i^c - \mathbf{x}_j^c \right\|_2 . \tag{2.14}$$

The minimum separation distance can be visualized as the radius of the largest ball that can be placed around each center in Ξ such that none of the balls overlap.

An estimate of the condition number of the MQ system matrix B can be made by obtaining upper and lower bounds on its eigenvalues. Gershgorin's theorem [165] states that

$$|\,\lambda_{\max} - b_{ii}| \leq \sum_{j=1, j \neq i}^{N} |\,b_{ij}|\,, \quad i = 1, 2, \ldots, N$$

and it allows us to bound the maximum eigenvalue in absolute value as

$$|\,\lambda|_{\max} \leq N \,\|\,B\|_\infty .$$

For $|\,\lambda|_{\min}$, the lower bound

$$|\,\lambda|_{\min} \geq q_\Xi^{1-d/2} e^{(2M_d)/(\varepsilon q_\Xi)} \tag{2.15}$$

in terms of the minimum separation distance holds where d is the space dimension and M_d is a constant depending on the space dimension. Although the estimate for largest eigenvalue appears bad, the linearly growing upper bound for $|\lambda|_{\max}$ is not the cause of the MQ stability problem. It is the exponentially decaying lower bound for $|\lambda_{\min}|$ that is to blame. For a fixed set of centers, the bound decays exponentially as the shape parameter decreases. Similarly, for a fixed value of the shape parameter, the lower bound decreases exponentially as the minimum separation distance decreases.

The convergence results require that the fill distance is small (centers close together) and/or the shape parameter be small in order to obtain good accuracy. The conditioning results require that in order for the RBF system matrix to be well-conditioned that the shape parameter and minimum separation distance be large (centers far apart). Obviously, both situations cannot occur at the same time. This observation has been referred to as the ***uncertainty principle*** [189]. In RBF methods we cannot have both good accuracy and good conditioning at the same time. The uncertainty principle is apparent in the stationary and non-stationary interpolation examples that are illustrated in figures 2.4 and 2.5. The smallest errors in both examples occur when the condition number of the system matrix is approximately 10e18. In figure 2.7, the above error bounds are illustrated. In both the stationary and non-stationary examples the maximum eigenvalue grows slowly with ε and N. However, the smallest eigenvalue rapidly deceases as ε decreases or N grows (q_Ξ decreases).

2.6 Local versus Global Condition Numbers

A condition number is used to provide bounds on the error in solving a linear system, $B\alpha = f$, caused by the perturbations in the matrix B and the right vector f. The perturbed system is

$$(B + \Delta B)(\alpha + \Delta\alpha) = f + \Delta f$$

where α is the exact solution of the unperturbed system. We denote a computed solution by $\hat{\alpha} = \alpha + \Delta\alpha$ and the residual by $r = \Delta f = f - B\hat{\alpha}$.

The condition number κ (2.12), often referred to as the ***global*** or ***absolute condition number***, was introduced in section 2.5. Another matrix condition number associated with the linear system, $B\alpha = f$, is the local or

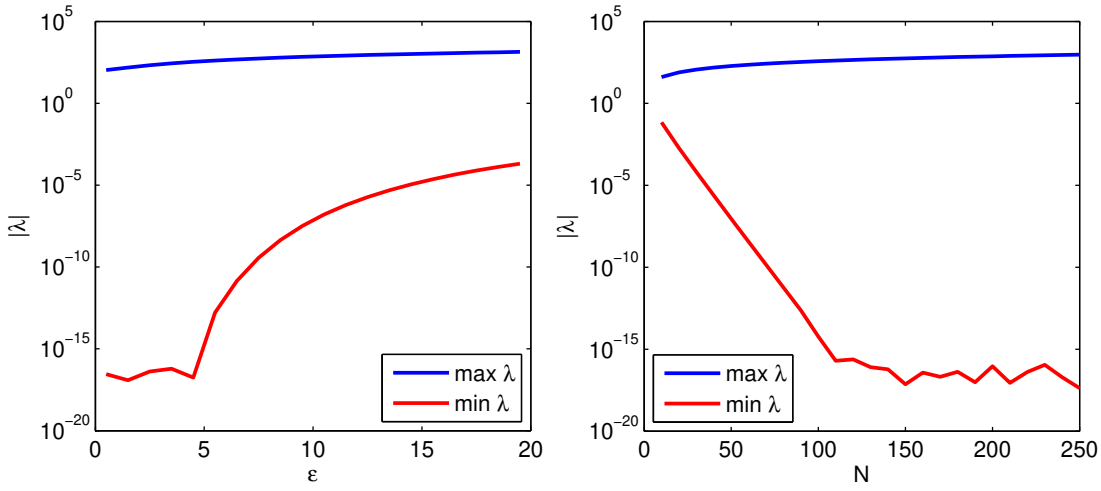

Figure 2.7: Maximum (upper blue) and minimum (lower red) eigenvalues of the system matrices form the stationary and non-stationary interpolation examples of figures 2.4 and 2.5. Left: stationary. Right: non-stationary.

natural condition number

$$\kappa_{loc}(B) = \frac{\| B^{-1}\|_2 \, \| f\|_2}{\| \alpha\|_2} = \frac{\| f\|_2}{\| \alpha\|_2 \, \sigma_{\min}}. \tag{2.16}$$

The number $\sigma_{\min}$ is the smallest singular value of B. Both condition numbers can be used to bound the relative error of the solution of the linear system by the relative residual as

$$\frac{\| \Delta\alpha\|}{\| \alpha\|} \leq \text{condition number} \times \frac{\| r\|}{\| f\|}. \tag{2.17}$$

The two condition numbers are related by

$$1 \leq \kappa_{loc}(B) \leq \kappa(B). \tag{2.18}$$

To calculate the local condition number, the unknown exact solution α can be estimated by using the singular value decomposition (SVD) of the matrix B (section 4.3). The SVD of the non-singular matrix B is $B = U\Sigma V^T$. The matrices U and V are orthogonal matrices and Σ is a diagonal matrix with the N singular values as its entries. The columns of U are an

orthonormal basis for $\mathcal{R}^N$, so the right vector f can be expanded in this basis as

$$f = \sum_{i=1}^{N} \beta_i u_i \tag{2.19}$$

where the u_i are columns of U and the expansion coefficients are $\beta_i = u_i^T f$. Using the SVD, the solution vector α is $\alpha = V\Sigma^{-1}U^T f$. The 2 norm of α is

$$\| \alpha\|_2 = \left\| V\Sigma^{-1}U^T f\right\|_2 = \left(\sum_{i=1}^{N} \frac{\beta_i^2}{\sigma_i^2} \right)^{\frac{1}{2}} \tag{2.20}$$

and since U and V are orthogonal matrices,

$$\| f\|_2 = \left(\sum_{i=1}^{N} \beta_i^2 \right)^{\frac{1}{2}}.$$

Thus, κ_{loc} can be computed by the formula

$$\kappa_{loc}(B) = \frac{\| f\|_2}{\| \alpha\|_2 \, \sigma_{\min}} = \frac{\sqrt{\beta_1^2 + \ldots \beta_N^2}}{\sigma_N \sqrt{(\beta_1/\sigma_1)^2 + \cdots + (\beta_N/\sigma_N)^2}}. \tag{2.21}$$

The following Matlab code calculates both the local and global condition number of a matrix B using the SVD:

```
[U,S,V] = svd(B);
      S = diag(S);

globalConditionNumber = max(S)/min(S);

         b = U'*f;
localKappa = sqrt(sum(b.^2))/(S(N)*sqrt(sum((b./S).^2)));
```

When the vector f is such that the solution α is parallel to the singular vector u_1, then $\beta_2 = \cdots = \beta_N = 0$. In this case $\| f\|_2 = |\beta_1|$ and $\kappa = \sigma_1/\sigma_N = \kappa_{loc}$. However, this special case rarely if ever occurs in applications and the local condition number is a better upper bound for the relative errors of the solution of linear systems than is the global condition number.

It has been observed [107] that in RBF collocation methods for PDEs that the RBF evaluation and system matrices are often such that $\kappa_{loc} \ll \kappa$.

This helps explain the excellent accuracy of the RBF methods at very large global condition numbers. A similar relationship exists between the global and local condition number in many other numerical methods for PDEs. The paper [138] and the references within can be consulted for details.

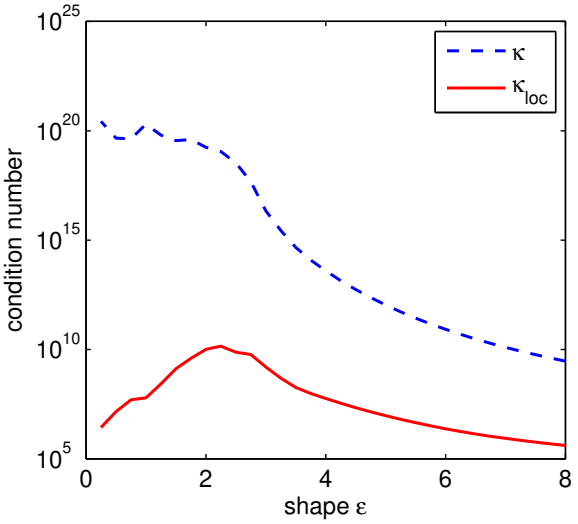

Figure 2.8: Local and global condition numbers over a range of shape parameters for the Poisson problem (4.1) with $\lambda = 1$ and $\mu = 2$.

As an example (***poissonLocalGlobalKappa.m***), the local and global condition numbers over a range of shape parameters are examined for problem (4.1). As shown in figure 4.6, the minimum error of about 10e-6 is reached for this problem with $\lambda = 1$ and $\mu = 2$ using a shape parameter of $\varepsilon \approx 2$. At this value of the shape parameter, $\kappa(H) = \mathcal{O}(10^{20})$ while $\kappa_{loc}(H) = \mathcal{O}(10^{10})$.

The same disparity between the local and global condition numbers seems to exist for the RBF system matrix in interpolation problems (***frankeProblemLocalGlobalKappa.m***). In figure 2.9, the local and global condition numbers are plotted versus a range of the shape parameter from the interpolation of the Franke function (2.8) from section 2.2. The smallest error is when $\varepsilon \approx 3$ and $\kappa(B) = \mathcal{O}(10^{18})$, but the local condition number has a significantly smaller value of $\kappa_{loc}(B) = \mathcal{O}(10^{10})$.

Despite realizing that the local condition number may be a better indicator of the accuracy of a RBF approximation, we continue to focus more on the global condition throughout. This is largely due to the fact that numerical software packages, such as Matlab, have a built-in function that efficiently

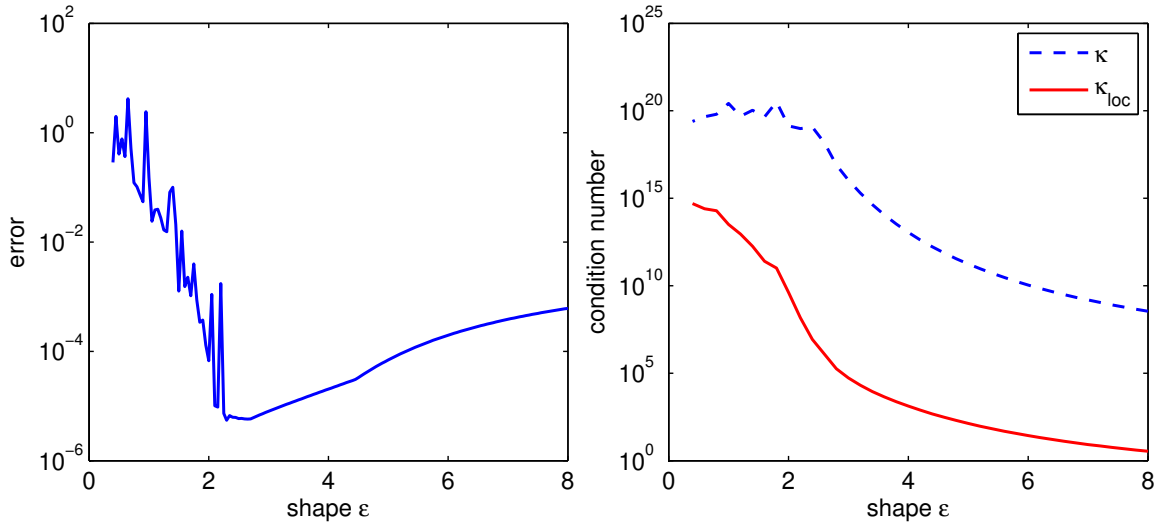

Figure 2.9: The interpolation of the Franke function (2.8) from section 2.2. Left: Accuracy versus shape parameter. Right: local (solid red) and global condition (dashed blue) number versus the shape parameter.

estimate the condition number in the 1 norm, rather than computing a SVD to calculate the global or local condition number in the 2 norm.

2.7 Good Data Independent Center Locations

RBF methods allow complete freedom in selecting center locations. When approximating functions with sharp gradients and rapidly varying features, adaptive methods (section 5.7) can be used to place more centers in regions with finer detail and less in areas where the function is smoother. In other situations, it may be desirable to locate centers in a data dependent way that does not assume any knowledge of the underlying function. The following strategy assumes a constant value of the RBF shape parameter and places centers so that they provide a fairly uniform coverage of a region. The algorithm is designed to produce sets of centers that have associated RBF matrices that have close to the lowest possible condition number on the given domain.

The conditioning and convergence results of the previous sections indicate that good sets centers are ones with large minimum separation distances

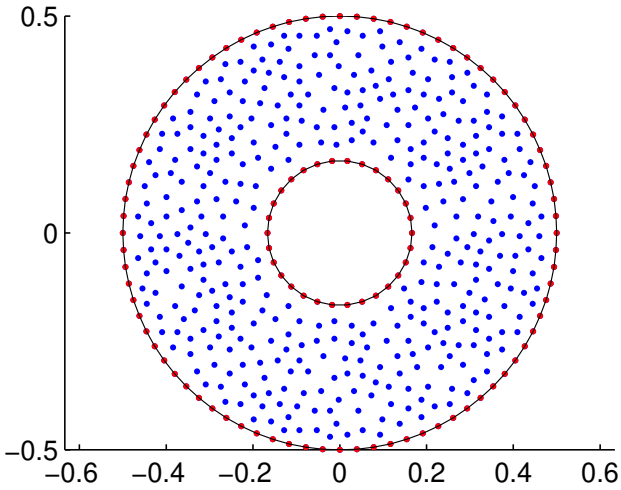

Figure 2.10: 500 near-optimal, data independent, centers with uniform boundary spacing on an annulus.

(2.14) and small fill distances (2.9). This can be accomplished with the geometric greedy algorithm of [164] which produces well-distributed near-optimal center sets in this sense. Additionally, the near-optimal center sets are not function dependent. The center sets are produced by adding a new center to fill the currently largest hole in the existing center set. In order to reduce the error in boundary regions, we have found it beneficial to start with a uniform coverage of the boundary and then add points to the interior according to the algorithm.

The program in listing 2.5 uses the geometric greedy algorithm to produce a near-optimal center set for the domain consisting of the region between two circles, of radius $r = 1/6$ and $r = 1/2$. The code has not been optimized for efficiency. The boundaries are first covered with equally spaced centers, 30 on the inner boundary and 80 on the outer boundary. Then 390 centers are added according to the algorithm. The added centers are selected from a larger set of uniformly spaced centers that cover the domain. The resulting set of centers is shown in figure 2.10.

```
1
2  function [Xn,bpi] = nearOptimalCentersAnnulus(NI,NO,N,M,interiorN)
3  % example usage: [Xn,bpi] = nearOptimalCentersAnnulus(30,80,200,200,390);
4  % equally space NI centers on interior boundary and NO on the outter
5     Xn = zeros(NI + NO + interiorN,2);
6     omegaNt = zeros(N*M,2);
7     ti = linspace(0, 2*pi, NI+1)'; ti = ti(1:NI);  r = 1/6;    % inner boundary
8     Xn(1:length(ti),1) = r.*cos(ti);
9     Xn(1:length(ti),2) = r.*sin(ti);
10    to = linspace(0, 2*pi, NO+1)';  to = to(1:NO);  ro = 0.5;  % outter boundary
11    Xn(length(ti)+1:length(to)+length(ti),1) = ro.*cos(to);
12    Xn(length(ti)+1:length(to)+length(ti),2) = ro.*sin(to);
13    nbPts = NI + NO;
14
15 % --- build a set of center candidates  ---------------------------
16
17       [X,Y] = meshgrid(linspace(-0.5,0.5,N),linspace(-0.5,0.5,M));
18       xL = reshape(X,N*M,1);  yL = reshape(Y,N*M,1);
19       [th,p] = cart2pol(xL,yL); I = 1;
20       for i=1:length(th)
21          t = th(i);  r = p(i); ri = 1/6;  ro = 0.5;
22         if (r>ri) & (r<ro)   % add centers between boundaries to omegaN
23           if (xL(i)~=0 & yL(i)~=0)
24             omegaNt(I,:) = [xL(i) yL(i)];  I = I + 1;
25           end
26         end
27       end
28       omegaN = omegaNt(1:I-1,:);  clear omegaNt
29
30 % ------------ add interiorN centers to Xn --------------------------
31
32              N = size(omegaN,1);        % number of candidate centers
33            bpi = 1:nbPts;               % index of boundary points in the  Xn(i,:)
34              n = nbPts;                 % number of centers currently in Xn
35     minDstToXn = zeros(N,1); % x in omegaN \ Xn,  distance to its nearest neighbor
36
37      for k=1:N
38        minDst = 100;
39        for j=1:n
40          dst = norm(omegaN(k,:)-Xn(j,:),2);
41          if dst<minDst & dst>eps, minDst = dst; end % omegaN in Xn already check
42          if  dst<eps, minDst = 0; end               % to within machine precision
43        end
44        minDstToXn(k) = minDst;
45      end
46
47      n = n + 1;                                      % add first interior point
48      [notUsed,indexAddToXn] = max(minDstToXn); Xn(n,:) = omegaN(indexAddToXn,:);
49      minDstToXn(indexAddToXn) = 0;
50
51     for k = 1:interiorN
52          for k=1:N                                % update distances for added point
53             dst = norm(omegaN(k,:)-Xn(n,:),2);
54             if dst<minDstToXn(k) & dst>eps, minDstToXn(k) = dst; end
55             if  dst<eps, minDst = 0; end
56          end
57           n = n + 1;                                % add next interior point
58           [notUsed,indexAddToXn] = max(minDstToXn); Xn(n,:) = omegaN(indexAddToXn,:);
59           minDstToXn(indexAddToXn) = 0;
60     end
```

Listing 2.5: nearOptimalCentersAnnulus

2.8 Errors in Boundary Regions

When RBF centers cover a domain in a fairly uniform manner, as they do with the near optimal center algorithm of section 2.7, the largest errors will typically occur near the boundary. That is of course, unless the function has complex features or rapid variations located exclusively in the interior of the domain. An example is in the right image of figure 2.12. The smoothly varying function (3.11) is interpolated on the unit circle using $N = 100$ centers that are marked with red x's in figure 2.11. A shape parameter of $\varepsilon = 0.2$ was used and the system matrix has a condition number of $\kappa(B) = \mathcal{O}(10e19)$. The interpolant was evaluated at $M = 290$ evaluation sites and the overall accuracy is very good, as the maximum error is 7.5e-9. However, as figure 2.12 shows, the largest errors do occur near the boundary.

An obvious approach to reduce errors in boundary regions is to locate centers more densely in boundary regions than in interior regions. Boundary clustering is effective, but with a fixed number of N centers, the approach should be used with caution as too much boundary clustering will reduce accuracy in interior regions. Additional support for the use of boundary clustering via potential theory is given in [177].

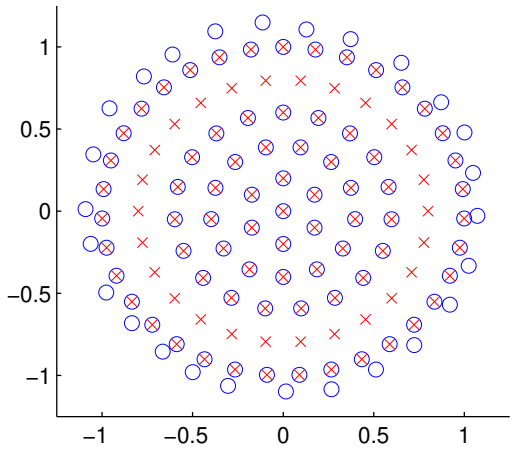

Figure 2.11: Centers (blue circles) and data points (red x's) for the NaK interpolation example.

Another strategy to reduce errors in boundary regions is to separate the location of the centers and the data points where the interpolation condition is enforced. The centers that are nearest to the boundary are moved outside

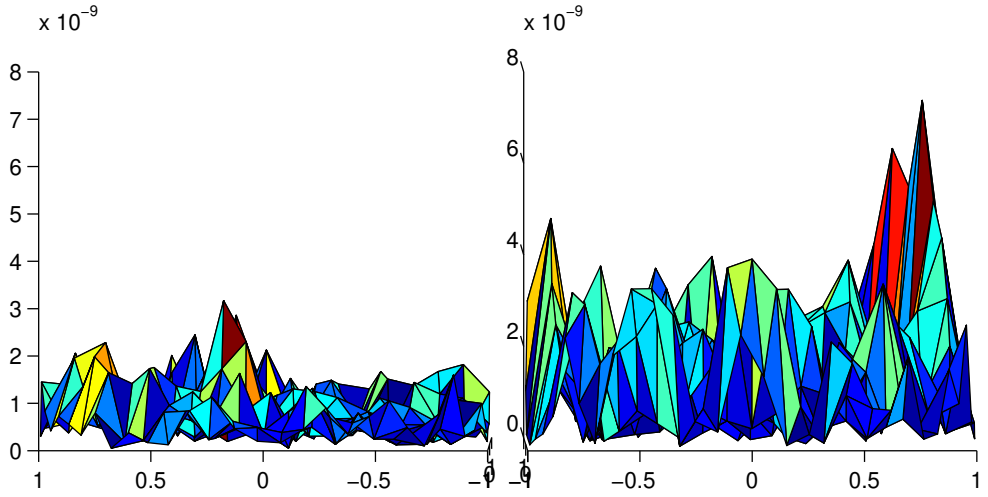

Figure 2.12: Left: NaK pointswise errors. Right: standard MQ method interpolation errors.

the domain, but the locations remain data points. Figure 2.11 illustrates this, with the centers being marked with circles and the data points with x's. Using the center and data points in the figure to solve the previous interpolation problem (***notAKnotInterpolationUnitCircle.m***), again with $\varepsilon = 0.2$, the system matrix has a condition number of $\kappa(B) = \mathcal{O}(10e19)$, as was the case in the previous example. The maximum error is 3.1e-9 and as can be seen from the point-wise errors in the left image of figure 2.12, the errors in the boundary regions are smaller than in the standard implementation. In reference [76], the above strategy is referred to as a ***Not-a-Knot (NaK)*** approach due to its connection with polynomial splines. The strategy can be extended to move more centers then just the centers next to the boundary out of the domain in order to further increase accuracy. Such a scheme is called as a generalized NaK method. Applications of the generalized NaK approach to interpolation problems and to Elliptic boundary value problems can be found in [129].

2.9 Approximating Derivatives

By linearity, the RBF expansion (2.29) can be used to approximate the derivatives of the function $f(\mathbf{x})$ as

$$\frac{\partial}{\partial x_i} s(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j \frac{\partial}{\partial x_i} \phi(\left\| \mathbf{x} - \mathbf{x}_j^c \right\| ; \varepsilon). \tag{2.22}$$

Higher order derivatives, partial derivatives, and mixed partial derivatives, are handled in a similar manner. After finding the RBF expansion coefficients from the interpolation problem, the derivative of the function can be evaluated via (2.22). The m-file ***mqDerivatives.m*** contains formulas for the first and second derivatives of the MQ.

If (2.22) is evaluated at the centers $\left\{\mathbf{x}_j^c\right\}_{j=1}^N$ and written vector-matrix notation we have

$$\frac{\partial}{\partial x_i} s(\mathbf{x}) = \frac{\partial}{\partial x_i} H\alpha \tag{2.23}$$

where the evaluation matrix is the $N \times N$ matrix $\frac{\partial}{\partial x_i} H$ with entries

$$h_{ij} = \frac{\partial}{\partial x_i} \phi(\left\| \mathbf{x}_i^c - \mathbf{x}_j^c \right\|_2), \qquad i, j = 1, \ldots, N. \tag{2.24}$$

The notation H_{x_i} indicates a matrix with such entries. Likewise, $H_{x_i x_i}$ is used for second derivatives. Higher derivatives and mixed partial derivatives are denoted using similar notation.

By substituting $\alpha = B^{-1} f$ into equation (2.23), the ***differentiation matrix***

$$D = \frac{\partial}{\partial x_i} H B^{-1} \tag{2.25}$$

can be defined. The derivative of the function $f(\mathbf{x})$ at the centers $\left\{\mathbf{x}_j^c\right\}_{j=1}^N$ can be approximated by the single matrix multiplication

$$\frac{\partial}{\partial x_i} \approx \frac{\partial}{\partial x_i} s(\mathbf{x}) = Df$$

in $\mathcal{O}\left(N^2\right)$ flops. The differentiation matrix is well-defined since it is known that the system matrix B is invertible.

For any sufficiently differentiable RBF, $\phi[r(\mathbf{x})]$, the chain rule gives

$$\frac{\partial \phi}{\partial x_i} = \frac{d\phi}{dr}\frac{\partial r}{\partial x_i} \tag{2.26}$$

and

$$\frac{\partial^2 \phi}{\partial x_i^2} = \frac{d\phi}{dr}\frac{\partial^2 r}{\partial x_i^2} + \frac{d^2\phi}{dr^2}\left(\frac{\partial r}{\partial x_i}\right)^2 \tag{2.27}$$

where

$$\frac{\partial r}{\partial x_i} = \frac{x_i}{r}$$

and

$$\frac{\partial^2 r}{\partial x_i^2} = \frac{1 - \left[\frac{\partial r}{\partial x_i}\right]^2}{r}.$$

For the MQ in particular,

$$\frac{d\phi}{dr} = \frac{\varepsilon^2 r}{\sqrt{1+\varepsilon^2 r^2}}$$

and

$$\frac{d^2\phi}{dr^2} = \frac{\varepsilon^2}{[1+\varepsilon^2 r^2]^{3/2}}.$$

Higher derivatives and mixed partial derivatives are calculated in a similar manner.

2.10 The Generalized MQ

A generalized version of the MQ RBF (GMQ) is

$$\phi(r;\varepsilon) = (1+\varepsilon^2 r^2)^\beta \tag{2.28}$$

where the exponent β may be any real number except the non-negative integers. Obviously, with $\beta = 1/2$, the GMQ reduces to the MQ. With $\beta = -1/2$ the GMQ is equal to the inverse multiquadric RBF, and with $\beta = -1$ is equal to the inverse quadratic RBF. For $\beta < 0$ the generalized MQ is strictly positive definite and for $0 < \beta < 1$ the generalized MQ is conditionally positive definite of order one. In both cases the system matrix for the interpolation problem can be shown to be invertible. With $\beta > 1$, the generalized MQ

is conditionally positive definite of order $\lceil \beta \rceil$ and in order to show that the system matrix B is invertible it is necessary to append low order polynomials to the RBF interpolant (2.29). However, in applications it is common to use the generalized MQ with $\beta > 1$ without the appended polynomials and good results are reported without encountering singular matrices.

Some researchers have reported good results with non-standard values of the GMQ exponent. Good accuracy with exponents $\beta = 1.03$ and $\beta = 1.99$ was described in [205] and [214] respectively. However, the utility of using the GMQ rather than the MQ is questionable and is easily explained via the uncertainty principle. The stationary interpolation example of section 2.4 is repeated (***frankeProblemGeneralizedMq.m***) with the GMQ exponents $\beta = -1.5, 0.5, 1.03, 1.99, 3.5$. As illustrated in the right image of figure 2.13, the condition number of the system matrix increases with the exponent at each value of the shape parameter. In the right image of figure 2.13, accuracy versus the shape parameter is shown and the expected correlation between good accuracy and poor conditioning is observed. The GMQ with $\beta = 3.5$ has a system matrix with the highest condition number over the set of shape parameters and produces the most accurate approximation over a large portion of the set of shape parameters. However, all produce roughly the same accuracy when their system matrices become ***critically conditioned***. Over the particular discrete set of shape parameters sampled in this example, the MQ produces the single most accurate approximation. There seems to be no particular advantage to using $\beta \neq 0.5$. The important factor is that the basis functions be employed at a shape parameter for which their system matrix is critically conditioned. A more detailed numerical study of the GMQ was recently conducted in [44].

2.11 Least Squares mode

Instead of enforcing the interpolation conditions (2.3), RBF methods can be implemented in a least squares (LSQ) mode. In least squares mode the number of basis functions, M, is less than the number of data locations, N. Least squares approximation is useful in several situations that include noisy data and problems with a large number of data locations. For convenience, we consider the case where the centers, Ξ, are a subset of the N data locations, χ, at which f is known. More general LSQ methods exist in which the location of the data sites and centers do not coincide.

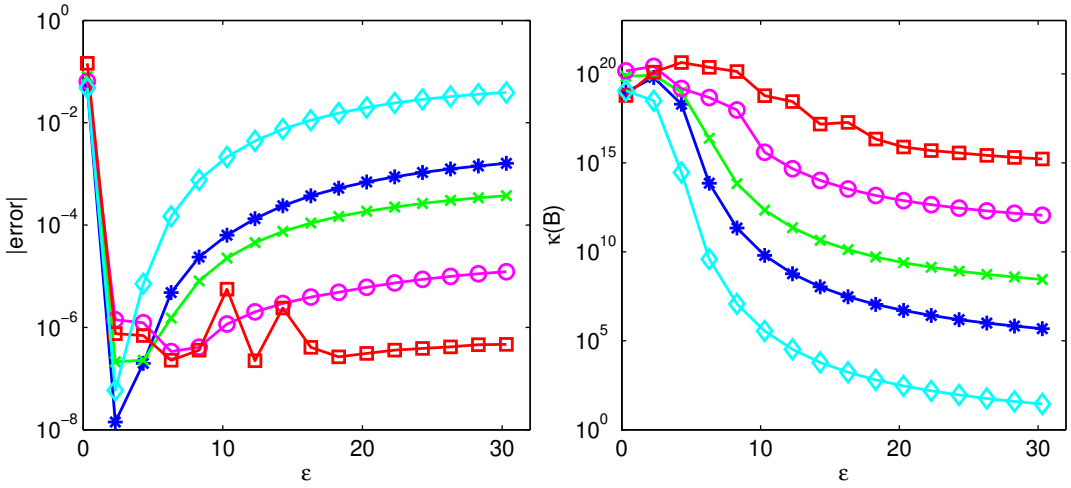

Figure 2.13: Generalized MQ stationary interpolation: $\beta = -1.5$ (cyan diamond), the MQ $\beta = 0.5$ (blue asterisk), $\beta = 1.03$ (green x), $\beta = 1.99$ (magenta circles), $\beta = 3.5$ (red squares). Left: Interpolation error versus the shape parameter. Right: Condition number of the system matrix versus the shape parameter.

The LSQ mode RBF approximant is

$$s(\mathbf{x}) = \sum_{j=1}^{M} \alpha_j \phi(\left\| \mathbf{x} - \mathbf{x}_j^c \right\|_2, \varepsilon). \tag{2.29}$$

The expansion coefficients are found by solving the over determined linear system, $B\alpha = f$, with the system matrix that has elements

$$b_{ij} = \phi(\left\| \mathbf{x}_i^c - \mathbf{x}_j^c \right\|_2), \qquad i = 1, \ldots, N,\ j = 1, \ldots, M. \tag{2.30}$$

In Matlab, the linear system can still be solved via the backslash operator that is used for the interpolation problem. For over determined systems, the backslash operator solves the system in a least squares sense using a QR decomposition.

Next, the example from section 2.2, which approximated function (2.8) using RBF interpolation, is reworked using a least squares approach. From the original $N = 618$ centers, 100 ($M = 518$) of interior centers are only used as data locations (right image of figure 2.14). The results in the left

image of figure 2.14 show that the LSQ method is more accurate than the interpolation method on this problem. Listing 2.6 displays a Matlab script implements the LSQ MQ method for the problem.

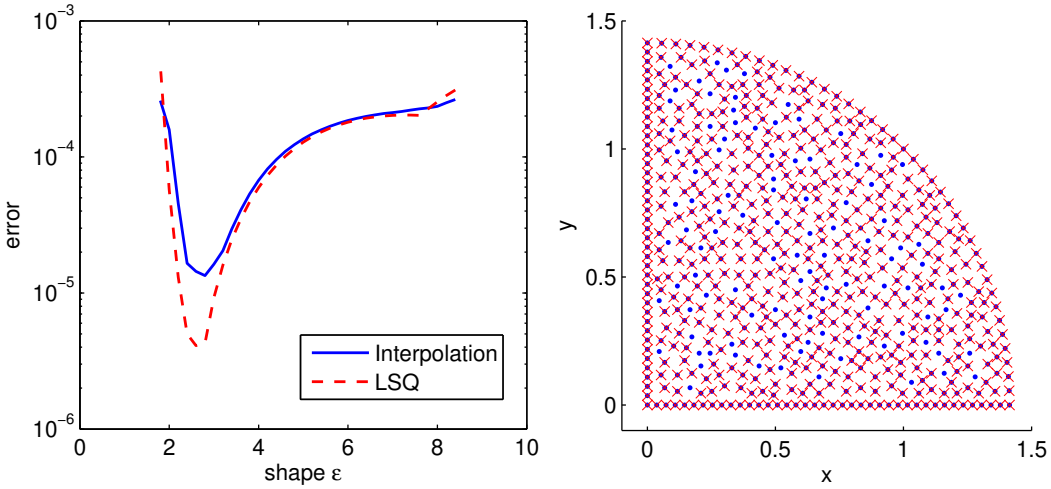

Figure 2.14: Left: Interpolation error (blue solid) and LSQ error (red dashed) versus the shape parameter. Right: The points marked with a blue dot are data locations only and not centers.

Hardy first used a RBF least squares method in his original paper [97] and he again discussed the least squares approach in the review paper [99]. Since that time, a variety of RBF least square methods have been suggested. The book [63] and the references within may be consulted for details.

2.12 Chapter Summary

The MQ RBF method is an effective method for scattered data approximation in complexly shaped domains. Implementing the method is essentially the same in higher dimensions as it is in one dimension. Theoretically, the MQ method is spectrally accurate. The spectral accuracy of the method can be achieved if the centers are not too close together. However, if the minimum separation distance is too small, conditioning problems may prevent the spectral accuracy from being realized. The meshfree property of the MQ method allows for complete freedom in choosing the location of centers. Unless the underlying function is known to have steep gradients or detailed

```
1
2  xc = dlmread('frankeProblemCenters.txt',' ');     %  N data points
3  xs = dlmread('frankeProblemCentersLsq.txt',' ');  %  S centers, xs a subset of xc
4   x = dlmread('frankeProblemEvaluationPoints.txt',' ');  % M evaluation points
5
6       N = length(xc(:,1));  M = length(x(:,1));  S = length(xs(:,1));
7
8           f = frankesFunction(xc(:,1),xc(:,2));
9      fExact = frankesFunction(x(:,1),x(:,2));
10
11     r = zeros(N,S);
12     for i=1:N
13       for j=1:S
14         r(i,j) = sqrt( ( xc(i,1) - xs(j,1)  )^2 + ( xc(i,2) - xs(j,2) )^2  );
15       end
16     end
17
18     rh = zeros(M,S);
19     for i=1:M
20       for j=1:S
21         rh(i,j) = sqrt( ( x(i,1) - xs(j,1)  )^2 + ( x(i,2) - xs(j,2) )^2  );
22       end
23     end
24
25         shape = 2.6;
26             B = mq(r,shape);   %  system matrix
27         lambda = B\f;
28
29           H = mq(rh,shape);
30     fApprox = H*lambda;
31
32     maxError = norm(fApprox - fExact,inf)
```

Listing 2.6: frankeProlemLsq

localized features (section 5.7), using centers that cover a domain in a fairly uniform manner is recommend so that the fill distance (2.9) and minimum separation distance (2.14) are balanced.

Chapter 3

Solving PDEs by Asymmetric MQ Collocation

The asymmetric MQ collocation method is so named due to the fact that its evaluation matrix is not symmetric. To describe the asymmetric MQ collocation method for PDEs, we consider a linear steady problem and a linear time-dependent problem. Let $\mathcal{L}$ be a linear differential operator. The steady problem is

$$\mathcal{L}u = f \quad \text{in } \Omega \tag{3.1}$$

and the time-dependent problem is

$$\frac{\partial u}{\partial t} = \mathcal{L}u \quad \text{in } \Omega. \tag{3.2}$$

Both PDEs have boundary conditions imposed on all or parts of the boundary, $\partial\Omega$, by a boundary operator, $\mathcal{B}$, so that the PDE is well-posed. Information on the correct specification of boundary conditions can be found in standard introductory text books on PDEs such as reference [59].

Let Ξ be a set of N distinct centers that are divided into two subsets. One subset contains N_I centers, $\mathbf{x}_I^c$, where the PDE is enforced and the other subset contains $N_\mathcal{B}$ centers, $\mathbf{x}_\mathcal{B}^c$, where boundary conditions are enforced. For simplicity, it is assumed that the centers are in an array that is ordered as $\Xi = [\mathbf{x}_I^c; \mathbf{x}_\mathcal{B}^c]$.

The MQ collocation method applies the operator $\mathcal{L}$ to the MQ interpolant (2.29) as

$$\mathcal{L}u(\mathbf{x}_i^c) = \sum_{j=1}^{N} \alpha_j \mathcal{L}\phi(\left\| \mathbf{x}_i^c - \mathbf{x}_j^c \right\|_2), \quad i = 1, \ldots, N_I, \tag{3.3}$$

at the N_I interior centers and applies an operator $\mathcal{B}$ which enforces boundary conditions as

$$\mathcal{B}u(\mathbf{x}_i^c) = \sum_{j=1}^{N} \alpha_j \mathcal{B}\phi(\left\| \mathbf{x}_i^c - \mathbf{x}_j^c \right\|_2), \quad i = N_I + 1, \ldots, N. \tag{3.4}$$

at the N_B boundary centers. In matrix notation, the right side of equations (3.3) and (3.4) can be written as $H\lambda$, where the evaluation matrix H that discretizes the PDE consists of the two blocks

$$H = \begin{bmatrix} \mathcal{L}\phi \\ \mathcal{B}\phi \end{bmatrix}. \tag{3.5}$$

The two blocks of H have elements

$$\begin{aligned} (\mathcal{L}\phi)_{ij} &= \mathcal{L}\phi(\left\| \mathbf{x}_i^c - \mathbf{x}_j^c \right\|_2), \qquad i = 1, \ldots, N_I \quad j = 1, \ldots, N \\ (\mathcal{B}\phi)_{ij} &= \mathcal{B}\phi(\left\| \mathbf{x}_i^c - \mathbf{x}_j^c \right\|_2), \qquad i = N_I + 1, \ldots, N \quad j = 1, \ldots, N. \end{aligned}$$

From the interpolation problem in section 2.2, we have that $\alpha = B^{-1}u$ where B is the system matrix with elements given by equation (2.5). The matrix that discretizes the PDE in space is the ***differentiation matrix***

$$D = HB^{-1} \tag{3.6}$$

which was defined for a single derivative in section 2.9. The differentiation matrix may discretize a single space derivative or an entire differential operator.

The steady problem (3.1) is discretized as

$$Du = f \tag{3.7}$$

and has solution

$$u = D^{-1}f = BH^{-1}f. \tag{3.8}$$

The evaluation matrix H cannot be shown to always be invertible. In fact, examples have been constructed in which the evaluation matrix is singular [108]. Depending on the differential operator $\mathcal{L}$, the functions used to form the matrix H may not even be radial. Despite the lack of a firm theoretical underpinning, extensive computational evidence indicates that the

matrix H is very rarely singular and the asymmetric method has become well-established for steady problems.

The time-dependent problem (3.2) is discretized in space as

$$\frac{du}{dt} = Lu \approx Du. \tag{3.9}$$

Equation (3.2) is a system of ordinary differential equations (ODES) that is advanced in time with a numerical ODE method. Such a strategy is commonly referred to as a ***method of lines*** approach. Since B is invertible (2.3), the matrix D can always be formed and the MQ collocation method is well-posed for time-dependent PDEs.

3.1 Steady Problems

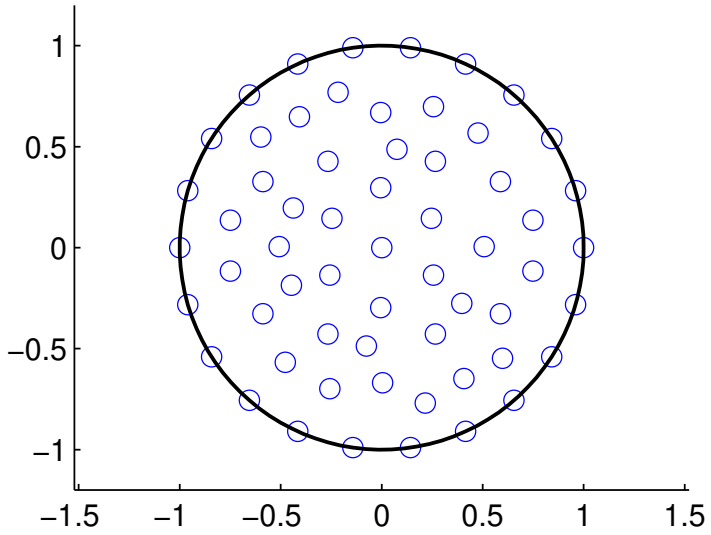

Figure 3.1: 60 centers on the unit circle consisting of 22 evenly space centers on the boundary and 38 near optional interior centers.

As a numerical example we consider the two-dimensional Poisson problem

$$u_{xx} + u_{yy} = f(x, y) \tag{3.10}$$

```
clear, home, close all
Nb = 22;       %  number of boundary points
Np = 60;       %  total number of center, interior and boundary

centers = dlmread('centersUnitCircle60.txt',' ');
x = centers(:,1);  y = centers(:,2);

u = 65./(65 + (x - 0.2).^2 + (y + 0.1).^ 2);                % exact solution

f = 130./(65 + (x-0.2).^2 + (y+0.1).^2).^3.*(2.*x-0.4).^2 - ...
    260./(65 + (x-0.2).^2 + (y+0.1).^2).^2 + ...
    130./(65 + (x-0.2).^2 + (y+0.1).^2).^3.*(2.*y+0.2).^ 2;

 H = zeros(Np,Np);  rx = zeros(Np,Np);  ry = zeros(Np,Np); r = zeros(Np,Np);

 f(1:Nb) = u(1:Nb);                      % Dirichlet Boundary conditions

 for i=1:Np
   for j=1:Np
      rx(i,j) = x(i) - x(j);
      ry(i,j) = y(i) - y(j);
      r(i,j) = sqrt( rx(i,j)^2 + ry(i,j)^2 );
   end
 end

index = 1;
for shape=1.0:-0.01:0.1;

      H(1:Nb,:) = mq(r(1:Nb,:),shape);   % enforce boudary conditions
      Hxx = mqDerivatives(r(Nb+1:Np,:),rx(Nb+1:Np,:),shape,2); % enforce PDE
      Hyy = mqDerivatives(r(Nb+1:Np,:),ry(Nb+1:Np,:),shape,2);
      H(Nb+1:Np,:) = Hxx + Hyy;      %   evaluation matrix

       kappa(index) = cond(H);

       warning off
       lambda = H\f;      % expansion coefficients via Gaussian elimination
       warning on

        B = mq(r,shape);   %  system matrix
       uh = B*lambda;      %  approximate solution

       er(index) = norm(u-uh,inf);
       sh(index) = shape;
           index = index + 1;
end

 semilogy(sh,er,'b'), xlabel 'shape parameter', ylabel 'max error'
 figure, semilogy(sh,kappa), xlabel 'shape parameter', ylabel '\kappa(H)'
```

Listing 3.1: poissonExampleMQ.m

on a unit circle domain, Ω. The function f in equation (3.10) is specified so that the exact solution is

$$u(x,y) = \frac{65}{65 + (x-0.2)^2 + (y+0.1)^2}. \tag{3.11}$$

Dirichlet boundary conditions are prescribed on the boundary, $\partial\Omega$, of the unit circle using the exact solution. Figure 3.1 shows the $N = 60$ center locations used in the example. The centers locations were determined by the near optimal algorithm of section 2.7. In the left image of figure 3.2 the error over a range of shape parameter is shown. For this example, the "optimal"

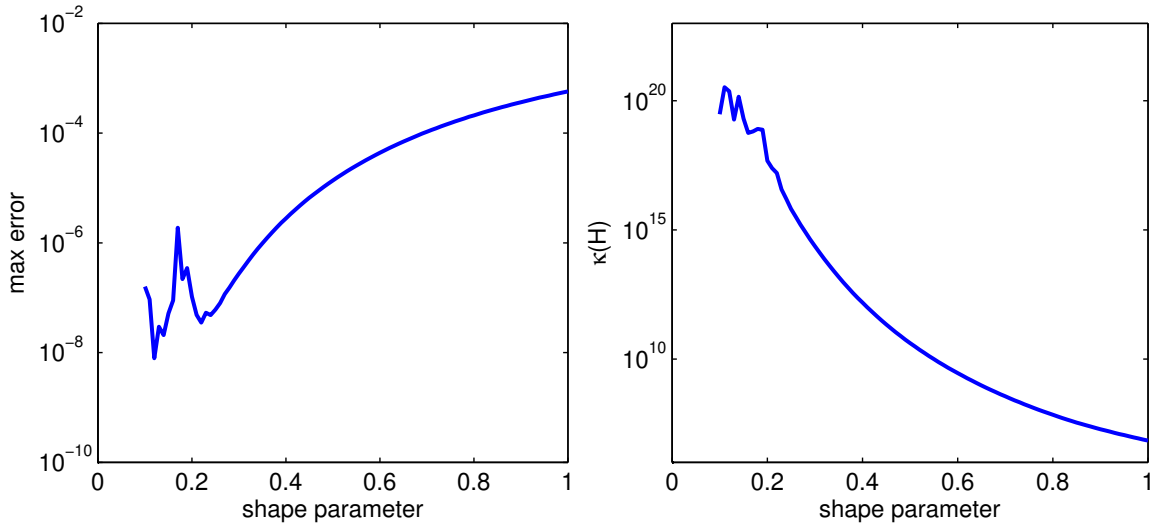

Figure 3.2: Left: Maximum error versus the shape parameter for problem (3.10). Right: Condition number of the system matrix from equation (3.10) versus the shape parameter.

value of the shape parameter is about $\varepsilon = 0.22$ at which the smallest error is reached. Matlab code that implements the example is in figure 3.1.

The implementation in listing 3.1 is a basic, straight forward implementation. Modifications can be made to the basic implementation for the purposes of improving accuracy, conditioning, and efficiency. Some of the possibilities include the following. The Contour-Padé algorithm (reference [129] and section 4.2) that can accurately evaluated the RBF approximant for small shape parameters by bypassing the ill-conditioned linear system if N is sufficiently small. Reference [124] suggests several strategies for mitigating the ill-conditioning problem for the MQ asymmetric collocation method for elliptic PDEs. The strategies include: domain decomposition (section 4.6), variable shape parameters (section 5.2), a truncated MQ basis, a multi-zone method, preconditioning (4.5.1), and adaptive center distributions. In reference [185] and section 5.8, integrated MQ RBFs are used to solve equation (3.10). Reference [53] and section 5.7 discusses the adaptive center location method of residual sub-sampling to solve boundary value problems. A greedy algorithm to select center locations is in section 4.7. A finite difference mode MQ method has been used to solve elliptic PDEs in reference [199] and section 5.6. Since errors are usually largest near boundaries, the authors in

reference [69] purpose a method that collocates both the boundary condition and the PDE at the boundary points. In line 37 of listing 3.1, the RBF expansion coefficients are solved for using Matlab's built-in version of Gaussian Elimination with scaled partial pivoting. Several algorithms have been suggested that may be a more robust way to solve the linear systems involved in the MQ method. The algorithms include, the truncated SVD solver (section 4.3.1), the improved truncated SVD solver (section 4.3.2), and an affine space method (section 4.4).

3.1.1 Neumann Boundary Conditions

In addition to Dirichlet boundary conditions the MQ collocation method easily handles, by modifying the boundary operator $\mathcal{B}$, Neumann (derivative) type boundary conditions as well as mixed types. As an example, we consider the Poisson problem (3.10) on the unit square. The domain is discretized with $N = 900$ uniformly spaced centers. The function f is set to $f(x, y) = -2(2y^3 - 3y^2 + 1) + 6(1 - x^2)(2y - 1)$ and Dirichlet boundary conditions of $u(0, y) = 2y^2 - 3y^2 + 1$ and $u(1, y) = 0$ are applied as well as Neumann boundary conditions of

$$\frac{\partial u}{\partial y} = 0, \quad \text{along} \quad y = 0 \text{ and } y = 1.$$

A shape parameter of $\varepsilon = 2.5$ results in an evaluation matrix to be inverted that has a very high condition number $\kappa(H) \approx 4.6e19$. Despite the large condition number, the maximum error of the approximation is 5.1e-5. The Matlab code in listing 3.2 carries out the example. As is the case with most computer code for implementing RBF methods, the program is relatively simple once the indices of the boundary and interior centers are identified. In this example the boundary centers are not located first in the array of centers as they were in the previous example. The indices of the centers where the Dirichlet and Neumann boundary conditions are to be applied and the indices of the interior centers where the PDE is applied are located using the Matlab function `find` on lines 13 through 15 of the listing.

3.1.2 Nonlinear Boundary Value Problems

The MQ collocation method is well-established for the solution of linear boundary value problems and hundreds of applications to this type of prob-

```
    shape = 2.5;
        N = 30;
       Np = N^2;
    [X,Y] = meshgrid(linspace(0,1,N),linspace(0,1,N));
        x = reshape(X,N^2,1);
        y = reshape(Y,N^2,1);

    f = -2*(2*y.^3 - 3*y.^2 + 1 ) + 6*(1 - x.^2).*(2*y - 1);
    u = (1 - x.^2).*(2*y.^3 - 3*y.^2 + 1 );                    % exact solution

    H = zeros(Np,Np);  rx = zeros(Np,Np);  ry = zeros(Np,Np);  r = zeros(Np,Np);

    dirichletBCs = find( x==0 | x==1 ); % identify boundary and interior centers
      neumannBCs = find( (y==0 | y==1) & ~(x==0 | x==1) );
        interior = find( x~=0 & x~=1 & y~=0 & y~=1 );

    f(dirichletBCs) = u(dirichletBCs);
      f(neumannBCs) = 0;

    for i=1:Np
      for j=1:Np
         rx(i,j) = x(i) - x(j);
         ry(i,j) = y(i) - y(j);
         r(i,j) = sqrt( rx(i,j)^2 + ry(i,j)^2 );
      end
    end

  H(dirichletBCs,:) = mq(r(dirichletBCs,:),shape);   %  evaluation matrix
    H(neumannBCs,:) = mqDerivatives(r(neumannBCs,:),ry(neumannBCs,:),shape,1);
                Hxx = mqDerivatives(r(interior,:),rx(interior,:),shape,2);
                Hyy = mqDerivatives(r(interior,:),ry(interior,:),shape,2);
      H(interior,:) = Hxx + Hyy;

      kappa = cond(H)
      alpha = H\f;
          B = mq(r,shape);    %   system matrix
         uh = B*alpha;
      error = norm(uh-u,inf)

    t = delaunay(x,y);  trisurf(t,x,y,abs(uh-u))
    xlabel 'x', ylabel 'y'   colormap('Summer')
```

Listing 3.2: poissonNeumannMQ.m

lem exist. Fewer examples of using the MQ method to solve nonlinear boundary values problems are available. Nevertheless, the MQ method is well suited for this type of problem as well. Applying the MQ method to nonlinear boundary value problems results in a nonlinear algebraic system to be solved. Several approaches have been successful in solving the nonlinear algebraic system. In [68] the authors used available software packages to solve the nonlinear algebraic systems. More often, some type of iterative method is used. In [21], a simple method which lagged the nonlinear term and then iterated was used. An Operator-Newton iterative algorithm that uses the MQ was developed in [64] and [60]. Further applications of the Operator-Newton method can be found in [19].

As an example, we consider the 1d nonlinear boundary value problem

$$u_{xx} + uu_x - u = f \tag{3.12}$$

on the interval $[0, 1]$. The vector f is specified so that $u(x) = x^2 e^x$ is the exact solution. Dirichlet boundary conditions of $u(0) = 0$ and $u(1) = e$ are applied. To linearize the equation the nonlinear term is lagged. That is, letting superscripts represent iteration numbers, we iterate

$$u_{xx}^{(n+1)} + u^{(n)} u_x^{(n+1)} - u^{(n+1)} = f, \qquad n = 0, \ldots, \text{until satisfied.} \tag{3.13}$$

At each iteration, a linear boundary value problem is solved by the MQ collocation method. The iteration can be stopped when the difference between successive iterations is below a specified tolerance. To start the method, we have taken $u^{(0)}$ to be the linear function that is fit between the the boundary points $(0, 1)$ and $(1, e)$.

The problem is discretized with $N = 40$ evenly spaced centers. A shape parameter of $\varepsilon = 3$ is used and a tolerance of $tol = 10\text{e-}8$ is specified. The method converges in 15 iterations and the maximum error is 2.0489e-6. The numerical solution and point-wise errors are shown in figure 3.3. The Matlab code for the example is in listing 3.3.

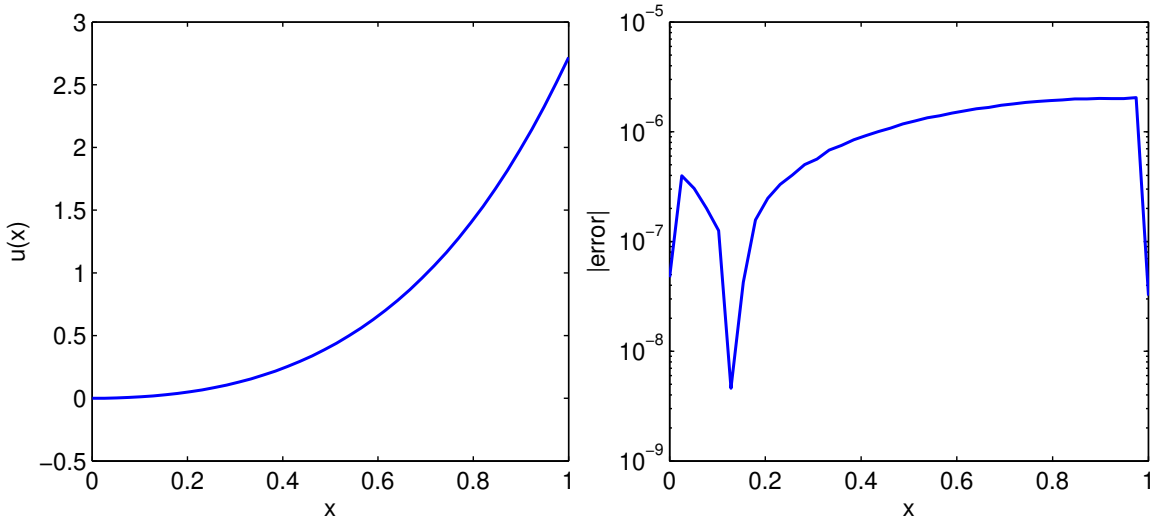

Figure 3.3: Solution of equation (3.12). Left: numerical solution. Right: point-wise errors.

```
      N = 40;
     xc = linspace(0,1,N)';
      o = ones(1,N);  rh = xc*o - (xc*o)';  r = abs(rh);   %  distance matrix
  shape = 3;

    f = 2*exp(xc) + 4*xc.*exp(xc) + (xc.^2).*exp(xc).*(2*xc.*exp(xc) ...
      + (xc.^2).*exp(xc));
   ex = (xc.^2).*exp(xc);            %  exact solution
    U = alpha + (beta - alpha)*xc;   % initial quess - linear fit between BCs

      alpha = 0; beta = exp(1); maxiter = 200; tol = 10e-8; iter = 0;  du = 1;

      plot(xc,U), pause

   f(1) = alpha;  f(N) = beta;          % Dirichlet bc at x=0 and x=1

      B = systemMatrix(xc,shape);

         H = zeros(N,N);
    H(1,:) = mq(r(1,:),shape);
H(2:N-1,:) = mqDerivatives(r(2:N-1,:),rh(2:N-1,:),shape,2) + ...
             diag(U(2:N-1))*mqDerivatives(r(2:N-1,:),rh(2:N-1,:),shape,1) - ...
             B(2:N-1,:);
    H(N,:) = mq(r(N,:),shape);

while ( norm(du,inf)> tol )

      lambda = H\f;
           u = B*lambda;
          du = norm(U-u,inf);
           U = u;

  H(2:N-1,:) = mqDerivatives(r(2:N-1,:),rh(2:N-1,:),shape,2) + ...
               diag(U(2:N-1))*mqDerivatives(r(2:N-1,:),rh(2:N-1,:),shape,1) ...
               - B(2:N-1,:);

     iter = iter+1;
     if iter==maxiter, disp('exceeded maximum iterations'), break;  end
   %  plot(xc,u,'b',xc,ex,'r'), pause  % uncomment to animate

end
plot(xc,u,'b',xc,ex,'r--')
```

Listing 3.3: nonlinearBvp1d.m

3.2 Time-Dependent PDEs

3.2.1 Method of Lines

After time-dependent PDEs are discretized in space with the MQ method, the remaining system of ODEs (3.2) is advanced in time with an ODE method using a method of lines approach.

A popular method in applications is the explicit, four-stage, fourth-order, Runge-Kutta (RK4) method

$$\begin{aligned} k_1 &= \Delta t F\left(u^n, t^n\right) \\ k_2 &= \Delta t F\left(u^n + 0.5k_1, t^n + 0.5\Delta t\right) \\ k_3 &= \Delta t F\left(u^n + 0.5k_2, t^n + 0.5\Delta t\right) \\ k_4 &= \Delta t F(u^n + k_3, t^n + \Delta t) \\ u^{n+1} &= u^n + \tfrac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right). \end{aligned} \tag{3.14}$$

Implicit methods such as the second-order accurate Trapezoid Rule

$$u^{n+1} = u^n + \frac{\Delta t}{2}\left[F(u^{n+1}) + F(u^n)\right] \tag{3.15}$$

may be used as well. The trapezoid rule was used in the first application of the MQ method to time-dependent PDEs in reference [118]. Using the trapezoid rule as a MOL integrator for time-dependent PDEs is sometimes referred to as a Crank-Nicolson approach.

Other numerical ODE methods that can be used to advance the system of ODEs in time can be found in standard references on numerical methods for ODEs such as reference [34].

3.2.2 Eigenvalue Stability

A rule of thumb is that the method of lines is stable if the eigenvalues of the discretized space operator, scaled by the time step Δt, lie in the stability region of the numerical ODE method. The stability region of the RK4 method (3.14) is shown in figure 3.9. The stability region of the Trapezoid rule (3.15) consists of the entire left half-plane of the complex plane and contains numbers with negative real parts.

In some instances, the MQ differentiation matrix may have eigenvalues with some, possibly large, positive real parts that prevent stable time integration. This is particulary the case in hyperbolic problems or advection-diffusion problems in which the advection term dominates. The phenomena is not well understood and has only recently started to be investigated [177, 187]. It is speculated that the eigenvalues with positive real parts are related to the application of boundary conditions. Experiments indicate that for a given N, the shape parameter ε can be chosen large enough so that all eigenvalues will lie in the left half-plane. However this requirement is rather restrictive, particulary for large values of N. In most cases ε has to be chosen so large that spectral convergence is compromised and in some cases the accuracy of the RBF methods are reduced to that of local finite difference methods. In [187] numerical experiments were performed on a group of time-dependent problems and bounds on the condition number of the system matrix were found that were necessary to form a differentiation matrix with eigenvalues with non-positive real parts. In reference [177], numerical experiments indicated that it may be better to base methods for time-dependent problems on the idea of least squares (section 2.11) rather than interpolation.

Nonlinear problems can be analyzed by linearizing the problem or a frozen coefficient approach. Reference [191] provides a more thorough analysis of the method of lines and RBF methods for nonlinear problems. The conclusion in [191] is that the method of lines approach leads to solvable ODE systems and small errors provided that the spatial discretization is fine enough and if the basis functions used in the spacial discretization are smooth enough.

3.2.3 Linear Advection-Diffusion Equations

```
function advectionDiffusionMQ()

    dt = 0.005;  finalTime = 0.5;
     a = 1;     nu = 0.002;           % advection and diffusion coefficients
  shape = 6;  N = 51;                % total collocation points
      x = linspace(0,1,N)';

      o = ones(1,length(x));
     rx = x*o - (x*o)';  % signed distance matrix
      r = abs(rx);       %  distance matrix

          H = zeros(N,N);   % evaluation matrix
     H(1,:) = mq(r(1,:),shape);        % Dirichlet BC at x=0
 H(2:N-1,:) = nu.*mqDerivatives(r(2:N-1,:),rx(2:N-1,:),shape,2) - ...
              a.*mqDerivatives(r(2:N-1,:),rx(2:N-1,:),shape,1);
     H(N,:) = mq(r(N,:),shape);         % Dirichlet BC at x=1

      B = mq(r,shape);      % system matrix
      dm = H/B;             % discretization D, of linear operator L

    U = exactSolution(x,0);  t=0;   % initial condition
    while t < finalTime
         u = rk4(U,t,dt,@F);
         t = t + dt;
      u(1) = 1;
      u(N) = exactSolution(1,t);
         U = u;
    end  % while

    exact = exactSolution(x,finalTime);
    subplot(1,2,1), plot(x,exact,'r',x,u,'b')
    subplot(1,2,2), plot(x,abs(U-exact))

   function fp = F(u,t)
       u(1) = 1;   u(N) = exactSolution(1,t); fp =  dm*u;
   end

   function ex = exactSolution(x,t)
          if t<dt
            if length(x)>1
              ex(1)=1; ex(2:length(x))=0; ex = ex(:);
            else, ex = 0; end
          else
             ce = 2.0*sqrt(nu*t); w1 = (x-a*t)/ce;  w2 = (x+a*t)/ce;
             ex = 0.5*(erfc(w1) + exp(a*x/nu).*erfc(w2));
          end
   end
end
```

Listing 3.4: advectionDiffusionMQ.m

The one-dimensional linear advection-diffusion equation is

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \quad \nu > 0. \tag{3.16}$$

As an example, the problem is approximated on a domain of $\Omega = [0, 1]$ with an initial condition of $u(x, 0) = 0$. For $t > 0$, a constant Dirichlet boundary condition of $u(0, t) = 1$ is applied and time-dependent Dirichlet boundary condition at $x = 1$ is applied using the exact solution

$$u(x,t) = \frac{1}{2}\left[\text{erfc}\left(\frac{x-t}{2\sqrt{\nu t}}\right) + \exp\left(\frac{x}{\nu}\right)\text{erfc}\left(\frac{x+t}{2\sqrt{\nu t}}\right)\right]. \tag{3.17}$$

Next, the problem is used to illustrate several options to advance the system (3.2) of ODEs in time.

Explicit Time Integration

In the example, $N = 51$ uniformly spaced centers were used with a shape parameter of $\varepsilon = 6$. The RK4 method with $\Delta t = 0.005$ was used to advance the problem to time $t = 0.5$. The maximum point-wise error at time $t = 0.5$ is 4.7e-4. The results from the example are shown in figure 3.4. The example is similar to the problem in reference [118] which was used to introduce the MQ collocation method for time-dependent PDEs and used to demonstrate the superiority of the MQ scheme over finite difference methods. Additionally, the MQ scheme was shown to be effective at large cell Reynolds numbers $Re_{\text{cell}} = (a\,\Delta x)/\nu$. The Matlab source code in listing 3.4 carries out the example. In this linear example, the entire differential operator can be discretized as $\mathcal{L} \approx a\frac{\partial u}{\partial x} + \nu\frac{\partial^2 u}{\partial x^2}$.

Implicit Time Integration

In this section we repeat the numerical experiment of the previous section that used RK4, but now the implicit Trapezoid rule is used for time integration. Discretizing equation (3.16) in space and then using the Trapezoid rule to approximate the time derivative results in

$$B\alpha^{n+1} = B\alpha_n + \frac{\Delta t}{2}\left[H\alpha^{n+1} + H\alpha^n\right] \tag{3.18}$$

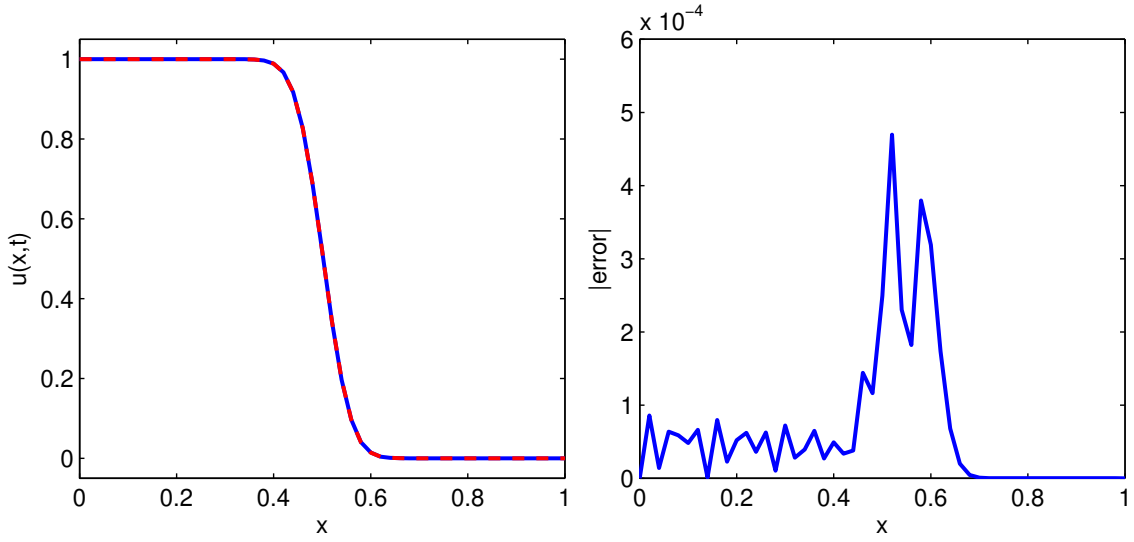

Figure 3.4: Left: MQ solution vs. exact solution of equation (3.16) at time $t = 0.5$ with $\Delta x = 0.02$, $a = 1$, $\nu = 0.002$, and cell Reynolds number $Re_{\text{cell}} = 10$. Right: Point-wise error from left image.

where $H = \nu H_{xx} - aH_x$ and the notation α^n denotes the MQ expansion coefficients at time level t^n. Equation (3.18) is easily manipulated into the form

$$\left(B - \frac{\Delta t}{2}H\right)\alpha^{n+1} = \left(B + \frac{\Delta t}{2}H\right)\alpha^n. \tag{3.19}$$

Now let

$$T_L = \left(B - \frac{\Delta t}{2}H\right)$$

and

$$T_R = \left(B + \frac{\Delta t}{2}H\right).$$

If we assume that T_L is non-singular and let $T = T_L^{-1}T_R$, then the MQ expansion coefficients of the PDE solution at the time level t^{n+1} are given by

$$\alpha^{n+1} = T\alpha^n.$$

Recalling that $\alpha^n = B^{-1}u^n$, the approximate PDE solution at t^{n+1} level is given by

$$u^{n+1} = B\alpha^{n+1}$$

$$= \quad B\,T\,B^{-1}u^n$$

or by letting $A_T = B\,T_L^{-1}\,T_R\,B^{-1}$,

$$u^{n+1} = A_T\,u^n. \tag{3.20}$$

Note that in the code in listing 3.5, that the two matrix inverses in the formula for A_T are never explicitly formed. After a more expensive set up procedure, the MQ-Trapezoid method needs less than one-fourth of the flops (floating point operations) to advance the solution one time step than does the MQ-RK4 method.

The Matlab code in listing 3.5 repeats the numerical example of the previous section, except that the trapezoid rule is used for time integration rather than RK4. The results from the example are shown in figure 3.5. We use $N = 51$ uniformly spaced centers and shape parameter of $\varepsilon = 6$. A time step of size $\Delta t = 0.002$ was used to advance the problem to time $t = 0.5$. The maximum point-wise error at time $t = 0.5$ is 7.7e-3.

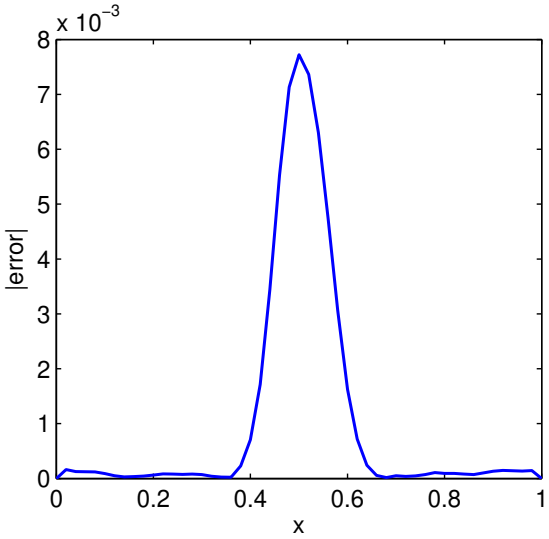

Figure 3.5: Point-wise error from the MQ-Trapezoid solution of equation (3.16) at time $t = 0.5$ with $\Delta x = 0.02$, $a = 1$, $\nu = 0.002$, and $\Delta t = 0.002$.

```
function A = advectionDiffusionMqTrapezoid()

     dt = 0.002;  finalTime = 0.5;
      a = 1;                              % advection coefficient
     nu = 0.002;                          % diffusion coefficient
  shape = 7;    N = 51;

     x = linspace(0,1,N)'; o = ones(1,length(x));
    rx = x*o - (x*o)';     r = abs(rx);

     U = exactSolution(x,0);
     H = nu.*mqDerivatives(r,rx,shape,2) - a.*mqDerivatives(r,rx,shape,1);
     B = mq(r,shape);

     TL = B - 0.5*dt*H;
     TR = B + 0.5*dt*H;
      T = TL\TR;
      A = B*(T/B);

     A(1,:) = 0;  A(end,:) = 0;
     A(1,1) = 1;  A(N,N) = 1;

   t=0;
   while t < finalTime
           u = A*U;
        u(N) = exactSolution(1,t);
           t = t + dt;
           U = u;
   end

   exact = exactSolution(x,finalTime);
   subplot(1,2,1), plot(x,exact,'r--',x,u,'b')
   subplot(1,2,2), plot(x,abs(U-exact))

  function ex = exactSolution(x,t)
          if t<dt
            if length(x)>1
              ex(1)=1; ex(2:length(x))=0; ex = ex(:);
            else, ex = 0; end
          else
             ce = 2.0*sqrt(nu*t); w1 = (x-a*t)/ce; w2 = (x+a*t)/ce;
             ex = 0.5*(erfc(w1) + exp(a*x/nu).*erfc(w2));
          end
  end
end
```

Listing 3.5: advectionDiffusionMqTrapezoid.m

Laplace Transform Time Integration

In addition to the method of lines approach, the Laplace Transform MQ (LTMQ) method has been suggested [167] as a very accurate method to advance linear PDEs in time. If the final time is large, and the PDE is linear or linearized, then it is possible to advance the PDE in time with less computational time than required by the MOL approach by means of the Laplace transform and inverse Laplace transform method. The Laplace transform is exact, thus the time integration scheme does not have truncation errors. However, the Laplace transform is not exactly evaluated and there will be some numerical error depending on the method used to invert the transform. The LTMQ method only computes the solution at the final time level and does not compute the solution at intermediate time levels as in

a time marching approach as the Method of Lines. Reference [167] can be consulted for the details of the method.

3.2.4 Nonlinear Equations

```
1
2  function vBurgersMQ()
3
4      nu = 5*0.000875; dt = 0.001;  finalTime = 1.2;
5       N = 140;  shape = 11;
6       x = linspace(-1,1,N)';
7       o = ones(1,length(x));
8       rx = x*o - (x*o)';  % signed distance matrix
9        r = abs(rx);       % distance matrix
10
11      Hx = mqDerivatives(r,rx,shape,1);
12     Hxx = mqDerivatives(r,rx,shape,2);
13
14         B = mq(r,shape);
15        dm = Hx/B;
16       dm2 = Hxx/B;
17
18     U = exactSol(x,0);  t=0;
19     while t < finalTime
20          u = rk4(U,t,dt,@F);
21          t = t + dt;
22          u(1) = exactSol(-1,t); u(N) = exactSol(1,t);  U = u;
23     end  % while
24
25    exact = exactSol(x,finalTime);
26    subplot(1,2,1), plot(x,exact,'r',x,u,'b')
27    subplot(1,2,2), plot(x,abs(U-exact))
28
29 % ------------ nested functions ----------------------------------
30
31  function ex = exactSol(x,t)
32    x2 = x+1.0; aa = 0.05*(x2 - 0.5 + 4.95*t)/nu;
33    bb = 0.25*(x2 - 0.5 + 0.75*t)/nu;  cc = 0.5*(x2-0.375)/nu;
34    ex = (0.1*exp(-aa)+0.5*exp(-bb)+exp(-cc))./(exp(-aa)+exp(-bb)+exp(-cc));
35  end
36
37   function fp = F(u,t)
38        u(1) = exactSol(-1,t); u(end) = exactSol(1,t);
39          fp =  -0.5*dm*u.^2 + nu*dm2*u;
40   end
41 % --------------------------------------
42 end
```

Listing 3.6: vBurgersMQ.m

Nonlinear time-dependent problems often require a slightly different approach than linear problems. Linear problems allow the entire differential operator to be discretized as a single matrix D and the entire space discretization can be evaluated by a single matrix multiplication. For all but the simplest nonlinear problems, this is not the case. For example, to evaluate the spatial discretization of Burgers equation

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{u^2}{2}\right) = \nu \frac{\partial^2 u}{\partial x^2}, \tag{3.21}$$

both the first and second order differentiation matrices must be formed. Then the space discretization is $-\frac{1}{2}D_1u^2 + \nu D_2 u$.

Listing 3.6 contains Matlab code that solves equation (3.21) on the interval $[-1, 1]$ with the initial condition and boundary conditions $u(-1, t) = g_l(t)$ and $u(1, t) = g_r(t)$ being taken from the exact solution. The exact solution is

$$u(x, t) = \frac{0.1e^a + 0.5e^b + e^c}{e^a + e^b + e^c} \tag{3.22}$$

where $a = -(x+0.5+4.95t)/(2\nu)$, $b = -(x+0.5)/(4\nu)$, and $c = -(x+0.625+0.75t)/(2\nu)$. The solution features two sharp fronts advancing left to right through the domain. The two fronts merge together with advancing time. With $\nu = 0.004375$, the example is discretized in space with the MQ method using $N = 140$ equally spaced centers and a shape parameter of $\varepsilon = 11$. The method is advanced in time with RK4 to time $t = 1.2$. The results are displayed in figure 3.6. In section 5.7, adaptive center MQ methods are

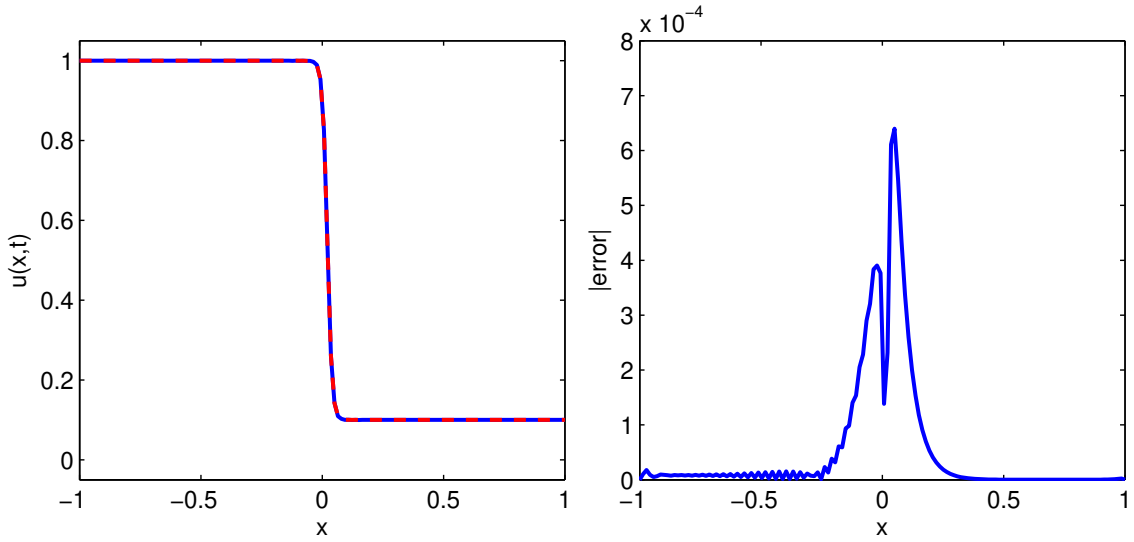

Figure 3.6: Left: MQ solution versus exact of equation (3.21) at time $t = 1.2$ with $N = 140$, $\nu = 0.004375$. Right: Point-wise error from left image.

used to solve the example with a smaller viscosity coefficient, and with fewer centers.

Another nonlinear example is the Cubic Schrodinger Equation initial

value problem

$$i\frac{\partial u}{\partial t} + \frac{\partial^2 u}{\partial x^2} + 8\,|\,u|^2\,u = 0 \tag{3.23}$$

with the initial condition $u(x,0) = sech(x)$. The problem is defined on the real line, but numerically the domain is truncated to a finite interval, $\Omega = [-10, 10]$, that is large enough so that the exponentially decaying solution does not reach the boundary. The problem has an exact ***breather solution*** $u(x,t) = a + bi$ where

$$a = \frac{\cosh(x)\left[4\cos(t)\cosh^2(x) + 3\cos(t)\cos(8t) - 3\cos(t) - 3\sin(t)\sin(8t)\right]}{4\cosh^4(x) - 3 + 3\cos^2(4t)}$$

and

$$b = \frac{\cosh(x)\left[3\cos(t)\sin(8t) + 4\sin(t)\cosh^2(x) + 3\sin(t)\cos(8t) - 3\sin(t)\right]}{4\cosh^4(x) - 3 + 3\cos^2(4t)}.$$

A breather is a localized periodic solution with a soliton structure. A soliton is defined to have three properties: they are of permanent form, they are localized within a region, and they can interact with other solitons and emerge from the collision unchanged except for a phase shift. There are two types of breathers: standing or traveling. Standing breathers correspond to localized solutions whose amplitude vary in time. In this problem the solution is a standing breather. The problem is discretized in space with the MQ method with a relatively small number of $N = 300$ evenly spaced centers on the large interval. A shape parameter of $\varepsilon = 2$ is used which results in a system matrix with $\kappa(B) = 2.0e14$. The problem is advanced in time with the fourth-order Runge-Kutta method (3.14) to time $t = 20$. The m-file in listing 3.7 carries out the example and animates the solution as it advances in time. The numerical solution and point-wise errors for the example are in figure 3.7.

3.2.5 Higher Dimensions

One of the most favorable feature of the MQ RBF method is that implementing the method in higher dimensions is nearly as simple as implementing the method in one dimension. To illustrate, a two-dimensional problem is solved in a complexly shaped domain. The two-dimensional heat equation is

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2}. \tag{3.24}$$

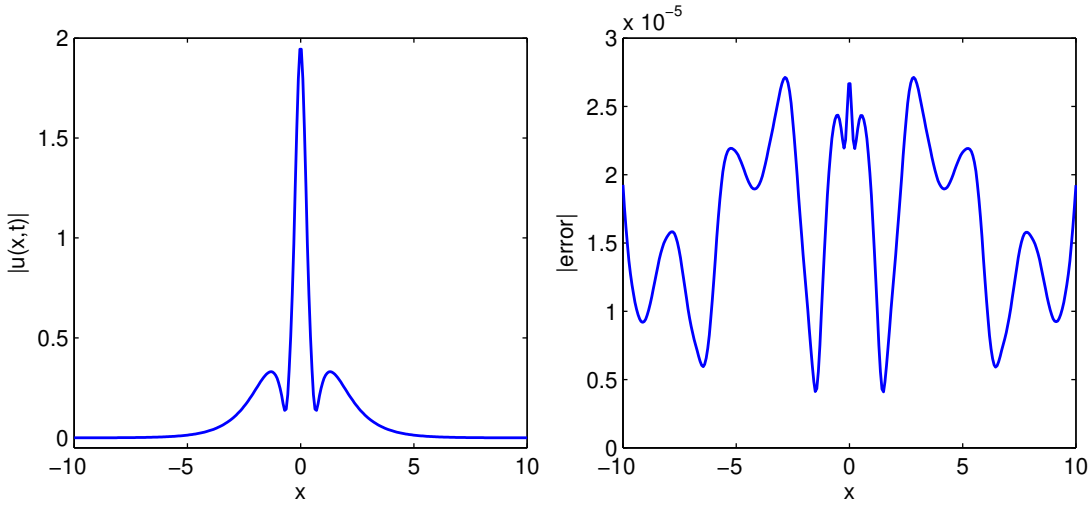

Figure 3.7: Left: Cubic Schrodinger equation (3.23) solution at time $t = 20$. Right: Point-wise error from left image.

The domain and center locations are shown in figure 3.8. In polar coordinates, the outer boundary is

$$r(\theta) = 1 + \frac{1}{5}\cos(\theta) + \frac{3}{20}\sin(4\theta)$$

and the inner boundary is

$$r(\theta) = \frac{3}{10} + \frac{1}{10}\sin(\theta) + \frac{3}{20}\sin(5\theta).$$

The center locations were produced by the near-optimal algorithm of section 2.7. The initial condition and time-dependent Dirichlet boundary conditions are specified using the exact solution

$$u(x, y, t) = e^{-5t\pi^2}\sin(\pi x)\sin(2\pi y).$$

A shape parameter of $\varepsilon = 2$ is used which results in the condition number of the system matrix being $\kappa(B) = 2.5\text{e}15$. The problem is advanced in time to $t = 0.1$ at which time a maximum error of 3.7527e-6 is recorded. The Matlab code that carries out the example is in listing 3.8. Comparing the Matlab code for this example with the Matlab code in listing 3.4 for the advection-diffusion equation in one-dimension illustrates how the MQ

```
function cubicSchrodingerMQ()

   A = -10; B = 10;
            dt = 0.0001;
    finalTime = 20;
             N = 300;  shape = 2;
             x = linspace(A,B,N)';

       o = ones(1,N);
      rx = x*o - (x*o)';   r = abs(rx);
     Hxx = mqDerivatives(r,rx,shape,2);

        B = mq(r,shape);
      dm2 = Hxx/B;

        U = sech(x);
       hh = plot(x,abs(U),'erasemode','xor');
       set(hh(1),'Color',[0 0 1]);    % numerical in blue
       axis([-10 10 -0.05 2]), xlabel('x'), ylabel('u(x,t)'), pause

    n=0; t=0;
    while n*dt < finalTime

          u = rk4(U,t,dt,@F);
          t = t + dt;  n = n+1;  U = u;

          set(hh(1),'ydata',abs(u)),  drawnow
    end


  function fp = F(u,t)
    fp = ( -dm2*u - 8*((abs(u)).^2).*u )/i;
  end

end
```

Listing 3.7: cubicSchrodingerMQ.m

method is essentially the same in $d > 1$ dimensions as it is in one dimension. The only differences in the computer code of the one dimensional problem with equally spaced centers and the computer code of the two dimensional problem on a irregularly shape domain is some bookkeeping to keep track of which centers are located in the interior of the domain and which centers are located on the boundary.

3.2.6 Hyperbolic PDEs

For any type of numerical method, Hyperbolic PDEs are the most challenging. Hyperbolic PDEs can be very sensitive to how boundary conditions are applied. For the MQ collocation method, eigenvalue stability can be a challenge as discussed in section 3.2.2. Nonlinear hyperbolic problems may develop discontinuities (shocks) even if the initial conditions are smooth. The MQ method is affected by the well-known Gibbs phenomenon when the underlying function being approximated has discontinuities. Techniques to alleviate the effects of the Gibbs phenomenon are in section 5.5.

```
function  heatComplexDomain()

centers = dlmread('centers.txt',' ');
      x = centers(:,1);  y = centers(:,2);  N = length(x);
    bpi = dlmread('centersBoundaryIndex.txt',' ');
     NB = length(bpi); NI = N - NB;

  xb = x(bpi);  yb = y(bpi); % re-order centers to put boundary centers last
 ipi = NB+1:1:N; xi = x(ipi); yi = y(ipi);  x = [xi; xb];   y = [yi; yb];

   o = ones(1,length(x));
  rx = (x*o - (x*o)');
  ry = (y*o - (y*o)');
   r = sqrt( rx.^2 + ry.^2 );

shape = 2;  dt = 0.0005;
H = mqDerivatives(r,rx,shape,2) + mqDerivatives(r,ry,shape,2); % Evaluation matrix
B = mq(r,shape);                                               % System matrix
dm = H/B;                                  % discretization D, of linear operator L

       finalTime = 0.1;
       U = exactSolution(x,y,0);                    % initial condition
       td = delaunay(x,y);  hh = trisurf(td,x,y,U);
       bpi = NI+1:N;                               % boundary points must be last

    pause, t = 0;
    while t < finalTime
       u = rk4(U,t,dt,@F);
       t = t + dt;
       u(bpi) = exactSolution(x(bpi),y(bpi),t);
       U = u;
       hh = trisurf(td,x,y,U);  drawnow;
   end  % while

xlabel 'x', ylabel 'y', pause
hh = trisurf(td,x,y,abs(U-exactSolution(x,y,finalTime)));
er = max( abs(U-exactSolution(x,y,finalTime)) ), xlabel 'x', ylabel 'y', drawnow

  function fp = F(u,t)
         u(bpi) = exp(-5*t*pi^2).*sin(x(bpi)*pi).*sin(2*pi*y(bpi));
         fp = dm*u;
  end

  function e =  exactSolution(x,y,t)
    e = exp(-5*t*pi^2).*sin(x*pi).*sin(2*pi*y);
  end

end
```

Listing 3.8: heatComplexDomain.m

The advection equation in one space dimension with appropriately specified initial and boundary conditions is

$$\frac{\partial u}{\partial t} - \frac{\partial u}{\partial x} = 0, \qquad -1 \le x \le 1, \quad t > 0 \tag{3.25}$$

$$u(1,t) = 0, \qquad u(x,0) = u_0(x). \tag{3.26}$$

The MQ method is used to discretize the spatial derivative in problem (3.25) with $N = 40$ equally spaced centers. The zero Dirichlet boundary condition at $x = 1$ is enforced by setting the last row of the differentiation matrix to zero (line 17 of listing 3.9). The eigenvalues of the discretized operator, scaled by $\Delta t = 0.04$, are shown in figure 3.9 in relation to the RK4 stability region.

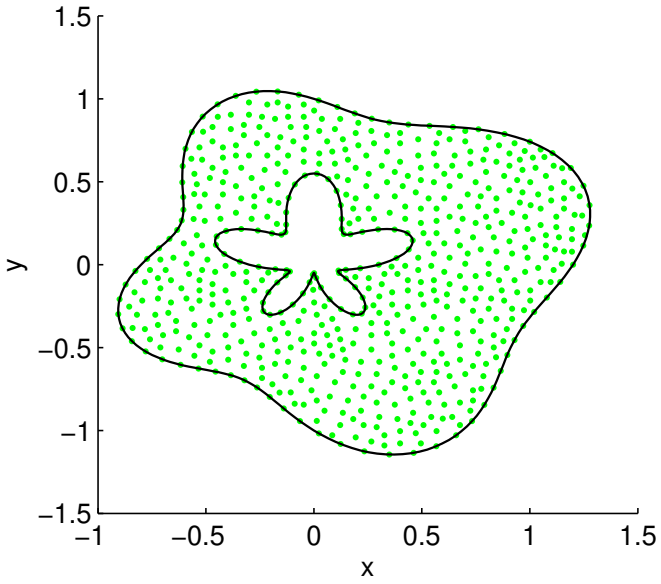

Figure 3.8: Computational domain and center locations, 130 boundary centers and 505 interior centers, for problem (3.24).

The left image corresponds to a shape parameter of $\varepsilon = 2$ and a system matrix with condition number $\kappa(B) = 2.9\text{e}15$. Two of the eigenvalues have positive real parts that are outside the stability region. For this example, it was necessary to raise the shape parameter to $\varepsilon = 4$ in order to get the scaled eigenvalues to lie within the stability region. At this value of the shape parameter the system matrix has a condition number of $\kappa(B) = 1.2\text{e}9$. The same problem is considered in section 5.2 using a nonconstant shape parameter.

3.3 Chapter Summary

The MQ asymmetric collocation method is well established as a method for the numerical solution of steady PDE boundary value problems. The solution procedure simply involves solving a linear system of equations if the PDE is linear. Nonlinear steady problems are solved by iterative methods in which a linear problem is solved in each iteration. Although examples may be constructed in which the evaluation matrix of the linear steady problem

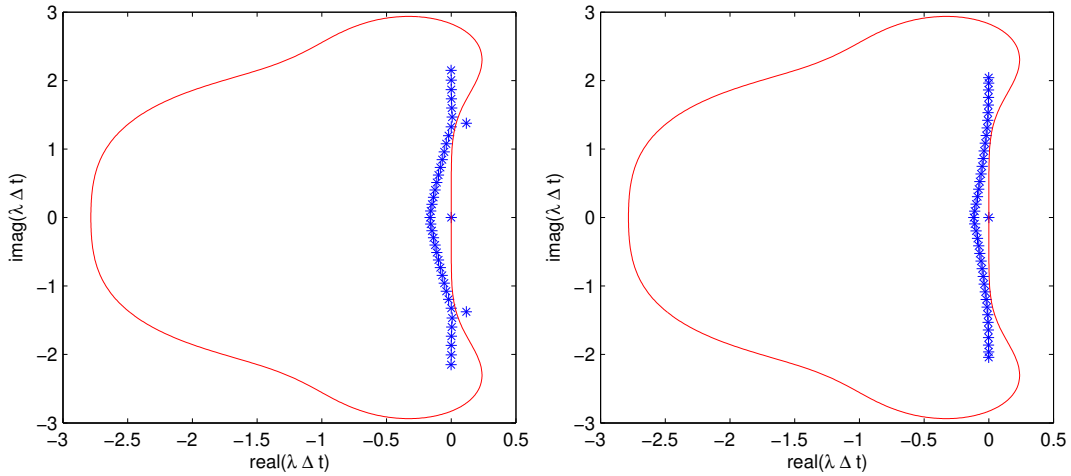

Figure 3.9: RK4 stability region and the scaled eigenvalues of the discretized 1d advection problem (3.25). Left: unstable spectrum with $\varepsilon = 2$ and $\kappa(B) =$ 2.9e15. Right: stable spectrum with $\varepsilon = 4$ and $\kappa(B) = 1.2$e9.

is singular, an extensive amount of numerical evidence from a large number of applications indicates that in applications the matrix is rarely, if ever, singular.

Time dependent PDEs are discretized in space by the MQ method and then are typically advanced in time with a method of lines approach. Eigenvalue stability for hyperbolic problems remains a challenge and is an active research topic.

```
function advectionMQ1d()

        N = 40;
    shape = 4;
       dt = 0.001;
finalTime = 1;

  x = linspace(-1,1,N)';
  o = ones(1,length(x));
 rx = x*o - (x*o)';  r = abs(rx);

 H = mqDerivatives(r,rx,shape,1);   % evaluation matrix H
 B = mq(r,shape);                   % system matrix B
dm = H/B;          % discretization D, of linear operator L
 K = cond(B); disp('K(B): '), disp(K)

 dm(end,:)=0;  % zero Dirichlet BC at x=1

 U = exactSolution(x,0);   exact = exactSolution(x,0);
 hh = plot(x,U,x,exact,'erasemode','xor');
 set(hh(1),'Color',[0 0 1]);     % numerical in blue
 set(hh(2),'Color',[1 0 0]);     % exact in red
 axis([-1 1 -0.05 1.05]), pause

 t = 0;
 while t < finalTime

    u = rk4(U,t,dt,@F);
    t = t + dt;
    U = u;

   exact = exactSolution(x,t);
   set(hh(1),'ydata',u,'Color',[0 0 1]), set(hh(2),'ydata',...
       exact,'Color',[1 0 0],'LineStyle','--'), drawnow

   end
   eInf = norm(U-exact,inf)

   function fp = F(u,t)
         fp =  dm*u;
   end

   function ex = exactSolution(x,t)
       ex = exp( -40*( (x-0.4 + t).^2) );
   end

end
```

Listing 3.9: advectionMQ1d.m

Chapter 4

Large N - small ε - small q_Ξ

The MQ method with a large number of N centers, and/or with a small shape parameter ε, and/or with a small minimum fill distance q_Ξ (2.9), may face numerical implementation difficulties. All three issues may be encountered in the same problem. Looking at each issue separately, we see that when using a large number of centers the numerical difficulty is efficiency. The number of floating point operations needed to solve a linear system with Gaussian Elimination is $\mathcal{O}(N^3)$. The $\mathcal{O}(N^3)$ flop count is very computationally expensive for large N. This issue can be addressed with domain decomposition methods (section 4.6) and with iterative methods (section 4.5) that use preconditioning (section 4.5.1) techniques. Additionally, a greedy algorithm (section 4.7) can be used to reduce the number of centers. A small shape parameter ε and a small minimum fill distance are both desirable for accuracy, but both cause conditioning problems. The conditioning problem can be addressed with extended numerical precision (section 4.1), or by evaluating the MQ method in a way that bypasses the ill-conditioned linear systems (section 4.2), or by using algorithms other than Gaussian elimination to solve the linear systems (sections 4.3 and 4.4).

In many of the following sections, the Poisson equation

$$\begin{aligned} u_{xx} + u_{yy} &= (\lambda^2 + \mu^2) e^{(\lambda x + \mu y)}, \qquad (x, y) \in \Omega \\ u(x, y) &= e^{(\lambda x + \mu y)}, \qquad (x, y) \in \partial\Omega \end{aligned} \tag{4.1}$$

is considered as an example with $\lambda = 0.3$ and $\mu = 0.8$. The exact solution is $u(x, y) = e^{(\lambda x + \mu y)}$. The domain Ω is taken to be the quarter circle in the left image of figure 2.2. The $N = 618$ centers used for the problem are shown in the left image of figure 2.2.

4.1 Extended precision

The effects of poor conditioning in finite precision computer arithmetic can be mitigated by using more digits of precision. The number of significant digits in standard precision IEEE 64-bit floating-point arithmetic is dictated by hardware architecture. Standard precision results in approximately 16 decimal digit accuracy. Expended precision involves software and computer memory and results in considerably longer computations.

To illustrate the costs and benefits of extended precision computer arithmetic, we revisit the Poisson problem (3.10) which was solved using the 60 scattered centers in figure 3.2. This problem was originally solved in section 3.1 using standard precision. The solution was computed 90 times over a range of shape parameters and took 0.41 seconds on a computer representative of desktop technology in the year 2009. The extended precision example was implemented using the variable-precision arithmetic (vpa) package from the Matlab version 2007a symbolic toolbox. Solving the problem (***poissonExtendedPrecision.m***) 5 times with 50 digits of precision over a set of small values of ε took approximately 8 minutes of computer time. The extended precision results (figure 4.1) show about 4 additional decimal places of accuracy at small shape parameters when compared to standard precision.

In reference [113], numerical experiments were performed using extended precision with 100-digit accuracy to gain some insight into the connection between the accuracy of the MQ method, the fill distance h, and the shape parameter. The authors determined that for a given h, an optimal value of the shape parameter c exists whose value should not be decreased unless h is refined. Once that value of the shape parameter is reached, further reduction of the error can be accomplished only by decreasing h.

Also in [113], the authors concluded that in order to achieve optimal accuracy and efficiency in solving elliptic boundary value problems, it is better to use a relatively coarse grid and extended precision than standard precision and a fine grid. Using extended precision, the number of centers needed to obtain a desired accuracy was significantly smaller than when using standard precision. Thus in our previous example, we should be able to match the smallest error from standard precision calculations over a range of shape parameters with an extended precision calculation using $M \ll 60$ centers. Additionally, instead of using packages such as Matlab, the authors recommended more efficient programs developed in C++ for extended precision calculations. Extended precision algorithms will only be useful in applica-

tions if they can be efficiently implemented so that better accuracy can be realized using significantly smaller N and with less CPU time than if standard precision precision had been used.

RBF methods are not alone among areas in scientific computing that benefit from extended precision floating arithmetic. A growing number of such areas, as well as some extended precision computing environments, are surveyed in references [6] and [7].

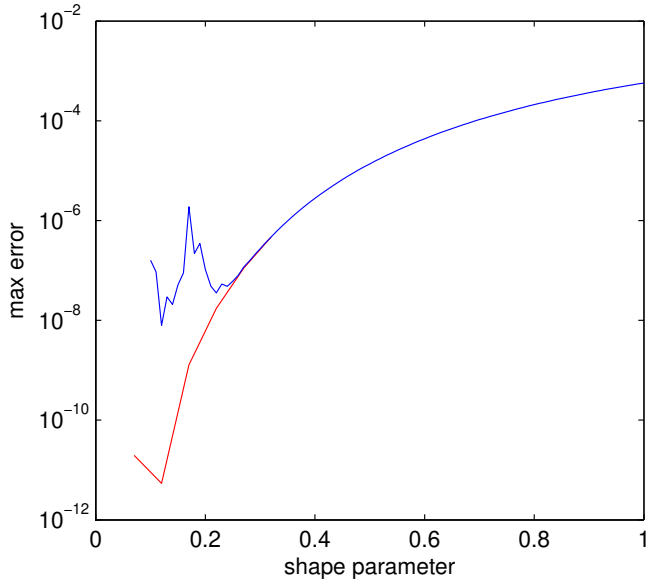

Figure 4.1: Standard precision (blue) and extended 50 digit precision (red) maximum error versus the shape parameter.

4.2 Contour-Padé algorithm

In references [211] and [81], an algorithm called the Contour-Padé algorithm is described for evaluating RBF approximation methods. The algorithm avoids working directly with the associated ill-conditioned linear systems. The algorithm stably calculates the RBF approximant for small values of the shape parameter ε that cannot be handled by direct methods. The method has proven useful in exploring the theoretically known, but previously unreachable, accuracy at small values of the shape parameter. The use of the

algorithm in applications is severely limited by the restriction that it only works with a small number of centers.

The motivation for the algorithm starts by considering the use of complex values of the shape parameters, $\varepsilon = a + bi$. Figure 4.2 shows the condition numbers of the system matrices resulting from using shape parameters from a square domain centered around the origin in the complex plane. Near $\varepsilon = 0$ the condition numbers approach infinity, but in floating point arithmetic they remain bounded by approximately 10^{21}. In figure 4.2, there are several sharp spikes that correspond to complex shape values where the system B is singular. None of these can occur on the real axis as a result of the theorem in section 2.3.

In a large region around $\varepsilon = 0$, the RBF interpolant is a meromorphic (its only singularities are poles) function [211] and therefore at a single center x_0^c can be written as

$$s(x_0^c, \varepsilon) = (\text{rational function of } \varepsilon) + (\text{power series in } \varepsilon). \tag{4.2}$$

If a circular contour around $\varepsilon = 0$ can be found that avoids any branch points of the RBF and where B is well-conditioned so that the RBF interpolant can be evaluated by direct methods, the Contour-Padé algorithm can stably determine the expansion coefficients in (4.2). Then the interpolant can be evaluated for all $\varepsilon \leq \rho$ where ρ is the radius of the circle. If there are any poles inside the circle, the algorithm will have to compensate for the poles. The details of the algorithm can be found in [81].

As an example, the Contour-Padé algorithm is used to approximate the first order partial derivative with respect to x of the function $f(x, y) = \exp(x/2+y/4)$ using the 60 scattered centers in figure 4.3. The approximation is evaluated at the center that is located at approximately (0.7487,-0.1156) and is marked with a red asterisk in figure. On the unit circle the maximum possible distance between two points is 2, thus there will not be any branch points closer than $\pm i/2$, so we must have $\rho < 0.5$. This is what limits the algorithm to small N. However with only 60 centers a suitable circle is located at $\rho = 0.35$. At the values of the shape parameter on the circle, the system matrix has condition numbers of $\mathcal{O}(10e13)$ and the linear system can be accurately evaluated using a direct method. Then the Contour-Padé algorithm can evaluate the approximation over the epsilon range $0 \leq \varepsilon \leq 0.35$ at the center. The resulting error is show as a solid blue line in figure 4.4. The results of the direct method using Gaussian Elimination are given for

comparison. The Contour-Padé algorithm reveals the true optimal shape parameter of $\varepsilon = 0.12$ for this calculation. The results are typical in that the Contour-Padé algorithm usually shows that the optimal value of the shape parameter is at some small, but nonzero value, at which direct methods are unable to accurately resolve the linear systems involved.

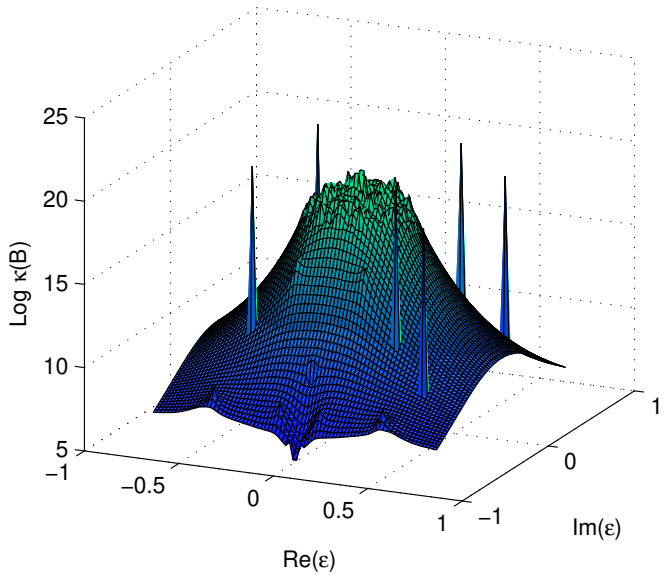

Figure 4.2: The condition number of the MQ interpolation matrix with complex valued shape parameters. The z axis is on a log scale.

4.3 SVD based methods

The singular value decomposition [203] (SVD) of an invertible matrix $B \in \mathcal{R}^{N\times N}$ is

$$B = U\Sigma V^T. \tag{4.3}$$

The matrices V are $N \times N$ orthogonal matrices and the $N \times N$ diagonal matrix is $\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_N)$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_N > 0$. The σ_i's are called the singular values of B. The columns of U and V are the left and right singular vectors of B respectively. The inverse of B via the SVD

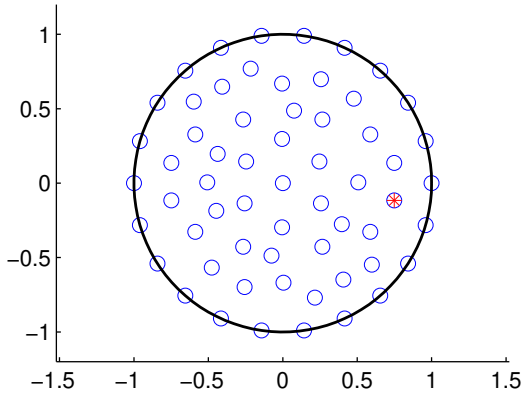

Figure 4.3: Left: The Contour-Padé algorithm evaluates $\partial f/\partial x$ at the center marked with the red asterisk.

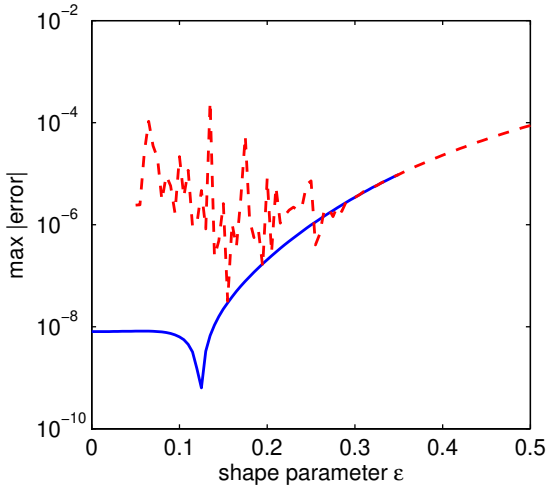

Figure 4.4: Gaussian Elimination (red dashed) and Contour-Padé (blue solid) error versus the shape parameter for approximating $\partial f/\partial x$ of function $f(x, y) = \exp(x/2 + y/4)$ at the point (0.7487,-0.1156).

is $B^{-1} = V\Sigma^{-1}U^T$ and the SVD can be used to solve a linear system, such as (2.4), as

$$\alpha = V\Sigma^{-1}U^T f.$$

The operation count for the solving a linear system with the SVD is higher than that of Gaussian Elimination, but it has numerical rank revealing properties [203] that can be useful when dealing with ill-conditioned matrices.

As an example, the Poisson problem (4.1) is solved and compared to results obtained by solving the RBF system by Gaussian Elimination. The results are illustrated in figure 4.5. The SVD solution is not qualitatively different than the Gaussian elimination solution. However, the SVD method can be modified so that it performs better with very poorly conditioned systems.

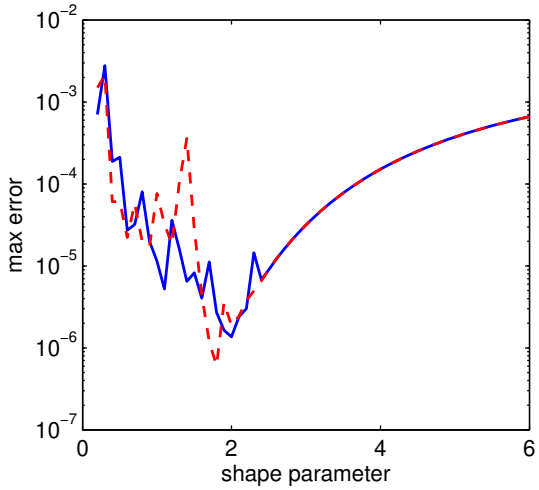

Figure 4.5: SVD results from problem (4.1). Accuracy versus the shape parameter. Gaussian Elimination (solid blue) and the SVD (dashed red).

4.3.1 Truncated SVD (TSVD)

The SVD solution of an ill-conditioned system is dominated by the contributions of very small singular values which may not have been computed accurately. To lessen this problem, a regularization parameter μ can be introduced and contributions from singular values with $\sigma_i < \mu$ disregarded. If

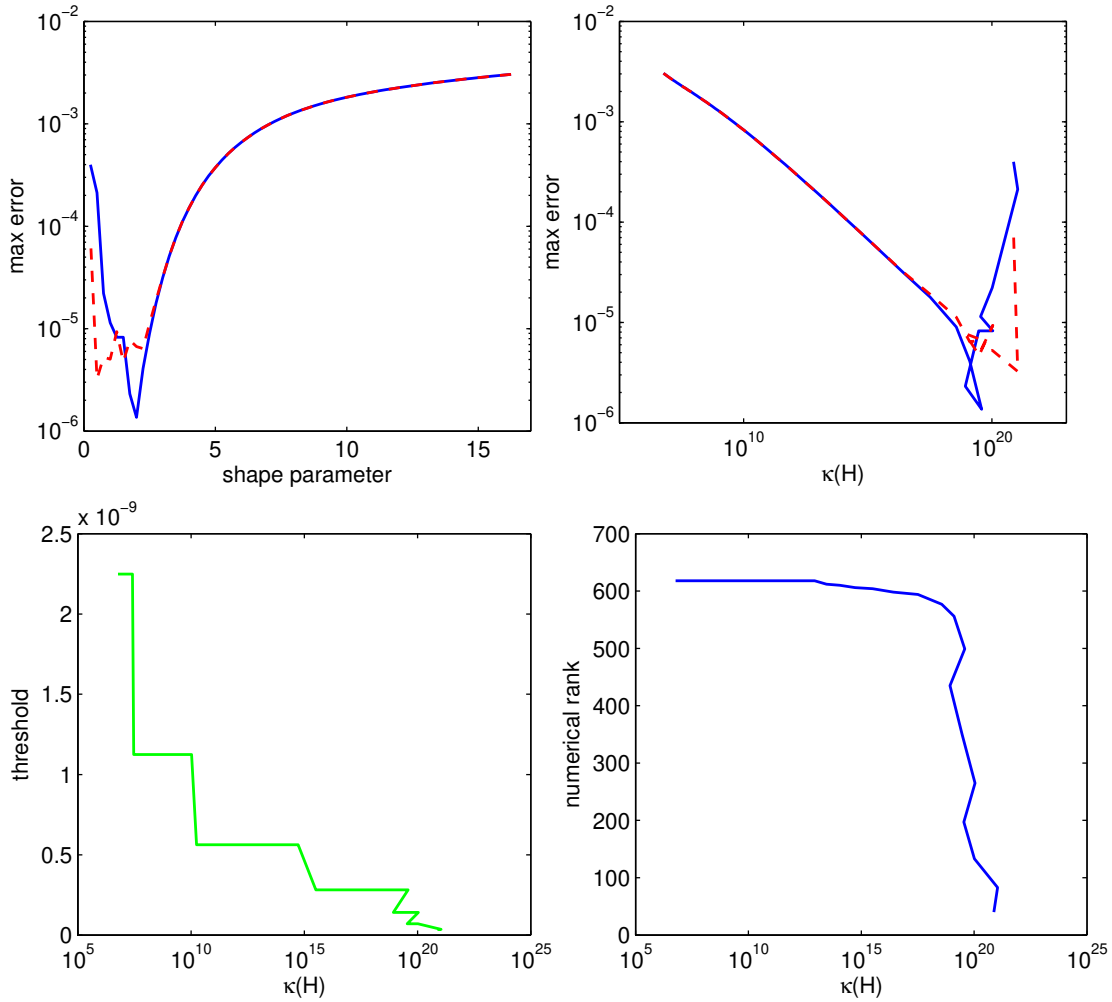

Figure 4.6: TSVD results from problem (4.1). Top: accuracy versus the shape parameter and condition number. Gaussian Elimination (solid blue) and TSVD (dashed red). Bottom left: the truncation threshold calculated by equation (4.4) versus the condition number. Bottom right: The numerical rank as determined by the threshold versus condition number. The full rank is 618.

k singular values are above the threshold, the inverse of B is computed via the truncated SVD as

$$B_t^{-1} = V\Sigma_t^{-1}U^T$$

where $\Sigma_t^{-1} = \text{diag}(1/\sigma_1, 1/\sigma_2, \ldots, 1/\sigma_k, 0, \ldots, 0)$. The truncated SVD scheme is widely used as an effective solver for ill-conditioned systems [96]. Matlab code that implements the truncated SVD algorithm is in listing 4.1. If the user does not supply a threshold value, the code uses the algorithm that Matlab uses to determine numerical rank to decide how many of the singular values are used in the solution. Matlab determines the numerical rank of a matrix $A \in \mathcal{C}^{M\times N}$ by calculating a threshold

$$\mu = \max(M, N) \times \text{eps}(\sigma_{\max}) \tag{4.4}$$

where eps(x) is the positive distance from $|x|$ to the next larger in magnitude floating point number of the same precision as x. The numerical rank is the number of singular values that are greater than the threshold.

```
1
2  function [x,invA,r,thres] = truncatedSvd(A,b,thres)
3
4     [U,S,V] = svd(A);
5     sigma = diag(S);
6
7     if nargin<3, thres = max(size(A))*eps(max(sigma)); end
8
9      r = sum( sigma > thres );        % estimate of numerical rank
10    Si = diag( (sigma > thres)./sigma );
11  invA = V*Si*U';
12     x = V*Si*(U'*b);
```

Listing 4.1: truncatedSvd.m

The solution of the Poisson problem (4.1) by the TSVD solver (***poissonTSVD.m***) is compared to the Gaussian Elimination solution in figure 4.6. Equation (4.4) is used to calculate the threshold that determines how many of the singular values are used to solve the linear system. The accuracy is compared over a range of shape parameters and against the corresponding condition number in the upper two images in the figure. With shape parameters less that 2, the TSVD is more accurate than Gaussian elimination or the standard SVD method as shown in figure 4.5. The lower left image of figure 4.6 shows that for this example, the truncation parameter is about 10^{-9} when the condition number is relatively small and that it decrease as the condition number increases. The bottom right image shows how many singular values are used versus the condition number of the system matrix.

All singular values are used until $\kappa(H) \approx 10^{13}$ and then a gradual reduction occurs until $\kappa(H) \approx 10^{20}$ at which point the TSVD uses less than 100 of the 618 singular values to compute the solution. The accuracy of the TSVD solution of this problem could not be improve by using either a larger or a smaller threshold than was computed by equation (4.4).

4.3.2 Improved Truncated SVD (ITSVD)

In [204], the authors describe an another truncated SVD solver that is claimed to perform better than the truncated SVD. The method does not disregard any of the singular values. It uses all the information contained in the SVD, but projects the very small singular values onto the nullspace to construct a more stable method. In references [139] and [56], the authors report good results in using the ITSVD with the MQ collocation method for PDEs. Matlab source code for the ITSVD solver is in listing 4.7.

```
1
2  function [x,aInv,r,thres] = improvedTruncatedSvd(A,b,thres)
3
4       [V,S,U] = svd(A);
5         sigma = diag(S);
6
7       if nargin<3, thres = max(size(A))*eps(max(sigma)); end
8  %     if nargin<3, thres = max(sigma)*(10e-8); end
9
10      r = sum( sigma > thres );        % estimate of numerical rank
11
12       sigmaI = diag( 1./diag(S) );               %  inverse of full S
13
14        I = diag(S) >= thres;
15        m = min(find(I==0));           %  m is index of first sigma thresholdeded
16       Si = diag( I./diag(S) );        %  inverse of thresholded S
17       x1 = U*Si*(V'*b);                 %  truncated SVD solution
18
19     if m>=1
20
21           U1 = U(:,1:m);      % left singlular vectors of TSVD
22           V1 = V(:,1:m);      % right singlular vectors of TSVD
23
24           U2 = null(U1');     % orthogonal complement of U1
25           V2 = null(V1');     % orthogonal complement of V1
26
27                b2 = V2'*b;
28                 C = V2'*A*U2;
29            kappaC = cond(C)
30
31           z2 = (U2')*(U*sigmaI*V')*(V2)*V2'*b;     %  or   z2 = C\b2;
32            x = x1 + U2*z2;
33         aInv = U*Si*V'  +  U2*(U2')*(U*sigmaI*V')*(V2)*(V2');
34
35     else
36
37         x = x1;
38      aInv = U*Si*V';
39
40     end
```

Listing 4.2: improvedTruncatedSvd.m

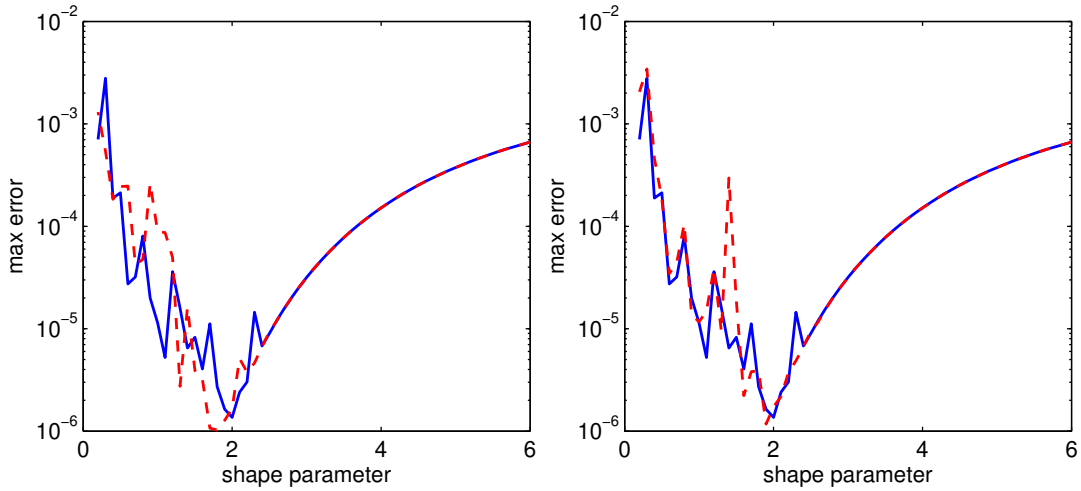

Figure 4.7: ITSVD results from problem (4.1), accuracy versus shape, ITSVD solution (dashed red) and Gaussian elimination (solid blue). Left: threshold = $\max(\sigma) \times 10^{-8}$. Right: same threshold (4.4) as the TSVD.

The results of applying the ITSVD solver (***poissonITSVD.m***) to the Poisson problem (4.1) are in figure 4.7. In the right image the same threshold (4.4) as the TSVD is used and the results are similar to when the standard SVD procedure is used. In the results in the right image a threshold of $\max(\sigma) \times 10^{-16}$ is used. In this example, the "improved" truncated SVD solver does not show any improvement over the TSVD method. However, in [56], the authors demonstrate better results by taking the threshold parameter to be

$$\text{threshold} = \max(\sigma) \times 10^{-8}. \tag{4.5}$$

The authors also recursively apply the ITSVD solver to the matrix C if the condition number is beyond a certain threshold.

If the errors from using the SVD to solve a poorly conditioned linear system are indeed due to the smallest singular values being inaccurately computed, then the motivation for using the small singular values in the ITSVD scheme seems questionable. Analysis of the stability properties as well as more thorough numerical experiments should be performed before the ITSVD is accepted as an improvement to the TSVD method.

4.4 Affine Space Approach

The Affine solver is not for use with general RBF linear systems, but is a specialized solver for linear boundary value problems. The RBF matrix that discretizes PDEs consists of two parts: the part that discretizes the PDE on the interior of the domain and the part that collates the boundary conditions. It is hypothesized that these two contributions are scaled dramatically different which in turn worsens the conditioning problem as compared to just interpolation. The affine space approach separates the boundary and interior contributions in order to lessen the conditioning problem [143].

Consider the asymmetric RBF collocation discretization of steady boundary value problem

$$\begin{bmatrix} \Phi_L \\ \Phi_B \end{bmatrix} \lambda = \begin{bmatrix} f \\ g \end{bmatrix}$$

or

$$\Phi\lambda = b.$$

From above we know that

$$\Phi_B\lambda = g.$$

Let the pseudoinverse of Φ_B be denoted by $\Phi_B^\dagger \in \mathcal{R}^{N\times N_B}$ and the null matrix of Φ_B by $\Psi_B \in \mathcal{R}^{N\times N_I}$. Then λ must lie in an affine space of a vector space formed by the columns of Ψ_B over a field $\left\{\Phi_B^\dagger g\right\}$. So

$$\lambda = \Phi_B^\dagger g + \Psi_B\gamma \quad \text{such that} \quad \Phi_B\Psi_B = 0$$

and

$$\Phi_B\lambda = \Phi_B\Phi_B^\dagger g + \Phi_B\Psi_B\gamma = g.$$

Substituting into

$$\Phi_L\lambda = f$$

we get

$$\Phi_L\left(\Phi_B^\dagger g + \Psi_B\gamma\right) = f$$

or

$$\left(\Phi_L\Psi_B\right)\gamma = f - \phi_L\Phi_B^\dagger g. \tag{4.6}$$

The reduced matrix $\Phi_L\Psi_B \in \mathcal{R}^{N_I\times N_I}$ has orthonormal columns. Thus, the length scale is determined completely by the PDE operator on the interior.

In [143], the authors develop several versions of the affine solver. We concentrate on the one that features the SVD (ASVD). Other versions feature the QR algorithm and Gaussian Elimination. In the ASVD approach, both $\Phi_B^\dagger$ are Φ_B formed via the SVD. $\Phi_B^\dagger$ is the inverse of Φ_B ignoring the singular values less than a threshold value, *tol*, and the null matrix consists of the right singular vectors corresponding to the singular values less than *tol*. The reduced system (4.6) is solved by the SVD as well. The ASVD solver gives extra flexibility to the asymmetric collocation method since it gives the method the ability to handle identical or extremely close centers. Additionally, the ASVD method allows safe computation over a large range of shape parameter. Matlab code that implements the affine solver is in listing 4.3.

```
1
2  function lambda = affinceSpaceSvdSolver(H,Nb,Np,f,thres)
3
4       phiB = H(1:Nb,:);
5       phiL = H(Nb+1:Np,:);
6
7      [U,S,V] = svd(phiB);
8        sigma = diag(S);
9
10     if nargin<5, thres = max(size(phiB))*eps(max(sigma)); end
11
12      r = sum( sigma > thres );          %  estimate of numerical rank
13     Si = diag( (sigma > thres)./sigma );
14
15     phiBdag = V(:,1:Nb)*Si*U';         %   phiBdag = pinv(phiB);
16
17       [U,S,V] = svd(phiB);
18       psiB = V(:,Nb+1:end);            %      psiB = null(phiB);
19
20     F = f(Nb+1:Np);
21     G = f(1:Nb);
22
23     [U,S,V] = svd(phiL*psiB);
24       sigma = diag(S);
25
26     if nargin<5, thres = max(size(phiL*psiB))*eps(max(sigma)); end
27
28      r = sum( sigma > thres );
29     Si = diag( (sigma > thres)./sigma );
30
31    % gamma = (phiL*psiB)\(F-phiL*phiBdag*G);    % or using GE
32      gamma = V*Si*U'*(F - phiL*phiBdag*G);
33     lambda = phiBdag*G + psiB*gamma;
```

Listing 4.3: affinceSpaceSvdSolver.m

Again, we consider the Poisson problem (4.1). The truncation parameter in the TSVD solver is set to $\mu = 10e-11$ as for this problem it produced better results that using formula (4.4) for μ. The Affine solver results are illustrated in figure 4.8. In the results, the Affine solver produced more accurate solutions that did Gaussian Elimination over a range of small shape parameters.

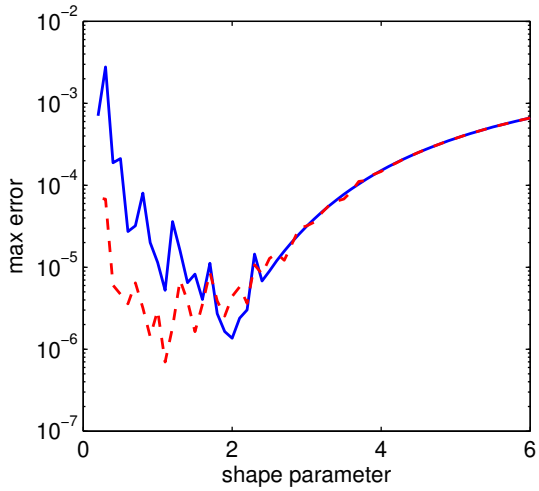

Figure 4.8: Affine solver results from problem (4.1) with *tol* = 10e-11, accuracy versus shape. Gaussian Elimination (solid blue) and Affine solver (dashed red).

4.5 GMRES iterative method

For RBF approximations with large N, using direct methods to solve the linear systems may not be possible due to their $\mathcal{O}(N^3)$ flop count and $\mathcal{O}(N^2)$ storage requirements. However, iterative methods have been successfully used to evaluate RBF approximations of problems with $N \gg 10,000$. The goal of iterative methods is to reduce the residual $r_k = \| B\alpha_k - f \|$ to below a specified tolerance by taking a small number of iterations $k \ll N$ with each iteration having a flop count of $\mathcal{O}(N^2)$ or less.

The GMRES (Generalized Minimal Residual) method [203] is a universally applicable (it does not require B to have special properties such as symmetry) Kyrlov subspace iterative method for solving linear systems $B\alpha = f$. In a basic implementation taking k iterations, the total storage requirements are $\mathcal{O}(kN)$ and the flop count is $\mathcal{O}(k^2 N)$. With some additional techniques, this can be reduced to $\mathcal{O}(N)$ storage and $\mathcal{O}(N \log N)$ flops respectively [12]. The convergence of the GMRES method depends on the locations of the eigenvalue of B in the complex plane. If B has a single cluster of eigenvalues and some outliers, the bound [35]

$$\| r_{\tau+k} \|_2 \leq \| r_0 \|_2 C\rho^k$$

holds where τ is the number of outliers, C quantifies the distance of the outliers to the cluster, and ρ is the radius of the clustered eigenvalues. The next section discusses a preconditioner for accelerating the convergence rate of the GMRES method.

4.5.1 Approximate Cardinal Basis Function Preconditioning

```
function neighbors = generateACBFStencils(x,y,ns,specialI)

                   n = length(x);
  numberOfSpecialPts = length(specialI);
           neighbors = zeros(n,ns+numberOfSpecialPts);

         xt = x;  yt = y;
         x(specialI) = 1e15;
         y(specialI) = 1e15;

for i=1:n
      neighbors(i,1:numberOfSpecialPts) = specialI;
                                     x0 = x(i);
                                     y0 = y(i);
                                      r = sqrt((x(:)-x0).^2 + (y(:)-y0).^2);
                                 [r,ix] = sort(r);
  neighbors(i,numberOfSpecialPts+1:end) = ix(1:ns);
end

% -- redo the special points --------------------------

x = xt; y = yt;
for j=1:numberOfSpecialPts
      i = specialI(j);
      neighbors(i,1:numberOfSpecialPts) = specialI;
                                     x0 = x(i);
                                     y0 = y(i);
                                      r = sqrt((x(:)-x0).^2 + (y(:)-y0).^2);
                                 [r,ix] = sort(r);
  neighbors(i,numberOfSpecialPts+1:end) = ix(2:ns+1);
end
```

Listing 4.4: generateACBFStencils.m

Approximate Cardinal Basis Function (ACBF) preconditioning is designed to tightly cluster eigenvalues of the preconditioned matrix around one so that the Generalized Minimal Residuals (GMRES) iterative method converges in a small number of iterations. The method computes a preconditioner W such that

$$WB\alpha = Wf$$

is easier to solve by GMRES than equation $B\alpha = f$. ACBF preconditioning was first used for the RBF interpolation problem in [12]. Later ACBF preconditioning was used with the RBF asymmetric collocation method for PDEs in [144], [145], and [27].

To set up the ACBF preconditioner, a small subset of centers from Ξ of size $N_s \ll N$ is selected for each center $\mathbf{x}_i^c$, and the indices of the support centers are put in the index set

$$\mathcal{S}_i = \left[s_i^{(1)}, \ldots, s_i^{(N_s)} \right]. \tag{4.7}$$

The support centers consist of some nearest neighbors to the center $\mathbf{x}_i^c$ as well as a group of special centers that cover the domain with a very course mesh. In figure 4.9, the special centers are the seven red colored centers on the boundary as well as the one red colored center in approximately the middle of the domain. The addition of the special points to the stencil has been found to improve the performance of the preconditioner on unbalanced stencils such as those near boundaries [12]. Then condition

$$\sum_{j \in \mathcal{S}_i} w_j \phi_j(\mathbf{x}) = \delta_i(\mathbf{x}), \quad \text{for all} \quad \mathbf{x} \in \Xi \tag{4.8}$$

is enforced. If equation (4.8) is enforced for all $\mathbf{x} \in \Xi$, the inverse matrix is formed. Instead, with $N_s \ll N$, equation (4.8) is satisfied in a least squares sense. A sub-matrix B_i of the system matrix B is associated with each center $\mathbf{x}_i^c$. The $N_s \times N$ sub-matrix contains the rows of B corresponding to the the index set $\mathcal{S}_i$. Thus,

$$B_i = \begin{bmatrix} B(s_i^{(1)}, 1) & B(s_i^{(1)}, 2) & \cdots & B(s_i^{(1)}, N) \\ B(s_i^{(2)}, 1) & B(s_i^{(2)}, 2) & \cdots & B(s_i^{(2)}, N) \\ \vdots & \vdots & & \vdots \\ B(s_i^{(N_s)}, 1) & B(s_i^{(N_s)}, 2) & \cdots & B(s_i^{(N_s)}, N) \end{bmatrix}$$

and equation (4.8) can be written in matrix form as

$$B_i^T w_i = e_i \tag{4.9}$$

where e_i is the i^{th} standard unit vector and $\mathbf{w}$ contains the nonzero elements of the i^{th} row of W.

Finally, the preconditioner W is formed by solving N least square problems. The least square problem may be solved by either using the QR factorization or the SVD [203]. In the examples we have used the reduced SVD to solve equation (4.9). For $N_s < N$, the reduced SVD is

$$B = \hat{U} \hat{\Sigma} V^T$$

where $\hat{U}$ has dimension $N \times N_s$, and $\hat{\Sigma}$ and V^T are $N_s \times N_s$. The solution of equation (4.9) is then computed by first solving the diagonal system

$$\hat{\Sigma} q = \hat{U}^T e_i$$

for q and then

$$w_i = V q.$$

Forming the the ACBF preconditioner naively using the SVD requires results in a setup cost of $\mathcal{O}\left(N_s^2 N^2 + N_s^3 N\right)$. It is shown in [145] that the setup cost can be reduced to $\mathcal{O}\left(N_s N^2 + N_s^3 N\right)$. A Matlab function that implements the preconditioner is in listing 4.5. In reference [144], the authors combine the techniques of preconditioning and domain decomposition to efficiently solve problems using more that 10,000 centers.

As an example, the ACBF preconditioner is applied to problem (4.1). We have taken $N_s = 28$ with the 28 supporting points consisting of the 20 nearest neighbors and the eight special points from figure 4.9. The Matlab code that finds the stencils is in listing 4.4. With $\varepsilon = 7$, the condition number of the matrix H is $\kappa(H) = \mathcal{O}\left(10e9\right)$ and the preconditioned matrix WH has the condition number $\kappa(WH) = \mathcal{O}\left(10e2\right)$. The eigenvalues of H and WH are shown in figure 4.10. GMRES converged in 35 iterations with a relative residual of 9.3e-9 and the error in the MQ Poisson solution was 9.7e-4. Larger shape parameters allowed the solution to converge in fewer iterations. Evaluating the example at other values of the shape parameters leads us to the same conclusion as in [145]. The preconditioner effectively lowers the condition number and clusters the eigenvalues so that GMRES converges in a small number of iterations, provided the condition number of the original problem is not too large. In this example, the number of iterations required for convergence grows with the condition number until the original problem has a condition number of approximately $\mathcal{O}\left(10e11\right)$ at which point the GMRES solution of the preconditioned problem does not converge.

4.6 Domain Decomposition

Domain decomposition methods are another tool for dealing with large N. The methods are based on the assumption that the given computational domain Ω is partitioned into subdomains Ω_i, $i = 1, \ldots, M$ which may or may

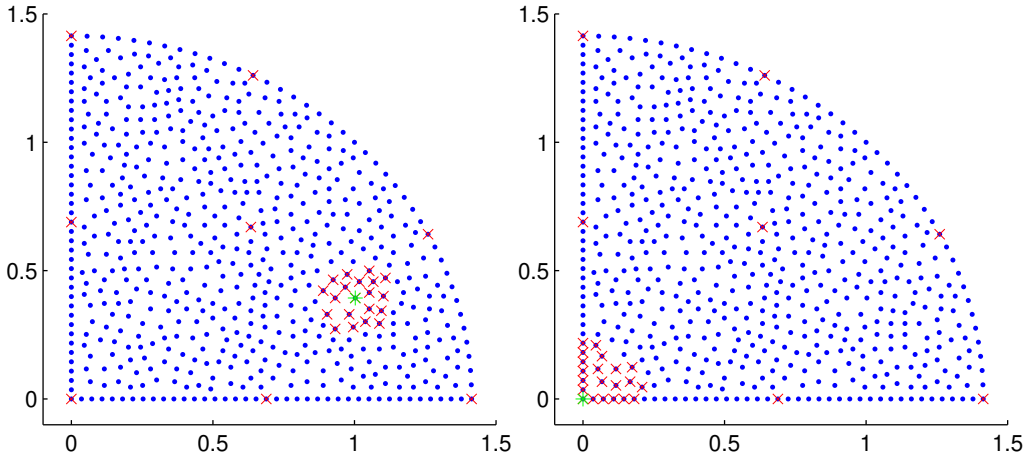

Figure 4.9: ACBF stencils with $N_s = 28$. The stencil is made up of 20 nearest neighbors and 8 special points. Stencil points in red (x's) and center point in green (*). Left: interior point. Right: boundary point.

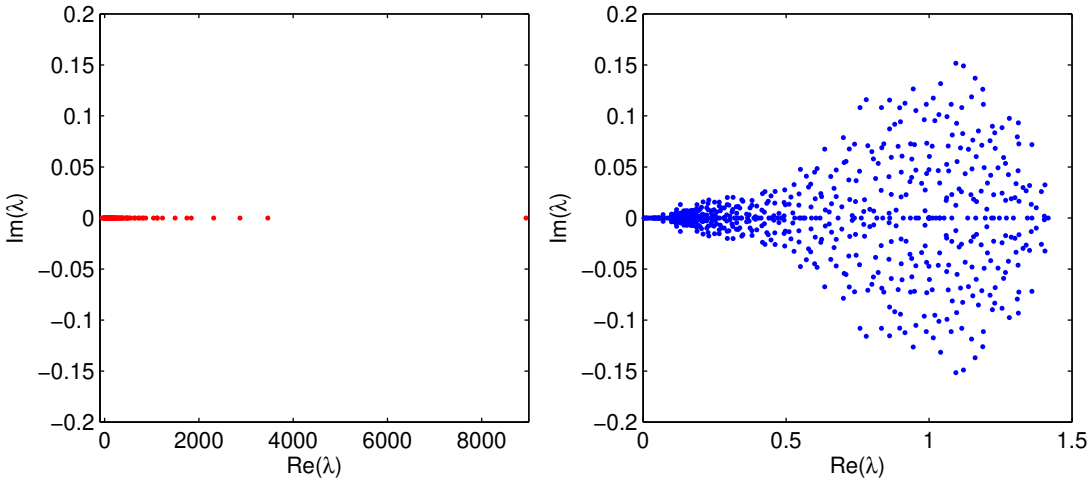

Figure 4.10: ACBF preconditioning example. Left: Eigenvalues of H. Right: Eigenvalues of the preconditioned WH.

```
%  Approximate Cardinal Function Preconditioner
%
%  Input
%       A   N x N matrix
%   neigh   neighorhoods of each centers
%
%  Output
%       W   preconditioned matrix

function W = acbfPreconditioner(A,neigh)

     N = length(A(1,:));
     W = zeros(N,N);

     for i=1:N

          [U,S,V] = svd(A(neigh(i,:),:)',0);
  W(i,neigh(i,:)) = ( V*( S\(U(i,:)')) )';

     end
```

Listing 4.5: acbfPreconditioner.m

not overlap. Then a subproblem of reduced size is solved on each subdomain Ω_i. Often, the M smaller subproblems may be solved with considerably less computational effort that it would take to solve the original problem on the entire domain. Domain decomposition methods also present the opportunity for the MQ method to be efficiently implemented on parallel computers. The family of subproblems are coupled to each other through the values of the unknown solution at the subdomain interfaces. Domain decomposition methods have been developed for finite difference, finite volume, finite element, and spectral methods for PDEs. An analysis of a large number of domain decomposition strategies can be found in the book [181].

In the RBF field, Hardy was the first to use a domain decomposition method in [98] in an application of the MQ method in image processing. Additional applications of domain decomposition with RBF methods can be found in references [137, 144, 114, 168, 46, 159].

The follow example illustrates an overlapping multiplicative Schwarz algorithm for an Elliptic PDE using $M = 3$ subdomains. The example is for illustrative purposes only. With a further division into smaller subdomains the problem could be solved more efficiently. Problem (4.1) is solved using the domain and centers in figure 4.11. The domain Ω is divided into three overlapping subregions (figure 4.11). Artificial boundaries Γ_i are defined on the parts of Ω_i that are in the interior of Ω. The remaining portion of the boundary (the natural boundary) is denoted by $\partial\Omega_i \backslash \Gamma_i$. The multiplicative Schwarz method for the problem is to first solve

$$L\phi_1^n = f \quad \text{in} \quad \Omega_1 \tag{4.10}$$

$$\begin{aligned} B\phi_1^n &= g \quad \text{on} \quad \partial\Omega_1 \backslash \Gamma_1 && (4.11)\\ \phi_1^n &= \phi_3^{n-1} \quad \text{on} \quad \Gamma_1 && (4.12) \end{aligned}$$

then

$$\begin{aligned} L\phi_2^n &= f \quad \text{in} \quad \Omega_2 && (4.13)\\ B\phi_2^n &= g \quad \text{on} \quad \partial\Omega_2 \backslash \Gamma_2 && (4.14)\\ \phi_2^n &= \phi_3^{n-1} \quad \text{on} \quad \Gamma_2 && (4.15) \end{aligned}$$

and finally

$$\begin{aligned} L\phi_3^n &= f \quad \text{in} \quad \Omega_3 && (4.16)\\ B\phi_3^n &= g \quad \text{on} \quad \partial\Omega_3 \backslash \Gamma_3 && (4.17)\\ \phi_3^n &= \phi_1^n \quad \text{on} \quad \Gamma_3^a && (4.18)\\ \phi_3^n &= \phi_2^n \quad \text{on} \quad \Gamma_3^b. && (4.19) \end{aligned}$$

The method is iterated until convergence criteria are satisfied. One possibility is to stop the iteration when the change in the solution at every center is less than a desired tolerance. For comparison, the problem is first solved on the entire domain with $\varepsilon = 2.9$ which results in a RBF evaluation matrix with condition number $\kappa(H) = 6.5e16$. The calculation takes 0.35s and the maximum error is 7.7e-5. In the domain decomposition method, we take the shape parameter as $\varepsilon = 2.4$ is each subregion which results in evaluation matrices with condition numbers $\kappa(H_1) = 1.5e18$, $\kappa(H_2) = 9.8e17$, and $\kappa(H_3) = 7.6e17$. The initial condition on the artificial boundaries are taken as the average of nearby points on the physical boundary. The method converges with 20 iterations in 0.15s with a maximum error of 6.7e-5. The domain decomposition method achieved a slightly smaller error with less than half of the computing time. The Matlab code that reproduces the domain decomposition example is in m-file ***poissonDomainDecomp.m***.

For parallelization, the additive Schwarz method is preferred. In [114], a 3d time-dependent diffusion equation was solved using 20,000 centers with up to 64 subdomains. Each subdomain was assigned to a processor.

4.7 Greedy Algorithm

Greedy algorithms ([146] and references within) work by first selecting a set of centers from a larger set in an adaptive way that keeps the condition number

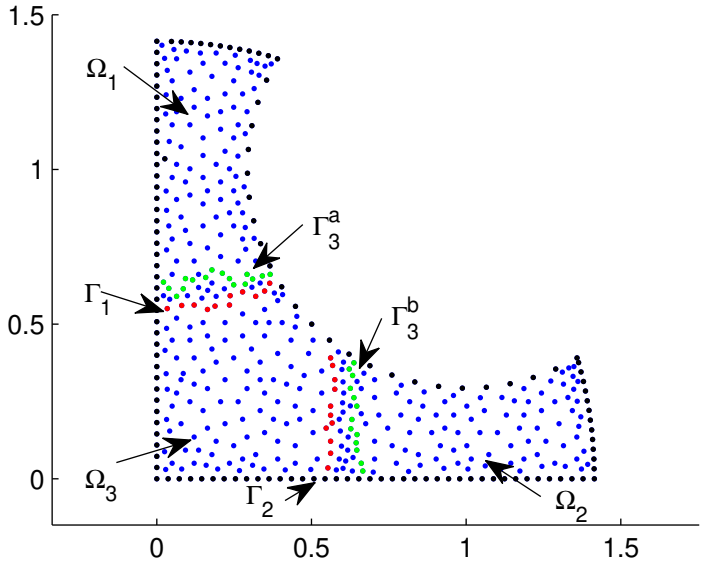

Figure 4.11: Subdomains and boundaries for domain decomposition of problem (4.1).

moderate and prevents numerical breakdown. In [146], a greedy algorithm is analyzed that uses a linear optimization scheme to select the n trial centers. After the subset of n centers is selected, the linear optimization algorithm performs a fit to all $N \gg n$ available test centers.

In reference [146], it was demonstrated that the over-determined linear system is best solved by a least squares solver rather than the linear optimization solver. Our numerical experiments confirmed this conclusion. This can be accomplished in Matlab using the backslash operator which solves the least squares problem using QR decomposition. Numerical results show that the proposed method is very stable. Matlab code for the greedy algorithm is in listing 4.6. The authors would like to thank Leevan Ling for providing the source code that listing 4.6 is based on.

As an example, the greedy algorithm is applied over a range of the shape parameter to the Poisson problem (3.10). The results are in figure 4.12. With a shape of $\varepsilon = 0.1$ (***poissonGreedySingleShape.m***), the smallest maximum error of 2.4e-9 is achieved with only 19 centers from the set of 60

```
% Takes matB (MC x nc*MC) matrix, with nc different
% values of c to find a column set, colset, of length MC. This method
% depend on the rhs column vector.  Solve  matB*ans = rhs

function [colset,ans] = greedyAlgorithm(matB,rhs)

 [MC,NC] = size(matB);
    mrhs = length(rhs);
       m = MC-1;
       n = NC-1;
     mat = zeros(MC,NC);
    rhsB = zeros(MC,1);
    rhsB = rhs;
     mat = matB;

 [mv imv] = max(abs(rhs ));           % get maximum residual
 [nv inv] = max(abs(mat(imv,:)));     % maximum in corresponding rowmat

    coef = zeros(NC,1);
    ainv = pinv(mat(imv,inv) );        % psuedo inverse

 coef(inv,1) = rhs(imv,1)*ainv;
 mcoef = length(coef);
  mimv = length(imv);
  ninv = length(inv);

    colset = [inv];     % this will be the selected column set
    rowset = [imv];     % this will be the selected row set
         v = zeros(MC,1);
  v(imv,1) = -coef(inv,1)*ainv;

 for k=2:MC;           %  loop for the new primal-dual algorithm
    r = mat*coef-rhs;
    q = (coef+mat'*v);
    [mv imv] = max(abs(r));
    [nv inv] = max(abs(q));
    rowset = [rowset imv];
    colset = [colset inv];

     localMatrix = mat(rowset,colset);          % set up local system
    [mM,nM] = size(localMatrix);
    kappaLocalMatrix = cond(localMatrix);

   if kappaLocalMatrix > 1e18, break;  end

      rhsloc = rhs(rowset);
     inv_mat = pinv(localMatrix);                % solve local system

                 Mrhs = length(rhsloc);
        [minmL,ninML] = size(localMatrix);
     [minvmtK,ninvmtK] = size(localMatrix);

                 xloc = inv_mat*rhsloc;
                 vloc = -(inv_mat')*xloc;
        coef(colset,1) = xloc;
           v(rowset,1) = vloc;
 end

 ans = coef(colset);
```

Listing 4.6: greedyAlgorithm.m

trial centers as shown in figure 4.13. The greedy algorithm solution is more accurate with 19 centers than any solution that was obtained over a range of shape parameters using all 60 centers in section 3.1. As the shape parameter ε gets smaller the greedy algorithms selects fewer and fewer centers. In figure 4.12, the performance of the greedy algorithm is shown over a range of shape parameters and is compared with the solution using all 60 centers. In the right image of the figure, the number of selected centers is shown for each value of the shape parameter.

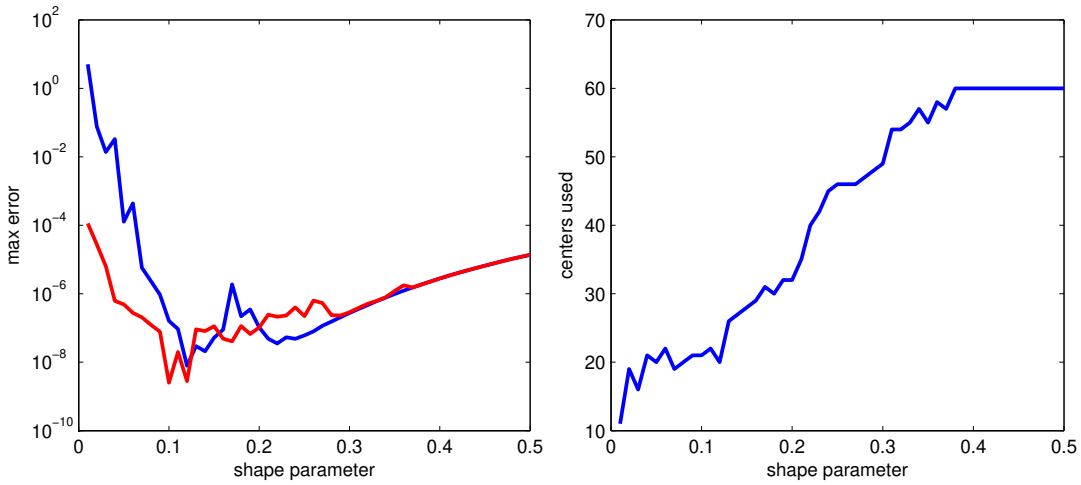

Figure 4.12: Greedy algorithm solution of equation (3.10). Left: Error versus the shape parameter. Standard method (blue) with all 60 centers, Greedy (red). Right: Number of centers selected by the Greedy algorithm versus shape parameter.

The Greedy algorithm can be used to select values of the shape parameter (***greedyInterpolation.m***) as well. To illustrate, we use the Greedy algorithm to select both the center locations and the shape parameters in interpolating the function

$$f(x) = e^{\sin(\pi x)} \qquad x \in [-1, 1]. \tag{4.20}$$

We start with $N = 40$ equally spaced centers on the interval $[-1, 1]$. As possible candidates for centers and shape parameter combinations, we consider 22 values of the shape parameter with each of the 40 centers. The 22 shape parameters are $\varepsilon = 0.5, 0.1, \ldots 1.05, 1.1$. The Greedy algorithm selected 23

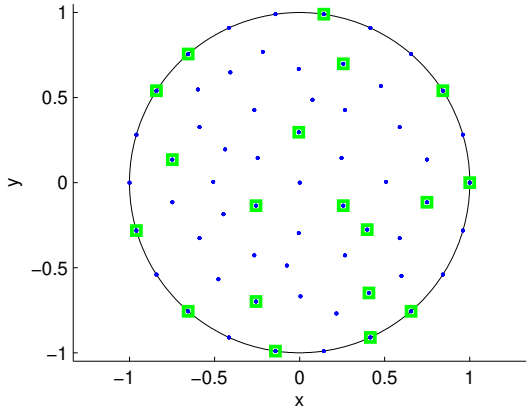

Figure 4.13: Greedy algorithm solution of equation (3.10). With $\varepsilon = 0.1$, the Greedy algorithms selects 19 (squares) of the original 60 centers. The maximum error is 2.4e-9.

center/shape combinations as shown in figure 4.14. The maximum error in interpolating the function using the selected centers/shapes is 1.5e-7. Note that the center at $x = -1$ is used twice, once with $\varepsilon = 0.65$ and once with $\varepsilon = 1.1$. The center at $x = 1$ is used four times, with $\varepsilon = 0.05, 0.4, 0.8, 1.1$.

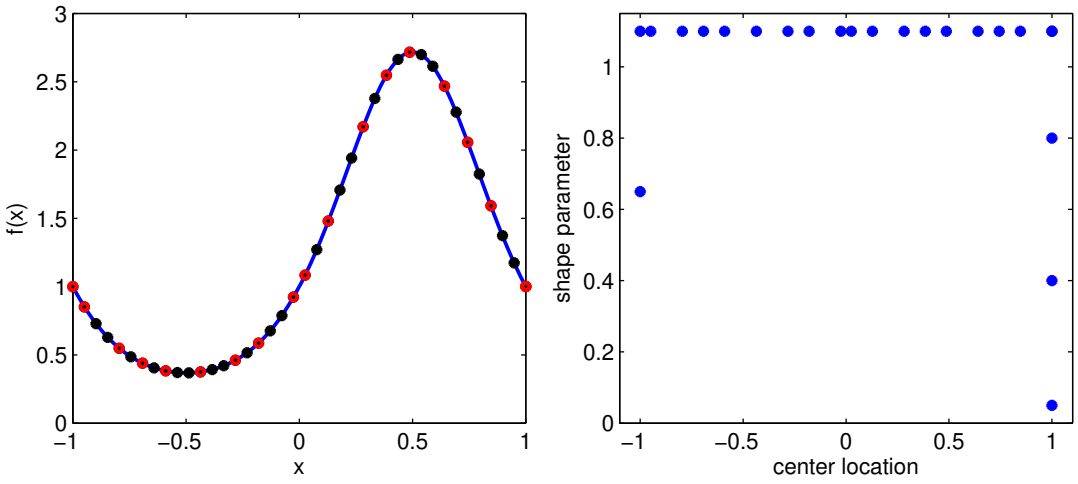

Figure 4.14: Greedy algorithm for interpolating function (4.20). Left: possible center locations (black), selected center locations (red). Right: shape parameter values at the selected center locations.

Recently in reference [147], an improved greedy algorithm has been developed which was shown in numerical experiments to outperform the greedy algorithm of [146].

4.8 Chapter Summary

The GMRES iterative method with ACBF preconditioning and Domain Decomposition can be efficiently used with large N to implement the MQ method with a flop count of less than $\mathcal{O}(N^3)$. Several methods were examined for accurately solving linear systems that result from using small centers and/or from having centers very close together in the MQ method. For small N, the Contour-Padé algorithm is a valuable tool for examining the accuracy of RBF methods with small values of the shape parameter ε. Using extended precision reveals the true accuracy of the MQ method but does so with a large increase in computational cost. Extended precision is justified if the same accuracy is realized with fewer data centers in a way that results in the total CPU time of the extended precision calculation being less than that of the standard precision calculation. RBF methods would benefit from improved computer architectures and software that could efficiently implement extended precision floating point arithmetic. In many places in the literature, SVD based algorithms have been cited as accurate linear system solvers for used with RBF methods. However, on our test problems, the SVD methods were outperformed by Gaussian Elimination with partial pivoting. The Affine Space solver for linear PDE boundary values problems displayed some promising accuracy at small values of the shape parameter ε. Finally, a greedy algorithm was found to be very effective in selecting both shape parameter values and center locations in order to accurate resolve problems using a small number of centers.

Chapter 5

Additional Tools, Techniques, and Topics

5.1 Formulas and Strategies for Shape Parameter selection

Many attempts have been made at finding, by numerical experiments or heuristic arguments, an "optimal" shape parameter. The simplest strategy, brute force, is to evaluate the MQ interpolant over a range of the shape parameter and call the value that results in the smallest error, optimal. In addition to being extremely inefficient, the strategy only works if the function being approximated is known.

The first formulas for "good" shape parameter values depended on the spacing between centers. In an early work, Hardy [97] recommended using

$$c = 0.815d \quad \text{where} \quad d = \frac{1}{N}\sum_{i=1}^{N} d_i \tag{5.1}$$

and d_i is the distance from the i^{th} center to its nearest neighbor. This is easily calculated in Matlab in terms of the distance matrix r as

```
c = 0.815*sum(min(r,[],2))/N
```

Later, Franke [84] offered a similar formula

$$c = (1.25D)/\sqrt{N} \tag{5.2}$$

where D is the diameter of the smallest circle enclosing all the centers.

Another strategy for selecting good values of the shape parameter is the statistical tool leave-one-out cross validation (LOOCV). In this algorithm, an optimal value of ε is selected by minimizing a cost function that collects the errors for a sequence of partial fits to the data. The predicted "optimal" shape parameter is usually close to the optimal value that would be found by the brute force method. In reference [182], the LOOCV algorithm was used for choosing an optimal value of the RBF shape parameter in the setting of scattered data interpolation. Subsequently, LOOCV was used in the context of RBF collocation methods for PDEs [67]. Listing 5.1 contains a Matlab script taken from reference [67] that calculates the cost function for a scattered data interpolation problem.

```
1
2  function costShape = costFunctionShape(shape,r,f)
3
4        B = mq(r,shape);
5     invB = pinv(B);
6        E = (invB*f)./diag(invB);        % cost matrix
7     costShape = norm(E(:),inf);
```

Listing 5.1: costFunctionShape.m

As an example, the shape strategies are used to calculate a shape parameter for interpolating the Franke function (2.8) using the 618 scattered centers in the left image of figure 2.2. The results are summarized in table 5.1. The shape parameters c have been converted to $\varepsilon = 1/c$. Listing 5.2 calls the script in listing 5.1 to find the minimum of the LOOCV cost function and to find the shape parameter.

```
1  % interpolationFrankeExampleLOOCV
2
3     xc = dlmread('frankeProblemCenters.txt',' ');
4      x = xc(:,1);     y = xc(:,2);
5      N = length(x);
6      f = frankesFunction(x,y);
7      o = ones(1,N);
8      r = sqrt(  (x*o - (x*o)').^2 + (y*o - (y*o)').^2 );
9
10     [shape,fval] = fminbnd( @(sh) costFunctionShape(sh,r,f), 1 ,5);
11     display('optimal shape by LOOCV: '), display(shape)
```

Listing 5.2: interpolationFrankeExampleLOOCV.m

The left image of figure 2.9 plots the interpolation error from this problem over a range of shape parameter for this example and the optimal brute force shape parameter of $\varepsilon = 2.8$ is identified. The LOOCV optimal shape is nearly

method	ε	$\kappa(B)$	$\max \lvert error \rvert$
equation 5.1	26.43	3.2e5	2.00e-3
equation 5.2	14.06	4.4e6	1.11e-3
LOOCV	2.9021	2.8e16	6.58e-6
brute force	2.8	8.1e16	4.27e-6

Table 5.1: Summary of shape parameter strategies for interpolation of function (2.8).

identical to the brute force optimal shape, while the strategies in equations (5.1) and (5.2) are not particularly effective in this example.

Various other attempts have been made to give formulas for a good values of the shape parameter, such as in references [75] and [201]. Many of the attempts to provide formulas for good values of the shape parameter reflect the uncertainty principle. The strategies have been developed by numerical experiments to produce a system matrix with a sufficiently large condition number to give good accuracy, but not too high so that errors began to dominate.

5.2 Variable Shape Parameter

Due to the extreme generality of RBF methods, it is very difficult to establish theoretical results. When theoretical results have been proven, in most cases a constant value of the shape parameter has been assumed. When a non-constant shape parameter is used, the theoretical analysis of the RBF methods becomes even more complex. However, in reference [24], sufficient conditions are provided for the non-singularity of the system matrix, B, arising from the use of a variable shape parameter.

Heuristically it has been argued that using a variable shape parameter is a good idea. A variable shape parameter refers to using a different value of the shape parameter at each center. This results in shape parameters that are the same in each column of the interpolation matrix, B. One argument for using a variable shape parameter is that it leads to more distinct entries in the RBF matrices which in turn leads to lower condition numbers. One negative consequence of using a variable shape is that B is no longer symmetric.

In [117], the formula

$$\varepsilon_j = \left[\varepsilon_{min}^2 \left(\frac{\varepsilon_{max}^2}{\varepsilon_{min}^2} \right)^{\frac{j-1}{N-1}} \right]^{\frac{1}{2}} \qquad j = 1, \ldots, N \tag{5.3}$$

which gives an exponentially varying shape parameter, was suggested in order to have a different value of the shape parameter with each basis function in the expansion (2.29). The shape parameter strategy (5.3) works equally as well as formula for ε_j using definition (2.1) of the MQ. This is due to the reciprocal relationship between the shape definitions and the fact that it does not matter whether the shape parameter increases or decreases with the center number [117]. Further numerical experiments with the variable shape parameter strategy (5.3) can be found in reference [123]. In [123] it was shown that very accurate approximation results could be obtained if ε_{min}^2 and ε_{max}^2 varied by several order of magnitude. The strategy was successful even when the underlying function varied rapidly or had steep gradients. However, also in reference [123], it was shown that this recipe did not always work. One such example was the surface of a sphere where a constant shape parameter worked the best.

Another variable shape parameter strategy is the linearly varying parameter

$$\varepsilon_j = \varepsilon_{min} + \left(\frac{\varepsilon_{max} - \varepsilon_{min}}{N-1} \right) j \qquad j = 0, 1, \ldots, N-1. \tag{5.4}$$

Recently, good numerical results have been reported [188] with the random variable shape strategy parameter

$$\varepsilon_j = \varepsilon_{min} + (\varepsilon_{max} - \varepsilon_{min}) \times \text{ rand(1,N)}. \tag{5.5}$$

The function `rand` is the Matlab function that returns N uniformly distributed pseudo-random numbers on the unit interval. Equation (5.5) returns N random shape parameters between ε_{min} and ε_{max}, and unlike the strategies (5.3) and (5.4), the parameters are not monotone increasing or decreasing.

With $\varepsilon_{min} = 2$ and $\varepsilon_{max} = 4\varepsilon_{min}$, the three variable shape strategies are used to calculate shape parameters for interpolating the Franke function (2.8) using the 618 scattered centers in the left image of figure 2.2. The results are summarized in table 5.2. The results with other ranges of ε_{min} and ε_{max} were similar. The exponentially varying strategy (5.3) gave the best results

method	$\kappa(B)$	max $\lvert error \rvert$
(5.3)	9.9e19	8.8e-5
(5.4)	4.9e19	1.1e-3
(5.5)	3.6e15	5.4e-4

Table 5.2: Summary of some variable shape parameter strategies for interpolation of function (2.8).

method	$\kappa(H)$	max $\lvert error \rvert$
(5.3)	5.2e19	2.12e-10
(5.4)	7.5e19	3.73e-10
(5.5)	3.2e19	1.90e-10

Table 5.3: Summary of some variable shape parameter strategies for the elliptic PDE (4.1).

on this problem but was unable the match the accuracy of the constant shape strategies of brute force and LOOCV of the previous sections.

Next, with $\varepsilon_{min} = 0.4$ and $\varepsilon_{max} = 4\varepsilon_{min}$, the three variable shape strategies are used to calculate shape parameters for the Poisson problem (4.1). In contrast to the previous variable shape interpolation problem, all three variable shape strategies give about four more decimal places of accuracy than the best constant shape result which had a maximum error of 1.35e-6 with $\varepsilon = 2.0$. The variable shape Poisson results are summarized in table 5.3.

The Greedy algorithm of section 4.7 is also a very effective method to specify the shape values in a variable shape parameter strategy. The was demonstrated in section 4.7 where the greedy algorithm was used to select both the subset of centers to use and the value of the shape parameter at each center.

In [210], the generalized MQ RBF (2.28) was used with a variable shape parameter strategy. In addition to varying the shape parameter ε, the exponent β was also considered a shape parameter as it also determines the shape of the basis functions. In this type of shape parameter strategy, both ε and β may be non-constant.

A variable shape parameter strategy has been used to successfully mit-

igate the effects of the Gibbs phenomenon that is associated with approximating discontinuous functions (reference [116] and section 5.5). In neighborhoods around discontinuities, the shape parameter c in the MQ is set to zero and the MQ becomes a first order linear spline with a first order algebraic convergence rate.

In section 3.2.2 eigenvalue stability was examined for the MQ spatial discretization of the advection equation 3.25. With $N = 40$ equally spaced centers, the smallest constant shape parameter that could be used to get a differentiation matrix with eigenvalues with non-positive real parts was $\varepsilon = 4$. Using the variable shape parameter (5.3) with $\varepsilon_{max} = 2.5$ and $\varepsilon_{min} = 1.8$ results in a system matrix with $\kappa(B) = 2.2\text{e+}15$ and a differentiation matrix with eigenvalues (left image of figure 5.1) all with non-positive real parts. To compare the accuracy (***advectionVariableShape.m***) of the two approaches, an initial condition of $u_0(x) = e^{-40(x-0.4)^2}$ is advanced to time $t = 1$ with a small time step of $\Delta t = 0.001$ to minimize temporal errors. The constant shape approach has a maximum error of 2.4e-5 while the variable shape approach has a maximum error of 2.02e-6. In this example, the variable shape approach results in overall better accuracy, while working with a system matrix with a higher condition number, but with a differentiation matrix with eigenvalues that all have non-positive real parts which allows for stable time integration.

5.3 Connection to Global Polynomial Approximation Methods

We have seen that small, but nonzero values of the shape parameter ε usually yield the best accuracy in numerical examples. Recently, RBF approximations in the limit $\varepsilon \to 0$ have been shown to have interesting properties. In the limit $\varepsilon \to 0$, the condition number of the system (2.4) grows rapidly and without bound. This fact has been a barrier to numerical investigation. However in [52], it was theoretically shown in 1-D that if some easily satisfied conditions are met in the limit $\varepsilon \to 0$, that the MQ interpolant is equivalent to the minimal-degree Lagrange interpolating polynomial. In higher dimensions, the limit may not exist. But when the limit does exist, it is a low degree multivariate polynomial [52]. References [128], [82], [32], [33], [132], and [190], provide further analysis of the limit $\varepsilon \to 0$ in two dimensions.

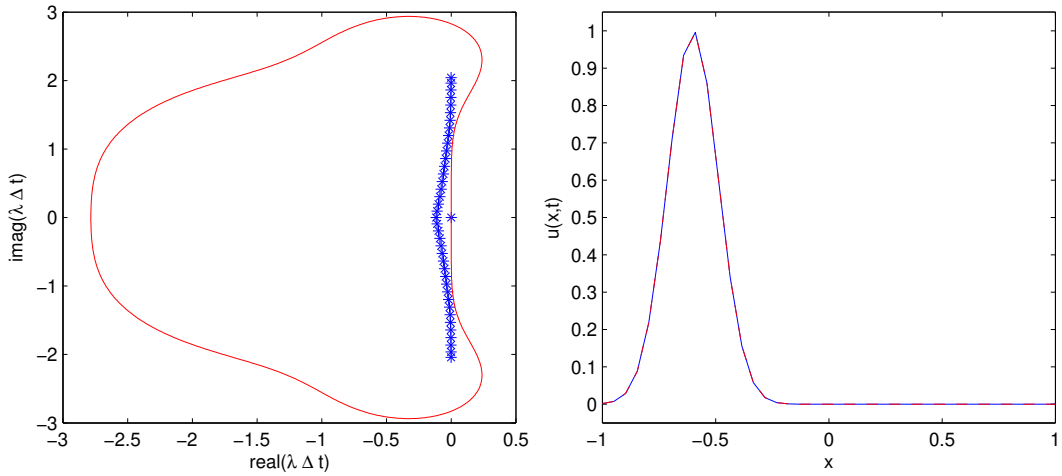

Figure 5.1: Left: stable spectrum for variable shape spatial discretization of problem (3.25). Right: variable shape numerical versus exact solution at $t = 1$ of problem (3.25).

The most interesting interpretation of this result is that in the limit $\varepsilon \to 0$, the MQ RBF method reproduces all the classical pseudospectral methods [100] such as Fourier, Chebyshev, and Legendre, whenever the center locations correspond to the pseudospectral grids. The MQ RBF method can therefore be thought of as a generalization of the pseudospectral approach that allows scattered points, in arbitrarily shaped domains, and that has a shape parameter that can be optimized.

5.4 Connection to Wavelets

By performing a wavelet (or prewavelet) decomposition a complicated function can be separated into simpler parts that can be analyzed separately. A wavelet decomposition of a square-integrable function, $f(x) \in L^2(\mathbb{R})$ is made at various levels of resolution by projecting it onto a nested sequence of spaces, $\{V_j\}$, $j \in \mathbb{Z}$, where

$$\{0\} \ldots \subset V_{j-1} \subset V_j \subset V_{j+1} \ldots \subset L^2(\mathbb{R}).$$

Each subspace $\{V_{j+1}\}$ can be decomposed into an approximation space $\{V_j\}$ and its orthogonal complement detail space $\{W_j\}$, i.e.

$$V_{j+1} = V_j \oplus W_j.$$

The name detail space comes from the fact that $\{W_j\}$ provides the details necessary to move from a coarser approximation space to a finer space. The space $L^2(\mathbb{R})$ can be expanded as an approximation space plus a sum of detail spaces as

$$L^2(\mathbb{R}) = V_{j=j_0} + \sum_{j=j_0}^{\infty} W_j$$

where j_0 is an arbitrary fixed level. The functions that generate the spaces W_j are called ***prewavelets***. The prewavelets are called ***wavelets*** if they are orthonormal bases of the W_j's.

The bases of approximation and detail subspaces are constructed with scaled and translated versions of functions $\phi_{j,k}$ and $\psi_{j,k}$ as respectively

$$\{V_j\} = \text{span}\{\phi_{j,k} = 2^{j/2}\phi(2^j x - k)\} \quad (5.6)$$

$$\{W_j\} = \text{span}\{\psi_{j,k} = 2^{j/2}\psi(2^j x - k)\} \quad (5.7)$$

where $j, k \in \mathbb{Z}$. The embedding relations $V_j \subset V_{j+1}$ and $W_j \subset V_{j+1}$ induce the following scaling and detail relations:

$$\phi(x) = \sum_k h_k \phi(2x - k) \quad (5.8)$$

$$\psi(x) = \sum_k g_k \phi(2x - k). \quad (5.9)$$

The sequences are respectively low-pass and high-pass filters.

The wavelet decomposition of $f(x)$ into $f_j \in V_j$ and $g_j \in W_j$ is

$$f(x) = \sum_k \gamma_{j,k}\phi_{j_0,k} + \sum_{j=j0} \sum_k \psi_{j,k}. \quad (5.10)$$

A key ingredient of the wavelet decomposition is the existence of the fast Discrete Wavelet Transform (DWT) [163] that provides a simple means of transforming data from one level of resolution j to the next coarser level of

resolution $j-1$ using

$$\gamma_{j-1,k} = \sum_{\ell} h_{2k-\ell}\gamma_{j,\ell} \tag{5.11}$$

$$\kappa_{j-1,k} = \sum_{\ell} g_{2k-\ell}\kappa_{j,\ell}. \tag{5.12}$$

The fast DWT avoids the tedious calculation of the coefficients, γ and κ, and allows the wavelet coefficients to be efficiently calculated using fast numerical convolutions followed by decimation. The scaling function coefficients $\gamma_{j,k}$ represent a smoothed version of the function at the current scale, while the wavelet coefficients $\kappa_{j,k}$ represent the irregular behavior (e.g. sharp gradients) of the function, at a certain region between the current scale and the next finest scale. The wavelet coefficients are therefore a measurement of the approximation accuracy at the corresponding level of resolution. Good approximations can be retained in smooth regions even though some detail functions associated with the low level wavelet coefficients at the finer levels are discarded. On the other hand, detailed local information can not be neglected if local irregularities exist.

The basic idea of an adaptive wavelet scheme is to represent a function with significantly fewer degrees of freedom, while still yielding a good approximation. At any level, the function under analysis is written as a sum

$$f(x) = f^1(x) + f^2(x)$$

where

$$f^1(x) = \sum_{k} \gamma_{j0,k}\phi_{j0,k} + \sum_{j=j0}\sum_{k} \kappa_{j,k}\psi_{j,k} \quad \text{for} \quad \kappa_{j,k} \geq \eta \tag{5.13}$$

$$f^2(x) = \sum_{j=j0}\sum_{k} \kappa_{j,k}\psi_{j,k} \quad \kappa_{j,k} < \eta. \tag{5.14}$$

For a sufficiently smooth function $f(x)$, the compression error is bounded by a prescribed threshold η as:

$$\big|\, f(x) - f^1(x)\big| \leq C\eta$$

where the parameter C depends on $f(x)$. The error indicator of the wavelet based adaptive algorithm is simply obtained by resolving the wavelet coefficients $\kappa_{j,k}$ in the corresponding detail spaces. Sharp transitions can be

captured by increasing the resolution in the regions with large values of the wavelet coefficients $\kappa_{j,k}$.

The work in references [29], [47], and [48] shows that MQ RBFs are prewavelets. The use of the MQ in the prewavelet methods allows the schemes to be extended to the case of non-equally spaced centers.

5.4.1 The Wavelet Optimized MQ method

The construction of a compressed adaptive distribution based on the standard wavelet scheme discussed in the previous section requires a solution at the finest level of resolution. This may cause a substantial overhead in problems with highly localized features, since these problems require a very fine distribution for properly capturing localized features. It would be more efficient to obtain a compressed solution by starting from a coarse distribution and then refining the solution where it is deemed necessary. Such an algorithm avoids the additional overhead of finding the solution on a non-adaptive fine distribution. Examples of standard wavelet approximation schemes that use local polynomial based finite difference methods can be found in reference [115] and the references within.

In [140], the authors have developed a more efficient approach that starts on a coarse distribution by using the MQ as the prewavelet. We refer to the method as the Wavelet Optimized MQ (WOMQ) method. The method starts on a coarse distribution of centers. Once the coarse solution is obtained, the magnitude of the current level prewavelet coefficients are examined. A large coefficient, $\kappa_{j,k}$, indicates significant variation in the solution and suggests refining the approximation by adding prewavelets at finer scales in this region. The threshold parameter η plays an important role both in determining the density of the resulting compressed center distribution and in the accuracy of the solution. Smaller values of η will result in more accurate results because a denser distribution is produced in highly localized areas. Assuming that the unknown dependent variables have been properly normalized by the maximum value, a threshold parameter between $\eta = 10^{-2}$ and 10^{-5} typically provides good results in applications. The main feature of the approach is its fast data distribution generation and its ability to create a non-uniform data distribution starting from an initial uniform coarse distribution. The initial solution can be obtained in the coarse approximation space, and refined by adding details over several levels until the equation is resolved to the desired accuracy. The error indicator is efficiently obtained by the fast

discrete wavelet transform algorithm.

In the higher n-dimensional case, the approach is analogous. The only difference is that the n-dimensional wavelet transform consists of the sequential application of n 1D wavelet transform in each of the n directions. A 2D implementation of the adaptive wavelet scheme is provided by the 2D DWT that leads to a decomposition of the coefficients at the $j+1$ level, $\gamma_{j+1,k,\ell}$, into four components: a scaling coefficients at level j, $\gamma_{j,k,\ell}$, and three wavelet coefficients, $\kappa_{j,k,\ell}$ in the horizontal, vertical and diagonal directions. One needs to consider all horizontal, vertical and diagonal details. Each wavelet coefficient indicates an important fluctuation in the corresponding direction. If the wavelet coefficient is greater than the threshold parameter, then the corresponding nodes at finer scales in this region are inserted. This allows the generation of anisotropic distributions that are a very important feature for multidimensional general adaptation schemes, but are not included in several standard adaptation strategies. More details may be found in reference [140].

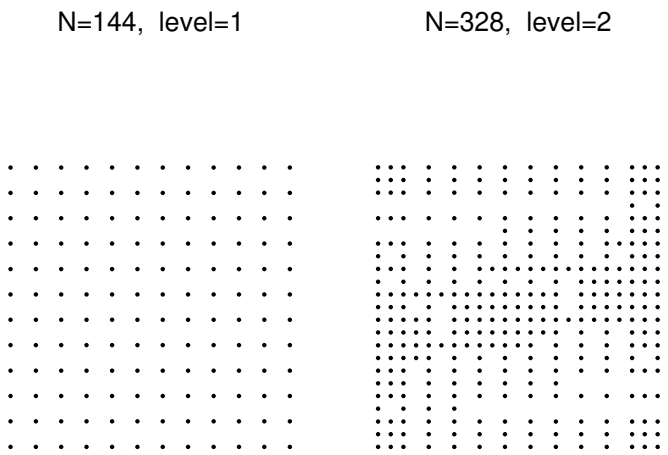

Figure 5.2: Left: Initial coarse WOMQ centers. Right: WOMQ centers after one level of detail has been added.

As an example we interpolate the function

$$f(x,y) = \arctan\left[12\left(y\left[\frac{2}{5}+\frac{1}{5}x\right]\right)\right] \tag{5.15}$$

using the WOMQ scheme. The method is started with with a 12×12 grid of uniformly spaced center location on the unit square. A threshold parameter

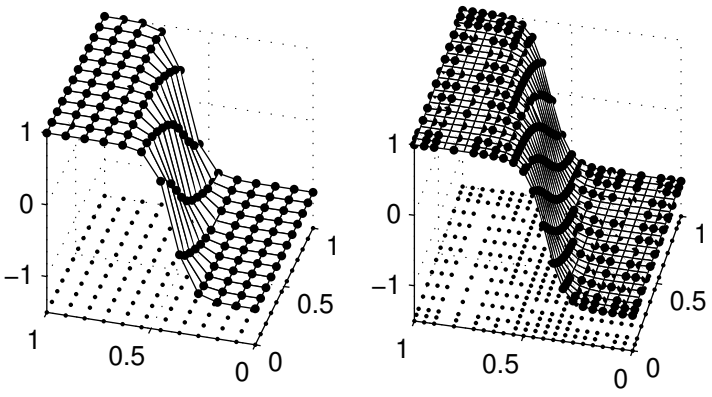

Figure 5.3: Function (5.15) on the initial coarse grid distribution (left) and on the WOMQ adaptive grid (right) after one level of detail has been added.

of $\eta = 5\text{e-}3$ is used to add a second level of detail. The new set of $N = 328$ centers is shown in the right image of figure 5.2. The centers and the function interpolated to a fine uniform grid at each level are shown in figure 5.3.

5.5 Approximating Discontinuous Functions

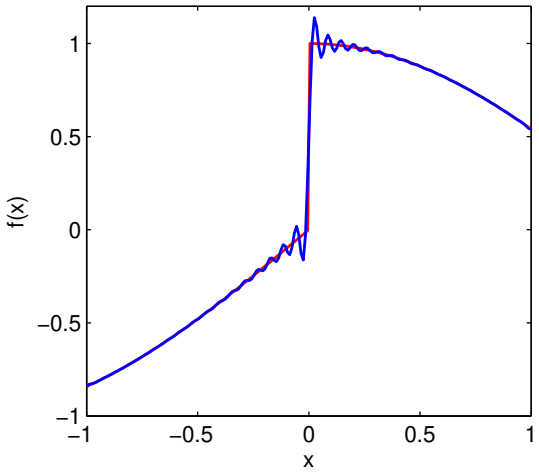

Figure 5.4: The Gibbs phenomenon in the MQ approximation of the discontinuous function (5.16).

For sufficiently smooth functions, MQ approximation methods are theoretically spectrally accurate (section 2.4). However, if the function being approximated has jump discontinuities, spectral accuracy will be lost and bounded oscillations will appear near the discontinuity as well as decreased accuracy throughout the domain. The oscillations and loss of accuracy are known as the Gibbs phenomenon. The Gibbs phenomenon was first observed in the context of truncated Fourier expansions but the term has evolved to characterize the nonphysical oscillations that appear in the approximation of discontinuous function by all interpolation methods. Figure 5.4 illustrates the Gibbs oscillations in the MQ approximation of the piecewise analytic function

$$f(x) = \begin{cases} \sin(x) & -1 \leq x < 0, \\ \cos(x) & 0 \leq x \leq 1. \end{cases} \tag{5.16}$$

In the example, a shape parameter of $\varepsilon = 4.5$ was used with $N = 70$ equally spaced centers. The interpolant was evaluated at $M = 200$ equally spaced evaluation points.

Recently, attempts have been made to characterize the Gibbs oscillations in RBF expansions and methods have been suggested to mitigate the effects of the Gibbs phenomenon. In [77], the authors examined the Gibbs phenomenon in one-dimension and concluded that the Gibbs phenomenon in RBF expansions has similarities to the Gibbs phenomenon in other expansions. It was also noted that the Gibbs oscillations in RBF approximations have some qualitative differences from the Gibbs oscillations in previously studied approximation methods, especially in terms of how the oscillations decay away from the jump. Previously, numerical experiments in reference [184] had revealed fundamental differences between the convergence rates of global polynomial and global RBF approximation methods for problems with jump discontinuities. While the presence of a discontinuity reduced the convergence rate of global polynomial methods from a spectral to a first order algebraic convergence rate throughout the domain, the MQ method exhibited a spectral convergence rate sufficiently far away from a discontinuity.

Following are some of the methods that have been developed to overcome or lessen the effects of the Gibbs phenomenon in RBF approximation methods.

5.5.1 Variable Shape Parameter

In [116], a variable shape parameter was used to lessen the effects of the Gibbs phenomenon in the MQ expansion. Using form (1.1) of the MQ, the shape parameter c is set to zero in regions around discontinuities. Thus, at centers around discontinuities, the basis functions become the first order accurate linear spline RBF $\phi(r) = r$. In numerical examples, the locally adaptive MQ method was shown to be very effective in reducing the effects of the Gibbs phenomenon. Figure 5.5 shows the results of applying the locally adaptive MQ method (***locallyAdaptiveMQ.m***) to the function (5.16). The parameters are the same as for the MQ method, except at five centers near the discontinuity at $x = 0$ the linear RBF is used. In the left image of figure 5.5, the locally adaptive approximation is compared to the exact function. In this example, the locally adaptive MQ method very sharply resolves the discontinuity and the accuracy throughout the domain is impressive when compared to the standard MQ method (right image of figure 5.5). The key to the success of the locally adaptive MQ method is the knowledge of the exact location of all discontinuities. In general, locating all discontinuities accurately is not easy. In [116], the MQ expansion coefficients α are used as an indicator of smoothness. It is proposed that if a function is smooth in a domain, then the maximum absolute value, $\max |\alpha_i|_{i=1}^{N}$, occurs in the neighborhood of boundaries and that otherwise the maximums exist in the neighborhood of the discontinuities. In this way, the expansion coefficients are used to determine in what regions the linear basis functions are used. For example, the MQ expansion coefficients α from the interpolation example in figure 5.4 are plotted in figure 5.6. The largest expansion coefficients are those corresponding to the basis functions that are centered near the discontinuity at $x = 0$. The idea is further explored in [54] to develop a method for using the MQ to locate jump discontinuities in a function.

5.5.2 Digital Total Variation Filtering

Instead of trying to prevent the Gibbs oscillations from occurring, a post-processing method can be used to remove the oscillations and to restore accuracy. The Digital Total Variation (DTV) post-processing method for RBFs was developed was developed in [184]. The DTV method is applicable with scattered center locations.

To describe the method, let Ξ be a set of centers and denote general

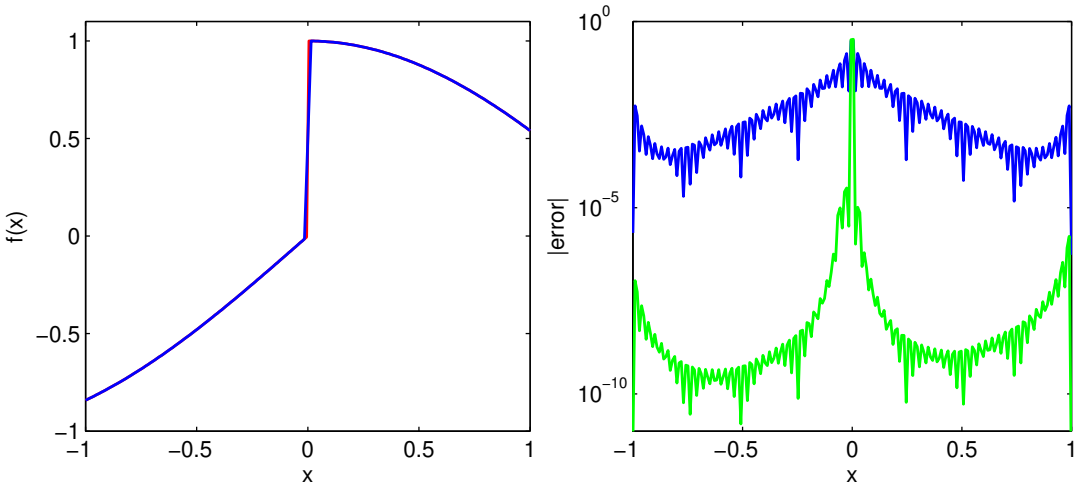

Figure 5.5: Left: Locally adaptive MQ approximation (blue) of function (5.16) versus exact (red). Right: Point-wise absolute error, MQ (blue) and locally adaptive MQ (green).

centers by $\alpha, \beta, \cdots$. The notation $\alpha \sim \beta$ indicates that α and β are neighbors. All the neighbors of α are denoted by

$$N_\alpha = \{\beta \in \Xi \,|\, \beta \sim \alpha\}. \tag{5.17}$$

The regularized location variation or strength function at any center α is defined as

$$|\nabla_\alpha u|_a = \left[\sum_{\beta \in N_\alpha} (u_\beta - u_\alpha)^2 + a^2\right]^{1/2}. \tag{5.18}$$

The notation u_α stands for the function value at center α. The variable u^0 represents the approximation containing the Gibbs oscillations and λ is the user specified fitting parameter. The regularization parameter a is a small value used to prevent a zero local variation and division by zero. The example that follows uses $a = 0.0001$.

The DTV method (***dtvFilter.m***) postprocesses the solution by solving a graph variational problem that minimizes a fitted TV energy. The unique solution to this problem is the solution of a nonlinear restoration equation. To evaluate the solution numerically, a preconditioned form of the restoration

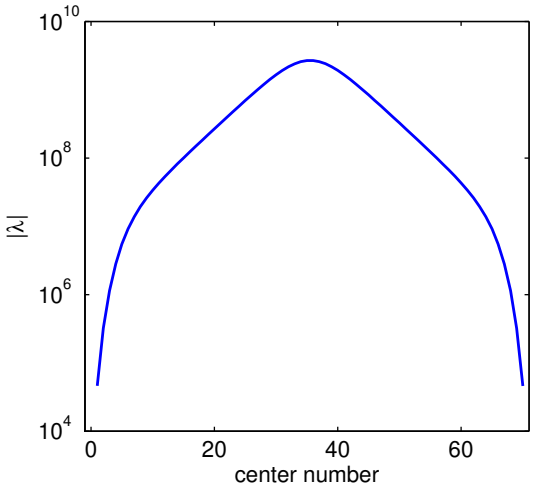

Figure 5.6: The magnitude of the expansion coefficients from the MQ example in figure 5.4. The largest expansion coefficients correspond to the basis functions centered near the discontinuity.

equation

$$\frac{du_\alpha}{dt} = \sum_{\beta\sim\alpha}(u_\alpha - u_\beta)\left(1 + \frac{|\nabla_\alpha u|_a}{|\nabla_\beta u|_a}\right) + \lambda\,|\nabla_\alpha u|_a\,(u_\alpha - u_\alpha^0) \tag{5.19}$$

is advanced to a steady state using time-marching with Euler's method. Typically about 100 time steps are required to reach a steady state. An optimal value of the fitting parameter is not known. However, a large range of values for the fitting parameter result in a "good" postprocessing. In general, stronger oscillations are best handled with a small fitting parameter (< 10) while weaker oscillations require a larger value of the fitting parameter. More details on selecting the value of the shape parameter can be found in reference [184].

In one-dimension, N_α simply consists of the nearest neighbors of each center. For two-dimensional scattered data, there are many ways to define N_α. One is to consider a p point neighborhood of a center α consisting of the p centers that are closest to α. For arbitrary scattered data, the flaw in this strategy is quickly exposed as it is possible for the points in the neighborhood to be configured as in the left image of figure 5.7, with all its members on one side of α. A more effective strategy is to divide the region surrounding

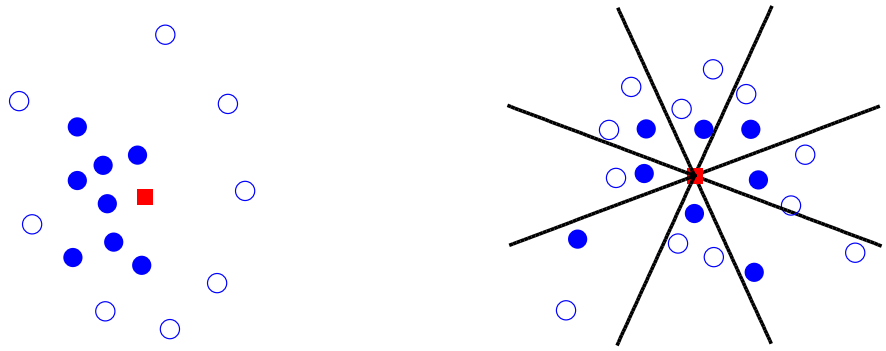

Figure 5.7: Left: A neighborhood N_α (filled circles) resulting from choosing the eight closest points to α. Such a neighborhood can result in a poor postprocessing due to points clustering on one side of α. Right: Eight point neighborhood N_α of the point α (the square in the center) chosen by selecting the closest point to α in each of eight regions. Filled circles are in N_α, open circles are not.

a center α in to p regions of equal angle and define N_α to consist of the point in each region that is closest to α. This strategy does not result in the p closest points to α, but it does ensure that N_α contains points in all directions around α. Taking $p = 8$ typically produces good results. An example neighborhood is shown in the right image of figure 5.7.

The results of applying the DTV filter to post-process the MQ approximation (***dtvFilterMQ.m***) from figure 5.4 are shown in figure 5.8. The DTV method sharply resolves the discontinuity but the accuracy throughout the domain is not nearly as impressive as that of the locally adaptive MQ method. However, unlike the locally adaptive MQ method, the DTV method does not need to know the locations of the discontinuities as it has built-in edge detection. This could be an advantage and make the DTV method more attractive in higher dimensions and with functions that have complex features, for which locating all the discontinuities could be difficult.

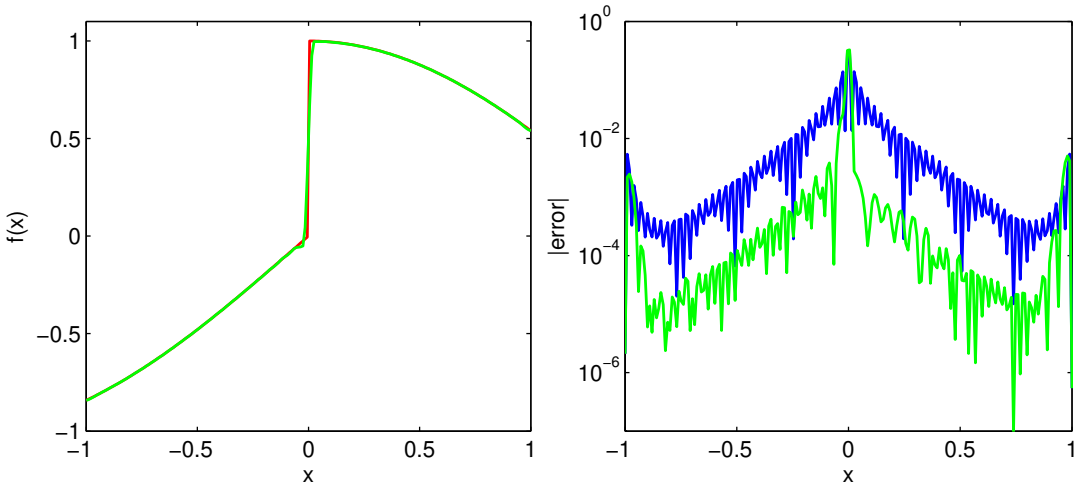

Figure 5.8: Left: DTV postprocessed solution of the oscillatory MQ approximation from figure 5.4. Right: Point-wise errors. MQ (upper blue), DTV postprocessed MQ (lower green).

5.5.3 Gegenbauer Post-Processing

The Gegenbauer Reconstruction Procedure (GRP) was developed to postprocess polynomial based spectral and pseudospectral approximations of discontinuous functions. Its development is chronicled in [90] and in the references within. Reprojection methods take the spectral approximation and project in onto another basis. In the new basis, spectral accuracy is recovered.

Recently, numerical examples were given in [26] to show that the GRP is effective in removing Gibbs oscillations and in restoring accuracy in RBF approximations of discontinuous functions. It is speculated, but yet proven, that the GRP is able to restore spectral accuracy in the RBF approximations of discontinuous functions. The GRP does not depend on the structured spectral grids. The interpolant just needs to be evaluated at Gaussian quadrature points. Thus, the GRP is easily applied to RBF approximations. Pinpointing the exact location of discontinuities is crucial to the success of the GRP. Methods for finding the locations of discontinuities in scattered data are given in reference [54].

The GRP uses the Gegenbauer or Ultraspherical polynomials, C_ℓ^λ, as the reprojection basis. The Gegenbauer polynomials are calculated via the three

term recurrence relationship

$$C_{k+1}^{\lambda}(\xi) = \frac{2(k+\lambda)\xi}{k+1} C_k^{\lambda}(\xi) - \frac{k+2\lambda-1}{k+1} C_{k-1}^{\lambda}(\xi), \qquad k = 2, 3, \ldots \tag{5.20}$$

with $C_0^{\lambda} = 1$ and $C_1^{\lambda} = 2\lambda\xi$. Let $\xi(x)$ be the map that takes $x \in [a, b]$ to $\xi \in [-1, 1]$ and let $x(\xi)$ be the inverse of the map. In each smooth subinterval i, the GRP postprocessed approximation is

$$f_P^i(x) = \sum_{\ell=0}^{m_i} g_\ell^i C_\ell^{\lambda}[\xi(x)] \tag{5.21}$$

where the Gegenbauer expansion coefficients are

$$g_\ell^i = \frac{1}{\gamma_\ell^{\lambda}} \int_1^{-1} (1-\xi^2)^{\lambda-1/2} C_\ell^{\lambda}(\xi) \mathcal{I}_N f[x(\xi)] d\xi \tag{5.22}$$

and

$$\gamma_\ell^{\lambda} = \pi^{\frac{1}{2}} \frac{\Gamma(\ell+2\lambda)\Gamma(\lambda+\frac{1}{2})}{\ell!\Gamma(2\lambda)\Gamma(\lambda)(n+\lambda)}.$$

The integral in (5.22) is evaluated via Chebyshev-Gauss-Lobatto quadrature. The Matlab code in listing 5.3 implements the GRP for 1d MQ approximations on the interval $[-1, 1]$. The code is a simple modification of the GRP for Fourier and Chebyshev pseudospectral methods from the Matlab postprocessing toolkit [186]. The gegenbauerPolynomial.m and im.m functions, which are called by the grpRbf.m, are part of the freely available Matlab postprocessing toolkit.

The result of post-processing the oscillatory MQ interpolant from figure 5.4 with the GRP is shown in figure 5.9. The GRP parameters in each smooth sub-interval were taken as $m = \lambda = 4$. Notice that accuracy is restored even at the point of discontinuity. The script that produces the example is in listing 5.4.

5.6 Finite Difference Mode

Finite difference mode RBF methods are the generalization to scattered node locations of the classical finite difference methods based on polynomial interpolation . Finite difference mode RBFs methods were originally independently investigated by Tjolstykh, et. al. [199, 200] and by Shu et. al. [195].

```
function ug = grpRbf(S,L,m,ak,xr,xc,shape)
  % gegenbauerPolynomial.m and im.m are part of the MPT
  %   http://www.scottsarra.org/mpt/mpt.html

     M = length(xr);          %  evaluate the GRP approx at M points
    sN = length(S);           %  number of discontinuities and endpoints
   siN = sN - 1;              %  number of sub-intervals

    Nq = 300;                 % number of quadrature points
    j = 0:Nq;                 %  Chebyshev-Gauss-Lobatto quadrature points
   xj = -cos(j*pi/Nq)';

      gh = zeros(siN,max(m)+1);  %  Gegenbauer coefficients
      ug = zeros(1,M);   %  reconstructed function at the reconstruction pts

 % ------- find GRP coefficients ---------------------------------

     wt = pi.*ones(1,Nq+1)./Nq;   % Chebyshev-Gauss-Lobbato quadrature weights
     wt(1) = 0.5*wt(1);
     wt(end) = 0.5*wt(end);

for i = 1:siN                                          % u_N( x[xi] )
   sL = S(i); sR = S(i+1); mi = m(i); Li = L(i);
    % RBF approx evaluated at mapped quadrature nodes
    H = evaluationMatrixMQ(xc,im(sL,sR,xj,0),shape);
   uC = H*ak;

 for el=0:mi
  hi = (gamma(Li)*(el+Li))/(sqrt(pi)*gegenbauerPolynomial(el,Li,1.0)*gamma(Li+0.5));
  fq = uC.*((1.0-xj.^2).^Li).*gegenbauerPolynomial(el,Li,xj);
  gh(i,el+1) = hi*sum(fq.*wt');
 end % el
end   % i

 % ------- reconstruct function ----------------------------------

for i=1:M
     for j=1:siN
      if xr(i)>=S(j) & xr(i)< S(j+1)
         sL = S(j); sR = S(j+1); mi = m(j); Li = L(j);
         for el = 0:mi
          ug(i) = ug(i) + gh(j,el+1)*gegenbauerPolynomial(el,Li,im(sL,sR,xr(i),1));
         end
         break  % get out of the j loop, the interval has been found
     end  % if in interval
    end  % for j
end  % for i

          i=M;
          ug(M)=0;
          sL = S(siN); sR = S(siN+1); mi = m(siN); Li = L(siN);
          for el = 0:mi
            ug(i) = ug(i) + gh(j,el+1)*gegenbauerPolynomial(el,Li,im(sL,sR,xr(i),1));
          end
          ug = ug(:);
```

Listing 5.3: grpRbf.m

```
      N = 70;  M = 200;
     xc = linspace(-1,1,N)';
      x = linspace(-1,1,M)';
      h = @(x) sin(x).*(x<=0) + cos(x).*(x>0);
      f = h(xc);
fExact = h(x);
  shape = 4.5;

  B = systemMatrix(xc,shape);
  H = evaluationMatrix(xc,x,shape);

   lambda = B\f;
  fApprox = H*lambda;

  S = [-1 0 1];   % endpoints and discontinuity locations
  L = [4 4];
  m = [4 4];

  fg = grpRbf(S,L,m,lambda,x,xc,shape);

   maxError = max(abs(fApprox - fExact))
  maxErrorPP = max(abs(fg - fExact))

  semilogy(x,abs(fApprox - fExact),'b',x,abs(fg - fExact),'r')
  xlabel 'x', ylabel '|error|'
```

Listing 5.4: grpRbfDriver.m

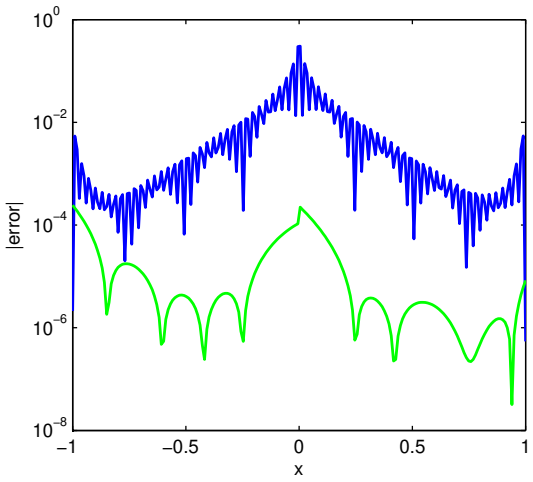

Figure 5.9: Point-wise errors from MQ approximation of function (5.16). MQ (upper blue), Gegenbauer postprocessed MQ (lower green).

Finite difference mode RBF methods are not exponentially accurate as the global MQ method is, but the RBF finite difference method does exhibit algebraic convergence rates that are similar to polynomial based finite difference methods on stencils of the same size.

The MQ method is a global method that uses information from all centers

```
1
2
3  function neighbors = generateFdStencils(x,y,ns)
4
5            n = length(x);
6    neighbors = zeros(n,ns);
7
8    for i=1:n
9       x0 = x(i); y0 = y(i);
10      r = sqrt((x(:)-x0).^2 + (y(:)-y0).^2);
11      [r,ix] = sort(r);
12      neighbors(i,1:ns) = ix(2:ns+1);              % don't include the i point
13   end % for i
```

Listing 5.5: generateFdStencils.m

in a computational domain to approximate a derivative at a single center. In finite difference mode, local information from a subset of N_s centers with $N_s < N$ is used in the approximation. The MQ finite difference (MQ FD) method associates with each center a vector of indices I containing the index of the center and the indices of its N_s supporting centers. The derivative of $f(\mathbf{x})$ at the j^{th} center is approximated as

$$\frac{\partial}{\partial x_i} f(\mathbf{x}_j^c) \approx w_I \cdot f(\mathbf{x}^c(I)). \tag{5.23}$$

The weights are the nonzero elements of row j of the differentiation matrix D given by equation (2.25), which is now a sparse matrix. The weights are found be restricting indices of i and j to I in the formulas for the system (2.5) matrix and i to the index of the center, and j to I in the formulas for evaluation matrix (2.24). Finding the weights at all N centers requires solving N linear systems of size $(N_s + 1) \times (N_s + 1)$.

Several methods have been suggested for selecting the subsets of supporting centers. If the centers are "nicely" distributed, such as by the near optimal algorithm (section 2.7), simply selecting the N_s closest centers works well. However, if the centers are more randomly located, a better selection criteria would be partition the region emanating from each center into slices of equal angle $N_s/(2\pi)$ and then selecting the closest center in each slice. This is the same strategy used for selecting the neighbors of a center in the DTV postprocessing method in section 5.5.2.

The following methods can be used to select the shape parameter for use in the MQ FD method. In references [194] and [195], the shape parameter is normalized by the radius of local support of each center. Another strategy for selecting the shape parameter at each center is motivated by the uncertainty principle (section 2.5). When calculating w_I, monitor the condition number

of the system matrix. Then, if necessary, increase or decrease the shape parameter and recalculate w_I until the condition number is within a range where good accuracy should be expected, i.e., until $\kappa(B)_{min} \leq \kappa(B) \leq \kappa(B)_{max}$. Taking $\kappa(B)_{min} = 1e{+}12$ and $\kappa(B)_{min} = 1e{+}15$ has proven effective in numerical experiments. These values have been used in the example that follows.

The diffusion problem that was solved by the MQ collocation method in a two-dimensional, complexly shaped domain in section 3.2.5 is now solved with the MQ FD method. Equation (3.24) is solved using the 635 centers in the left image of figure 3.8. The script in listing 5.5 finds the finite difference stencils based on the closest center criteria with $N_s = 11$. Two typical stencils are shown in figure 5.10. The weights for each stencil are calculated with the Matlab code in listing 5.6 and are put into a differentiation matrix that is returned in Matlab's sparse matrix format. The shape parameters are selected based on the condition number criteria discussed above and as implement in lines 29 through 35 of the listing. The sparse structure of the DM is illustrated in the left image of figure 5.11. The problem is advanced in time in ***heatComplexDomainFdMode.m*** to $t = 0.1$ at which time a maximum error of 5.4e-4 was recorded. The error using the global MQ method was 3.8e-6.

Additional development of the RBF finite difference method can be found in reference [194] where the authors combine the MQ finite difference method with approximate Riemann solvers and flux limiters to get an upwind MQ finite difference scheme for hyperbolic conservation laws with solutions featuring shocks. In reference [212], scattered node compact finite difference formulas based on RBF interpolation are developed.

5.7 Adaptive Center Locations

In problems that exhibit high degrees of localization, such as steep gradients, adaptive methods may be preferred over fixed grid methods. Since RBF methods are completely meshfree, implementing adaptivity in terms of refining and coarsening sets of centers is very straightforward compared to methods that require a grid.

In recent years, a number of adaptive RBF schemes have been suggested for use in both steady and time-dependent settings. In reference [25], B-spline techniques were used with scaled MQs as an adaptive method for interpo-

```
1
2  % INPUTS
3  %     st  indexes of the fd stencil points
4  %      x  grid
5  %      y
6  %      c  initial shape parameter
7  %   minK  min condition number of the rbf matrix, e.g. 1e+12
8  %   maxK  max condition number of the rbf matrix, e.g. 1e+15
9  %     dc  shape parameter increment, e.g. 0.1
10 % OUTPUTS
11 %     wt  weights
12 %     dm  differentiation matrix
13
14 function [dm,wt] = calculateFdStencilWeightsHeatComplex(st,x,y,c,minK,maxK,dc)
15
16       N = length(x);
17      wt = zeros(N,length(st(1,:)));        % weights
18      dm = zeros(N,N);
19
20      for i=1:N
21         tn = st(i,find(st(i,:)));
22         pn = [tn i];                 % include the base point of the stencil
23
24          o = ones(1,length(x(pn)));
25         rx = x(pn)*o - (x(pn)*o)';  ry = y(pn)*o - (y(pn)*o)';
26          r = sqrt(rx.^2 + ry.^2);
27
28         K = 1;
29         while (K<minK | K>maxK)      % find a system matrix with desired K
30                  B = mq(r,c);     K = cond(B);
31                  if K<minK
32                    c = c - dc;
33                  elseif K>maxK
34                    c = c + dc;
35                  end
36         end
37
38  H = mqDerivatives( sqrt( (x(i)- x(pn)).^2 + (y(i)-y(pn)).^2 ),x(i)- x(pn),c,2) + ...
39      mqDerivatives( sqrt( (x(i)- x(pn)).^2 + (y(i)-y(pn)).^2 ),y(i)- y(pn),c,2);
40
41         wt(i,1:length(pn)) = H'/B;
42         dm(i,pn) = wt(i,1:length(pn));
43
44      end % i
45
46   dm = sparse(dm);
```

Listing 5.6: calculateFdStencilWeightsHeatComplex

lation. Instead of using piecewise linear spline interpolants as in standard B-spline techniques, the scaled MQs provided a better interpolant. In problems with a boundary layer in one space dimension, the authors in [107] proposed an adaptive technique using the MQ RBF which reallocated centers to regions with sharp gradients. The work in [15, 16] combines an adaptive semi-lagrangian method with local thin plate spline interpolation. The adaptive method performed well on nonlinear time-dependent transport equations. In [183], an equidistribution of arclength algorithm in space is used to adaptively select center locations for time-dependent PDEs in one space dimension. The paper [53] describes the adaptive method of residual subsampling. Residual subsampling starts with non-overlapping boxes, each of which contain an active center. Once a solution has been computed for the active center set,

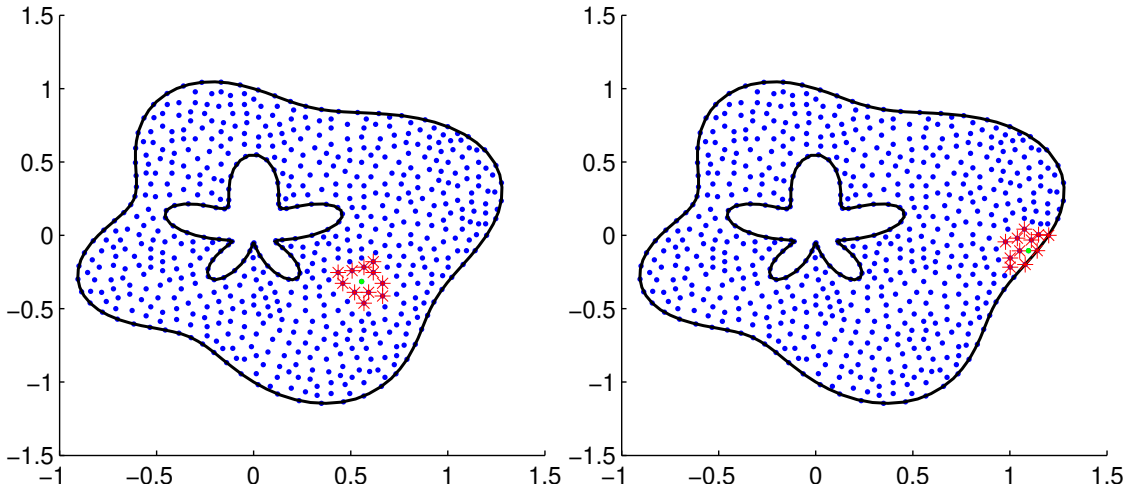

Figure 5.10: 12 point MQ finite difference stencils for problems (3.24). 11 red centers (stars) surround 1 green center at which point the derivative is being approximated. Left: an interior stencil. Right: a stencil that includes boundary centers.

the residual of the resulting approximation is sampled on a finer node set in each box. Nodes from the finer set are added to or removed from the set of centers based on the size of the residual of the PDE at those points. The solution is then recomputed using the new active center set for a new approximation. Residual subsampling has been applied to steady boundary value problems in one and two dimensions and to time-dependent problems in one dimension. For time-dependent problems a method of lines approach can be taken, or the problem may be considered as a space time problem and treated as a two dimensional problem with the distance function being taken as $r(x,t) = \sqrt{(x-x_j)^2 + (t-t_j)^2}$. Matlab code that implements residual subsampling can be found at [36]. Another adaptive method, the Wavelet optimized MQ (WOMQ) method [140], is discussed in section 5.4.

The Burgers' problem (3.21) with a small viscosity parameter develops a sharp gradient in the solution. Next, the problem is used to illustrate two of the adaptive center methods. Figure 5.12 shows center locations for the adaptive equidistribution of arclength and residual subsampling methods for Burger's equation (3.21) with viscosity parameter $\nu = 0.000875$ at time $t = 0.2$. Both methods were advanced in time using a method of lines approach. Equidistribution used a fourth order Runge-Kutta method and

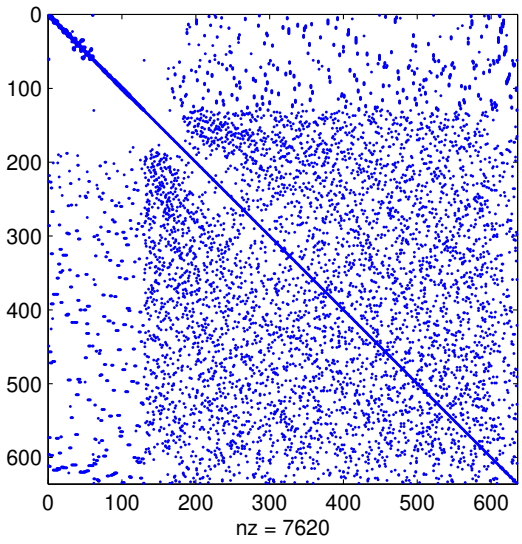

Figure 5.11: The sparse MQ finite difference mode DM for problem (3.24).

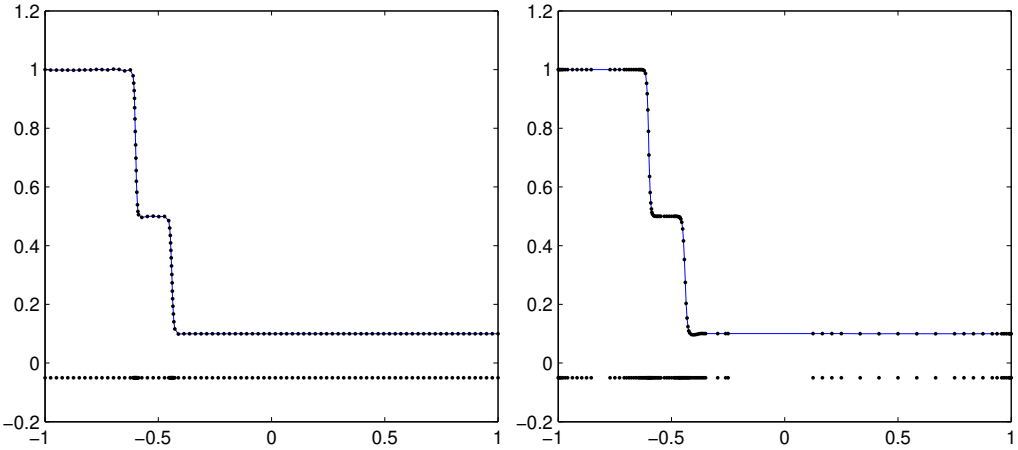

Figure 5.12: Adaptive MQ solution of problem (3.21) with $N = 108$. Left: Equidistribution of arclength. Right: Residual subsampling.

residual subsampling used the Matlab function ode15s. The equidistribution algorithm keeps the number of centers fixed and also prevents the ratio of maximum to minimum center spacing from becoming too large. Residual subsampling allows the number of centers to vary according to a tolerance on the residual and allows centers to become widely scattered if necessary.

Without any attempt to optimize the parameters of the methods, the maximum errors for this example were 0.0077 for equidistribution and 0.0209 for residual subsampling.

5.8 Integrated Multiquadric Methods

The motivation for Integrated Radial Basis Function (IRBF) methods is the fact that the order of convergence of an RBF derivative approximation is reduced by the order of the differentiation involved [155]. Integrated RBF methods integrate the original RBF with respect to r, one or more times, to get new basis functions in hope of restoring or even improving the convergence of the RBF methods. In [158], the authors apply the IRBF method with good results in approximating functions. Subsequently, the IRBF method was successfully used to solve differential equations [125, 148, 160, 161, 159]. Recently some general properties and theoretical results have appeared in [185].

The notation $\phi^n(r;\varepsilon)$ represent a RBF that has been integrated ($n > 0$) or differentiated ($n < 0$) n times with respect to r. IRBFs are easily found using a computer algebra system. For reference, the first four members of the MQ integrated RBF family are:

$$\phi^1 = \frac{\varepsilon r\sqrt{1+\varepsilon^2 r^2} + \sinh^{-1}(\varepsilon r)}{2\varepsilon}, \tag{5.24}$$

$$\phi^2 = \frac{(-2+\varepsilon^2 r^2)\sqrt{1+\varepsilon^2 r^2} + 3\varepsilon r \sinh^{-1}(\varepsilon r)}{6\varepsilon^2}, \tag{5.25}$$

$$\phi^3 = \frac{\varepsilon r\sqrt{1+\varepsilon^2 r^2}(-13+2\varepsilon^2 r^2) + 3(-1+4\varepsilon^2 r^2)\sinh^{-1}(\varepsilon r)}{48\varepsilon^3}, \tag{5.26}$$

$$\phi^4 = \frac{\sqrt{1+\varepsilon^2 r^2}(16-83\varepsilon^2 r^2+6\varepsilon^4 r^4) + 15\varepsilon r(-3+4\varepsilon^2 r^2)\sinh^{-1}(\varepsilon r)}{720\varepsilon^4}. \tag{5.27}$$

The integrated MQ basis functions are referred to as IMQ1, IMQ2, ... to indicate how many time they have been integrated.

For n even, the methods behave as those based on the parent MQ RBF. That is, they are generally most accurate and most poorly conditioned for small values of the shape parameter ε. For n odd, the methods are most accurate and most poorly conditioned for large ε. A summary of properties

	n even	n odd
$\kappa(B)$ as $\varepsilon \to 0$, N fixed	∞	$\kappa(B)$ for $\phi(r) = r$
$\kappa(B)$ as $\varepsilon \to \infty$, N fixed	$\kappa(B)$ for $\phi(r) = r^{n+1}$	∞
1d interpolant as $\varepsilon \to 0$	$\phi^0 \equiv$ Lagrange interpolant	$\phi(r) = r$
interpolant as $\varepsilon \to 0$, $d > 1$	depends on ϕ^n	$\phi(r) = r$
large ε interpolant	$\phi(r) = r^{n+1}$	$\phi(r) = \varepsilon^2 r^{n+1} + r^{n-1} \log r$

Table 5.4: Summary of IRBF properties.

is listed in table 5.4. As a result of the improved accuracy of the IRBFs over their non-integrated counterparts, the condition number of the system and evaluation matrices of the IRBFs will be larger as dictated by the uncertainty principle. This indicates they will be most effective if employed with small N (smaller than the non-integrated RBFs).

As an example, the IMQ7 method is applied (***poissonExampleIMQ7.m***) to solve the Poisson problem that was solved by the MQ method in chapter 3. The example uses the same set of $N = 60$ centers on the unit circle shown in figure 3.1. In the left image of figure 5.13, the error over a range of shape parameter is shown. For this example, the "optimal" value of the shape parameter is close to $\varepsilon = 10^5$ at which point the condition number (right image of figure 5.13) of the evaluation matrix is becoming critical. With $\varepsilon \approx 350,000$ the IMQ7 produces the smallest error of 6.5e-11 which is several decimal places more accurate than the MQ method in chapter 3. In this particular example of IMQ7 stationary approximation, with the number of centers fixed at $N = 60$, a second order algebraic convergence rate is observed as the shape parameter increases in the left image of figure 5.13 where the graph of the maximum error versus the shape parameter is a line of slope -2 on the loglog plot. This type of convergence is unlike polynomial based methods that do not have a shape parameter to refine.

5.9 Boundary Collocation

In [69], a modified version of the asymmetric collocation method is described for steady linear and nonlinear problems in one and two dimensions. The modified version enforces both the PDE and the boundary conditions at the boundary centers. The method is called the MQ PDECB method. In order

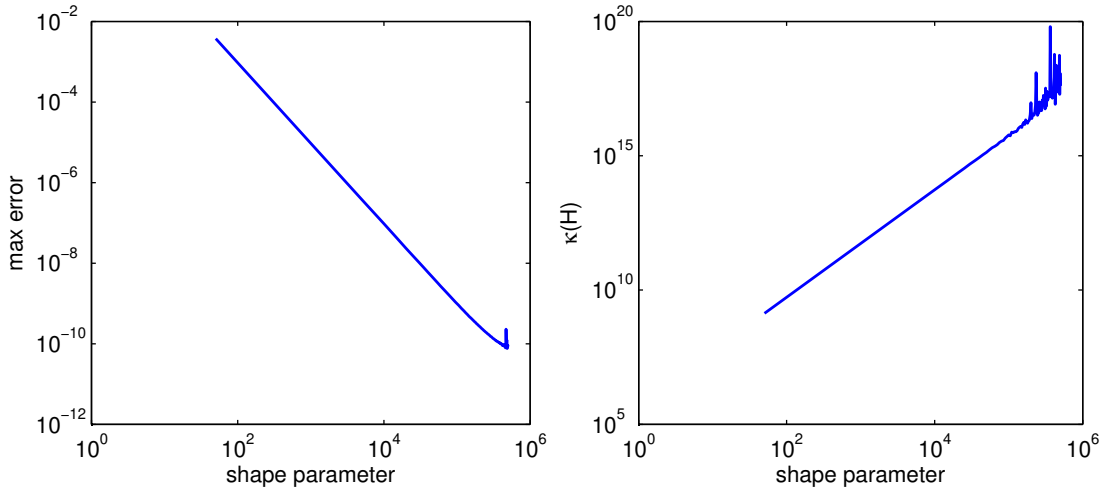

Figure 5.13: Left: Maximum error versus the shape parameter for problem (3.10) using the IMQ7 RBF. Right: Condition number of the evaluation matrix, H, from problem (3.10) versus the shape parameter using the IMQ7 RBF.

for the resulting linear system remains square, an additional set of centers are added adjacent to the boundary. The added centers can lie inside or outside of the domain, but numerical experiments indicate that it is best to place the additional centers outside the domain. The motivation for the MQ PDECB method comes from the observation that the residual is typically the largest near the boundary. The errors in boundary regions are often one to two orders of magnitude larger than in the interior. The goal of boundary collocation is to reduce the residual in boundary regions.

Let $\mathbf{w}^c$ represent the enlarged set of centers containing interior and boundary centers and the additional N_B centers outside the boundary. If the set $\mathbf{w}^c$ is ordered so that the N_I interior centers are first, followed by the N_B boundary centers, and the additional N_B centers outside the domain added last, then the MQ PDECB method can be described as follows. With the additional centers, the MQ interpolant (2.29) becomes

$$s(\mathbf{x}) = \sum_{j=1}^{N+N_B} \alpha_j \phi(\left\| \mathbf{x} - \mathbf{w}_j^c \right\|_2, \varepsilon) \tag{5.28}$$

and equation (3.3) and (3.4) from the standard collocation method are mod-

ified as

$$\mathcal{L}u(\mathbf{x}_i^c) = \sum_{j=1}^{N+N_B} \alpha_j \mathcal{L}\phi(\left\| \mathbf{x}_i^c - \mathbf{w}_j^c \right\|_2), \quad i = 1, \ldots, (N_I + N_B), \tag{5.29}$$

at the N_I interior centers, and as

$$\mathcal{B}u(\mathbf{x}_i^c) = \sum_{j=1}^{N+N_B} \alpha_j \mathcal{B}\phi(\left\| \mathbf{x}_i^c - \mathbf{w}_j^c \right\|_2), \quad i = N_I + 1, \ldots, (N_I + N_B). \tag{5.30}$$

at the N_B boundary centers. The structure of the linear system of the MQ PDECB method is the same as that of the standard collocation method but the MQ PDECB matrices are $(N + N_B) \times (N + N_B)$, rather than $N \times N$.

In [69], numerical results are given that show a considerable improvement in accuracy of the MQ PDECB method over the MQ collocation method. However, in reference [129], the asymmetric collocation method, the symmetric collocation method, and the MQ PDECB method are applied to a battery of 2d Elliptic test problem and of the three methods, the MQ PDECB method performed the poorest of the three methods.

To illustrate the MQ PDECB method, it is applied to the Poisson problem (3.10) using the 60 centers in figure 3.1 that include 22 centers located on the boundary. Figure 5.14 shows the locations of the 22 centers that are added outside of the domain for the MQ PDECB method. The locations of the centers outside the domain are not optimized in any way. In figure 5.15, the accuracy of the MQ PDECB method and the standard collocation method for the problem are shown over a range of shape parameters. With the MQ PDECB method, the amount of work is larger due to the larger matrix size and in this example its smallest maximum error maximum error is several orders of magnitude larger than that of the standard collocation method. The MQ PDECB method does have a smaller error over a large range of the shape parameter, but in order to make an better comparison, the number of total centers should be adjusted so that the same size matrices are worked with in each method. The Matlab code that implements the MQ PDECB method for this problem is in listing 5.7. This example and the conflicting results ([69],[129]) in the literature indicate that the effectiveness of the MQ PDECB method versus the standard method may be problem problem dependent or dependent in some way on the particular locations of the centers that are added outside the domain. The NaK boundary treatment in section 2.8 may be a more effective method to reduce boundary region errors.

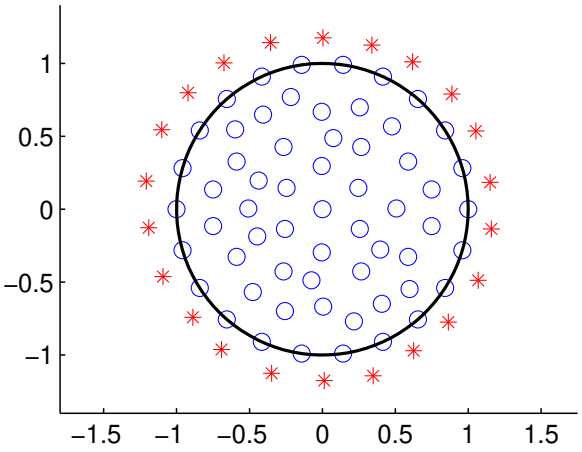

Figure 5.14: 22 additional centers (red *) added for boundary collocation to the 60 centers (blue circles) in figure 3.1.

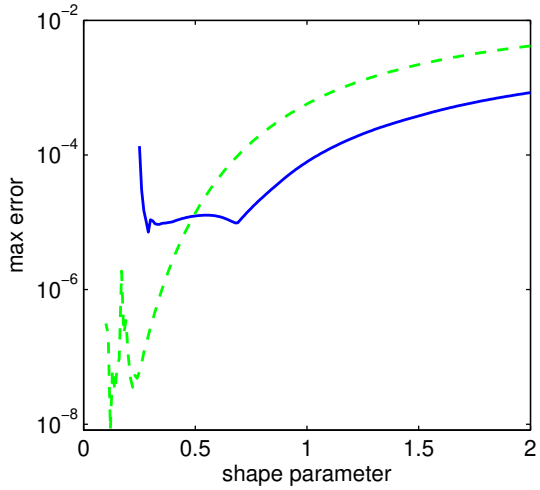

Figure 5.15: Standard collocation method error (green dashed) and MQ PDECB errors (solid blue) versus the shape parameters for problem (3.10)

```
1
2   Nb = 22;       %  boundary centers/exterior centers
3    N = 60;       %  interior and boundary centers
4   Np = 82;       %  total number of centers: interior, exterior, and boundary
5                  %  the 22 boundary centers are in 1:22
6   shape=0.6;
7
8   centers = dlmread('centersUnitCircle60BoundryCollocation.txt',' ');
9   x = centers(:,1);  y = centers(:,2);
10
11  u = 65./(65 + (x - 0.2).^2 + (y + 0.1).^ 2);              % exact solution
12  f = 130./(65 + (x-0.2).^2 + (y+0.1).^2).^3.*(2.*x-0.4).^2 ...
13    - 260./(65 + (x-0.2).^2 + (y+0.1).^2).^2 + 130./(65 ...
14    + (x-0.2).^2 + (y+0.1).^2).^3.*(2.*y+0.2).^ 2;
15
16   H = zeros(Np,Np);  rx = zeros(Np,Np);  ry = zeros(Np,Np);  r = zeros(Np,Np);
17
18   f(1:Nb) = u(1:Nb);                            % Dirichlet Boundary conditions
19
20   for i=1:N
21     for j=1:Np
22        rx(i,j) = x(i) - x(j);
23        ry(i,j) = y(i) - y(j);
24        r(i,j) = sqrt( rx(i,j)^2 + ry(i,j)^2 );
25     end
26   end
27
28   for i=1:Nb
29     for j=1:Np
30        rx(N+i,j) = x(i) - x(j);
31        ry(N+i,j) = y(i) - y(j);
32        r(N+i,j) = sqrt( rx(i,j)^2 + ry(i,j)^2 );
33     end
34   end
35
36       H(1:Nb,:) = mq(r(1:Nb,:),shape);
37       Hxx = mqDerivatives(r(Nb+1:Np,:),rx(Nb+1:Np,:),shape,2);
38       Hyy = mqDerivatives(r(Nb+1:Np,:),ry(Nb+1:Np,:),shape,2);
39       H(Nb+1:Np,:) = Hxx + Hyy;
40
41         alpha = H\f;
42             B = mq(r,shape);
43            uh = B*alpha;
44
45        errror = norm(u(1:N)-uh(1:N),inf);
```

Listing 5.7: poissonBoundaryCollocation.m

5.10 Chapter Summary

In this chapter tools and techniques that can be used to augment the basic MQ approximation method have been examined. The connections the MQ method shares with polynomial and wavelet methods have been discussed as well.

While few theoretical results exist for the MQ method when using variable shape parameters, it is well documented in applications that using a variable shape parameter strategy often results in improved accuracy. When the MQ method is used to approximate functions with discontinuities, the spectral accuracy of the method is lost and Gibbs oscillations appear in the numerical approximation. Postprocessing methods can be used to lessen the effects of or

even remove the Gibbs oscillations, or a variable shape strategy can be used to prevent the oscillations for occurring. As with all other global methods, with large N the MQ method can be very computationally expensive. Some techniques to reduce the computational costs were listed in the previous chapter. In this chapter the finite difference mode MQ method is introduced which can reduce the computational costs of the global method and can often be nearly as accurate as the global method. Functions featuring steep gradients or detailed localized features are best approximated with adaptive methods that select center locations based on features of the underlying function. Integrated MQ basis functions can be used to potentially achieve the same accuracy as their non-integrated counterparts but with smaller N. Boundary collocation is a technique for PDE problems in which both the PDE and the boundary conditions are collated on the boundary, rather than just the boundary conditions. Boundary collocation can potentially result in a more accurate solution.

Chapter 6

Further Development of the MQ method

The previous chapters have laid out the basics of the MQ approximation method for scattered data approximation and for the numerical solution of PDEs. In this chapter, several recent, more advanced, applications and techniques are described.

6.1 Simplifications of nonlinear time-dependent equations

Many of the commonly used PDEs in engineering and science are nonlinear; depending in which reference frame these governing are solved, some simplifications can be obtained. PDEs can be represented either in the fixed Eulerian frame, a moving frame in which points move at the fluid or elastic solid displacement velocity (Lagrangian frame), or a frame moving with the characteristic velocities. In finite difference, finite element, and finite volume methods, upwind differencing is used in the Eulerian representation for numerical stability; however, in the Lagrangian representation, no upwinding is necessary. To use mesh-based schemes, the physics inherent in the PDEs are modified in order to use these classical methods such as applying non-physical mesh stiffeners and excessive viscosity; the calculations need to be stopped to repair the meshing whenever the mesh has poor cell aspect ratios, exhibits hourglassing, etc. The primary motivations for using the Lagrangian formulation is that the artificial diffusion arising from upwind differencing in the

Eulerian schemes is eliminated and material interfaces and/or contact surfaces move in time with the fluid velocity. However, shocks and rarefaction fans are translated relative to the fluid velocity. In the Lagrangian representation, the fluid flow can produce cells with very poor cell-aspect ratios, or even negative volumes. In two or higher dimensions, one can question whether a meshless solution method can permit simplifications and higher accuracy.

Consider a subdomain, Ω_σ, that is bounded by either a physical or discontinuous interface, the hypothesis that is to be tested is:

1. One can find a rotated, moving frame in which the dependent variables are separable in space and time, where the spatial dependence arises from the choice of basis function and the time dependence arises from the expansion coefficients, χ(t).

2. There exist local rotational and translational transformations that can linearize the nonlinear Eulerian frame ideal gas Euler equations, and within the MQ-RBF context, transforms the nonlinear PDEs into linearized ordinary differential equations (ODEs) that have exact solutions.

3. By solving a sufficiently dense set of simplified localized PDEs at sample points, $x \in \Omega_\sigma$, a very good global approximation within Ω_σ can be obtained.

This hypothesis will shown to be valid on a 2D problem.

6.1.1 The 2d Euler equations

The hypothesis that was tested and validated is that one can find local rotational and translation transformations upon the nonlinear fixed frame Euler equations that simplifies the numerical solution procedure, provided the distribution of sampling data centers is sufficiently dense. In the fixed frame, the strong conservative form of the Euler equations in 2D are:

$$\frac{\partial U}{\partial t} + \frac{\partial F^1}{\partial x_1} + \frac{\partial F^2}{\partial x_2} = 0 \tag{6.1}$$

The vector of dependent variables, and fluxes are given in the follow order:

$$
\begin{aligned}
U &= [\rho, \rho u_1, E, \rho u_2]^T && (6.2)\\
F^1 &= [\rho u_1, (\rho u_1^2 + p), u_1(E+p), \rho u_1 u_2]^T && (6.3)\\
F^2 &= [\rho u_2, \rho u_1 u_2, u_2(E+p), (\rho u_2^2 + p)]^T && (6.4)
\end{aligned}
$$

and u_1 and u_2 are the x_1 and x_2 components of the fluid velocity, respectively. The total energy density, E, (Joules/$m3$), is related to the ideal gas pressure by

$$p = (\gamma - 1)\left[E - \frac{1}{2}(\rho u^2\right]. \tag{6.5}$$

The ideal gas has constant specific heat ratio, γ, and the adiabatic sound speed is $a = \sqrt{\gamma p/\rho}$.

Assume that with a physical subdomain, Ω_σ, that the dependent variables and the flux components are discretized at N data centers, $(x_1, x_2, \ldots, x_N) \in \mathcal{R}^2$ at some time t^n. These data centers represent a finite sampling of the continuous dependent variable in any subdomain. The notation used here is the subscript i refers to a variable at the location, x_i, $i \in [1, N]$. The superscript, k, refers to the k^{th} dependent variable or its flux component, and the superscript, n, or $n+1$, refers to the dependent variables and flux components at the time, t^n or t^{n+1}.

6.1.2 Local transformations of the Euler equations

The idea of rotating the Euler equations is an extension of the previous work of Levy, Powell, and VanLeer [134] and Kontinos [126]. These authors showed that one cannot simultaneously diagonalize *all* of the conservation equations because the momenta components are not scalars. However, three of the complete set of 2D Euler equations can be made to appear as local one-dimensional PDEs, see Kansa [120]. In three dimensions, one can perform either two consecutive rotations into the local principle momentum, or perform a 3D rotation in terms of the Euler angles.

At an interior point x_i rotate the momentum (fluid velocity) components by an angle, $\theta_i = tan^{-1}(u_{2,i}/u_{1,i})$ and rotate the *local* coordinate system centered at x_i by θ_i to form $\{x_1', x_2'\}$. Then, the local mass, principal momentum, and total energy conservation equations appear to be locally one-dimensional when one of the transformed coordinate axis, x_1', is aligned with the principal momentum, so $\rho u_2' = 0$ along the coordinate, x_2'. In this system:

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_1')}{\partial x_1'} = 0 \tag{6.6}$$

$$\frac{\partial (\rho u_1')}{\partial t} + \frac{\partial (\rho u_1' u_1' + p)}{\partial x_1'} = 0 \tag{6.7}$$

$$\frac{\partial E}{\partial t} + \frac{\partial [u_1'(E+p)]}{\partial x_1'} = 0. \tag{6.8}$$

The local angular momentum equation

$$m\left[\frac{\partial \theta}{\partial t} + u_1' \frac{\partial \theta}{\partial x_1'}\right] + \frac{\partial p}{\partial x_2'} = 0 \tag{6.9}$$

is strictly two-dimensional. In three dimensions, the rotations into the principal momentum direction produces three 1D PDEs (mass, principal momentum, and total energy density equations) and two 2D angular momentum density equations.

6.1.3 Local translations upon the rotated Euler equations

In the local rotated coordinate system, $\{x_1', x_2'\}$, the conservation of mass, principal momentum, and total energy densities at each $\mathbf{x}_1'$ are 1D. Einstein asserted that all the laws of physics are equally valid in all inertial frames, whether the frame be fixed or moving at a *constant* velocity. In numerical simulations, one assumes that within a time step, Δt, a snapshot of the physics is taken in a sufficiently small time interval so that the motions are approximately constant.

The physics of the strong form of the Euler equations (6.1) are unchanged if they are written as:

$$\frac{\partial U}{\partial t} + \lambda_1 \frac{\partial U}{\partial x_1'} + \lambda_2 \frac{\partial U}{\partial x_2'} + \left[\frac{\partial F^1}{\partial x_1'} - \lambda_1 \frac{\partial U}{\partial x_1'}\right] + \left[\frac{\partial F^2}{\partial x_2'} - \lambda_2 \frac{\partial U}{\partial x_2'}\right] = 0. \tag{6.10}$$

A very convenient set of velocities to use is found by choosing (λ_1, λ_2) such that both quantities within the square brackets vanish:

$$\frac{\partial F^1}{\partial x_1'} - \lambda_1 \frac{\partial U}{\partial x_1'} = 0$$

$$\frac{\partial F^2}{\partial x_2'} - \lambda_2 \frac{\partial U}{\partial x_2'} = 0.$$

A similar process can be performed in 3D or 4D. One can relate the fixed frame conservation equations to a Galilean or Lorentz transformation in which the data centers are permitted to move at constant "characteristic" velocities. Then the total time derivative operator is:

$$\frac{d}{dt} = \frac{\partial}{\partial t} + \lambda^k \cdot \nabla' \tag{6.11}$$

and we have

$$\frac{dU}{dt} = \frac{\partial U}{\partial t} + \lambda_1 \frac{\partial U}{\partial x_1} + \lambda_2 \frac{\partial U}{\partial x_2} = 0. \tag{6.12}$$

It is important to note that a transformation that transforms a nonlinear PDE into a linear PDE is valid at a local point, (x_1', x_2'), but not globally within the subdomain, Ω_σ. However, it is hypothesized that if there exists a "sufficiently" dense set of sample points within Ω_σ, these local simplifying transformations can accurately define the numerical solution over a subdomain, Ω_σ. The validity of this hypothesis will be demonstrated later. Linear combinations of these 1D PDEs can be combined to obtain the compatibility equations. In the local rotated and translated reference frame, the compatibility equations are ordinary differential equations along

$$\frac{d\mathbf{x}^k}{dt} = (\mathbf{e}_1 \lambda_1^k + \mathbf{e}_2 \lambda_2^k)$$

where $\lambda_1^k = (u_1' - a, u_1', u_1' + a)$ and $\lambda_2^k = 0$ for $k = 1, 2, 3$. See Kansa, Power, Fasshauer and Ling [125] for more details.

In compact notation, each compatibility equation in the local rotated, translated frame can be written as exact differentials:

$$\frac{d\mathbf{Q}}{dt} = \mathbf{W}\frac{d\mathbf{U}}{dt} = 0, \quad \text{along} \quad \frac{dx'}{dt} = \lambda \quad \text{over}\, \Omega_\sigma \backslash \partial\Omega_\sigma \tag{6.13}$$

where $\mathbf{W}$ is a matrix relating the values of $\mathbf{U}$ to the values of $\mathbf{Q}$.

In summary, in 2 or 3D, it is possible to find *local* rotations for the Euler PDEs, that diagonalizes the principal momentum, mass, and total energy equation, but not the angular momentum equations. In addition, by finding a *local* translational frame in which the rotated PDEs appear to be exact linear differentials, the solution procedure is simplified. Note that the nonlinear PDEs are linearized locally and the nonlinearities are placed in the "characteristic" velocities. Rather then solving the difficult

nonlinear Euler PDEs in the fixed frame, these *local linearized* transformed compatibility equations will be solved. The validity of this approach with the MQ RBF method has been successfully applied in Kansa [121] and Kansa, Aldredge, and Ling [122]. The construction of the interpolation and flux matrices in the rotated, translated frame are given in detail in Kansa [120, 125]. The boundary conditions for sub-sonic or super-sonic inflow and outflow are discussed in detail by Laney [127].

6.2 Exact Time Integration of the Exact Differentials

The starting point of the time integration process is interpolation. Given the initial conditions in which the mass, momentum, and total energy densities $\{U^k\}$ are specified at time $t = 0$, the initial values of the expansion coefficient $\{\chi^k(t = 0)\}$ are calculated that correspond to $U^k(\mathbf{x}, 0)$ with $\mathbf{x} \in \Omega_\sigma$.

Since mass, momentum components, and total energy are extensive quantities, the mass, momentum component, and total energy densities must be integrated over the subdomain, Ω_σ. Because it is not possible to find closed form integrals for irregular boundaries, a combination of analytic integration and Gauss-Legendre numerical integration schemes, as in Kansa, Aldredge and Ling [122] are employed. Matrices in each Ω_σ for the interpolation and fluxes in the rotated moving frame are

$$\begin{bmatrix} \mathbf{C}_{ii} & \mathbf{C}_{ib} \\ \mathbf{C}_{bi} & \mathbf{C}_{bb} \end{bmatrix} \begin{bmatrix} \chi_i^k \\ \chi_b^k \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_i^k \\ \mathbf{Q}_b^k \end{bmatrix} \tag{6.14}$$

where $\chi^k = [\chi_i^k, \chi_b^k]^T$ and $\mathbf{Q}^k = [\mathbf{Q}_i^k, \mathbf{Q}_b^k]^T$. The subscripts i refer to the interior, and b refer to the boundary. At any time t a point $\mathbf{x}_i'$ is advanced by the principal fluid velocity, $\mathbf{u}_i' = \mathbf{m}_i'/\rho_i$, to a position

$$\mathbf{x}_i^{(n+1)\prime} = x_i^{n\prime} + \Delta t \mathbf{u}_i'.$$

Then, at $\mathbf{x}_i^{(n+1)\prime}$, using the characteristic velocities, the loci of the intersections

$$\mathbf{x}_{i,k}^n = \mathbf{x}_i^{(n+1)\prime} - \Delta t \lambda_{\mathbf{i}}^{\mathbf{k}}$$

are formed.

The components of the $\mathbf{C}$ matrix are explicitly functions of space. For simplicity, denote

$$\nabla \mathbf{C} = \mathbf{e}_1' \frac{\partial \mathbf{C}}{\partial x_1'} + \mathbf{e}_2' \frac{\partial \mathbf{C}}{\partial x_2'}$$

formed by taking the appropriate x_1' and x_2' partial derivatives of the basis functions. In the initial value problem, all the expansion coefficients $\{\chi^k(t=0)\}$, in Ω_σ are found by specifying the initial values of $\mathbf{U}^j(x, t=0)$, and the conservation constraint. It is simple to construct such matrices evaluated at the set $\{\mathbf{x}_{i,k}^n\}$. Once the set of initial expansion coefficients are known, then a compatibility variable, $\mathbf{Q}^k$, can be reconstructed over Ω_σ as

$$\mathbf{Q^k}(\mathbf{x}, t) = \mathbf{C}(\mathbf{x})\chi^k(t), \text{ for all discrete } \mathbf{x} \in \quad \Omega_\sigma \quad \text{and all t } \geq 0.$$

The compatibility PDEs, now in ODE form, have the general structure:

$$\begin{aligned} C(x)\frac{d\chi^k}{\partial t} + [\lambda \cdot \nabla C(x)]\chi^k(t) &= f_{int}^k \quad \text{over} \quad \Omega_\sigma \backslash \partial\Omega_\sigma \\ BC(x)\chi^k(t) &= f_b^k \quad \text{on} \quad \partial\Omega_\sigma \end{aligned}$$

It is assumed that the primary time dependency with RBFs arises from the expansion coefficients, $\chi^k(t)$; if data centers move in time, then the spatial dependency is implicit in time. The loci of the boundary characteristics depend upon whether the flow direction is positive or negative, and sub-or super-sonic, see Laney [127]. For the incoming or outgoing characteristic, one usually specifies a $U^j(\mathbf{x}_{bound}, t)$ and rewrites the appropriate compatibility equation for the other components, $U^{k\neq j}$, along the appropriate characteristics. Consistent boundary conditions can be determined from the loci of the incoming or outgoing "characteristics". Between time steps, consistent boundary conditions require the incoming or outgoing "characteristics" to lie within, not outside of the domain Ω_σ.

The conservation constraint for area, mass, momentum components, and total energy is built-in the setup. Define a new matrix, $\mathbf{A}$, to be a matrix over Ω_σ that consists of both the interpolation terms and the flux terms over $\Omega_\sigma \backslash \partial\Omega_\sigma$ and on $\partial\Omega_\sigma$ written as:

$$\mathbf{A} = \mathbf{C}^{-1}(\lambda^k \cdot \nabla \mathbf{C}).$$

The exact solution of the system of ODEs for χ^k includes the homogeneous and particular solution:

$$\chi^k(t+\Delta t) \quad = \quad \chi^k(t)\mathbf{expm}\left[-\int_t^{t+\Delta t} \mathbf{A}(\tau)d\tau\right]$$

$$+ \quad \mathbf{expm}(-\mathbf{A}t) \int_t^{t+\Delta t} \mathbf{expm}(\mathbf{A}\tau)\mathbf{f}^k(\tau)d\tau.$$

The expression **expm** is the built-in Matlab routine that calculates the exponential matrix. The method is stable if the amplification matrix

$$\mathbf{expm}\left[-\int_t^{t+\Delta t} \mathbf{A}(\tau)d\tau\right]$$

is such that its maximum eigenvalue ≤ 1. Several factors were observed that are required for stability: (1) Well-posed boundary conditions and (2) The **C** matrix must be well-conditioned. Stability can be achieved for larger values of Δt if a two-or-three point Gauss-Legendre integration scheme is used, rather than assuming the matrix **A** is constant within a prescribed Δt. In contrast to the finite difference time-marching scheme, the CFL condition does not necessarily guarantee that the maximum eigenvalue ≤ 1. If **C** is poorly conditioned, then the amplification matrix can yield maximum eigenvalues ≥ 1 for all $\Delta t \leq \Delta t_{CFL}$. In general, it appears that the theoretical conditions for stability are more complicated than the simple CFL rule.

As with traditional Lagrangian schemes, the data centers will move within a subdomain, perhaps leaving some subregions very sparsely represented, and congregating in other regions. Wave breaking can be detected by monitoring the characteristic velocities, $\{\lambda_{\mathbf{i}}^{\mathbf{k}}\}$; if all these velocities have the same sign, then there is no wave steepening. If some velocities have opposite signs, then there will be a time at which data centers will coalesce, and a discontinuous curve can be inserted, forming two or more subdomains. However, whenever data centers are added, deleted, or rearranged, the interpolation process onto a more favorable distribution must be performed in a strictly conservative manner, so that the extensive quantities of mass, momentum components, and total energy are strictly conserved before and after interpolation. This conservation is strictly guaranteed because conservation constraints are built into the formation of both the interpolation and flux matrices.

6.3 The MQ and the Level Set Method

The level set method is a numerical technique for capturing interfaces whose topology evolves in time on a fixed grid in $\mathcal{R}^d$. The interface located at x^* is the locus at which the level set function $G(x,t)$ is such that $G(x^*,t) = 0$.

The propagation of the level sets are governed by the following hyperbolic PDE

$$G_t + v(x,t) \cdot |\nabla G| = 0. \tag{6.15}$$

The numerical solution of the level set equation requires sophisticated techniques. Simple finite difference methods fail quickly without upwinding. The level set method does not guarantee the conservation of the volume and the shape of the level set in an advection field that does conserve the shape and size in a uniform or rotational velocity field. Instead, the shape of the level set may get severely distorted and the level set may vanish over several time steps. For this reason, high-order finite difference schemes are generally required, such as high-order essentially non-oscillatory (ENO) schemes, and even then, the feasibility of long-time simulations is questionable. Level set and volume-of-fluid methods handle the topology changes due to non-smooth velocity fields at the interface by introducing viscosity to the hyperbolic equations. Sethian and Smeraka [193] warn one should avoid re-initialization too often since it introduces errors in the position of the front, x^*. The signed distance function helps minimize loss of conservation of mass. Further sophisticated methods to deal with this difficulty have been developed, e.g., combinations of the level set method with tracing marker particles advected by the velocity field.

Kansa, Aldredge and Ling [122] attempted to simulate the previous 2D combustion calculations of Aldredge [1] with 2D counter-rotating vortices in an infinitely periodic domain with MQ-RBFs and the level set method to determine whether the higher convergence rates reported with RBFs could reduce the CPU time from 14 hours using the upwinded finite difference method. The governing equation can be written as:

$$(\rho G)_t + \nabla \cdot [u - nS_n \delta(x_f) \rho G] = 0 \tag{6.16}$$

where the u and v components of the vortical flow are:

$$u(x,y) = 2A\cos(2\pi x)\sin(2\pi y)$$

and

$$v(x,y) = 2A\sin(2\pi x)\cos(2\pi y).$$

A is the root-mean-square intensity of either component of velocity fluctuation, ρ is the burned or unburned gas density, S_n is the local normal rate of propagation of the flame into the reactants. The vector n is the normal unit

vector from the flame front x_f and $\delta(x_f)$ is a delta function that is one at $x = x_f$ and zero elsewhere.

By combining the level set method with the MQ method, it soon became apparent that the level set method not only required a considerable amount of nonphysical viscosity but required re-initialization too often for stability. This requirement was contrary to what Sethian and Smeraka [193] warned.

It was argued in Kansa, Aldredge, and Ling [122] that since the levels near zero wanted to coalesce naturally, then the zero level should be a discontinuous function represented as a product of a Heaviside function in the normal propagation direction and a piecewise continuous function in the tangential direction that is approximated by a periodic MQ function having the following form:

$$\phi_j(x) = \left[1 + \sin^2\left(\frac{\pi(x - x_j)^2}{2c_j^2}\right)\right]^{\beta}. \tag{6.17}$$

Define the flame front position in terms of the unit normal and tangential unit vectors,

$$x_f = \xi e_\xi + \kappa e_\kappa.$$

Then the normal and tangential positions at the front are related implicitly:

$$F = \xi_j = \sum \phi_j(\kappa_i)\gamma_j = 0.$$

By finding the partial derivatives, F_ξ and F_κ, one can readily find the rotational angles to construct the normal and tangential unit vectors that relate $x_f = (x_f^1, x_f^2)$ to (ξ, κ). Then the level at the flame front is governed by:

$$(\rho G)_t + \nabla \cdot [u - nS_n\delta(x_f)\rho G] = 0. \tag{6.18}$$

The normal velocity component of the front is determined by the Rankine-Hugoniot jump conditions at the discontinuity. The tangential velocity component is determined by the angle of rotation.

The gas density in the burned zone (or subdomain) is different from the gas density in the unburned zone. The natural way to model this problem is to define two subdomains, Ω_b (burned gas) and Ω_u (unburned gas). Because the reactant speed S_n is a delta function, the governing equation becomes

$$(\rho G)_t + \nabla \cdot [u\rho G] = 0 \tag{6.19}$$

in either Ω_b or Ω_u where u is the vortical flow. The dependent variable is ρG that is approximated by MQ-RBFs as

$$\rho G(x,t)^k \approx \sum \phi_j(x)\alpha_j^k(t).$$

The superscript k is either burned or unburned. The data centers in the burned or unburned regions move in time. At each interior point, local rotational and translational transformations (section 6.1) are applied so the PDE is converted to a local exact differential, and the exact time integration (section 6.2) method is used to find the advanced time solution. To prevent interior points from moving too far, leaving holes, etc., a strictly conservative interpolation scheme is enforced in both the burned and unburned subdomains.

The flame front motion is influenced by both the burning speed, S_n, and the u and v components of the vortical flow. The flame front flattens, then distorts. At time $t = 0$, the flame front was a vertical line centered at $x = 0$. Because of this motion, and the properties of MQ, the modified Greedy Algorithm (section 4.7) was used. The set of evaluation and data centers with various MQ shape parameters were used in an over-determined systems. The trial set of shape parameters are:

$$\left\{c/2^0, c/2^1, \ldots, c/2^k\right\}.$$

At various time steps, the Greedy Algorithm chooses the data centers, evaluation points, and shape parameters for the front itself, and the burned and unburned subdomains. The chosen shape parameters change in time. The center locations and the position of the flame front at time $t = 0.091058$ are shown in figure 6.1.

The benefits of modeling the flame front as a product of a Heaviside function in the normal propagation direction and a piece-wise continuous function are numerous:

1. Without a mesh, front tracking is very simple to implement

2. There is no need to construct a conformal grid needed for a front with possibly very small cell sizes

3. There is no need to change the physics by introducing nonphysical numerical viscosity or large surface tension

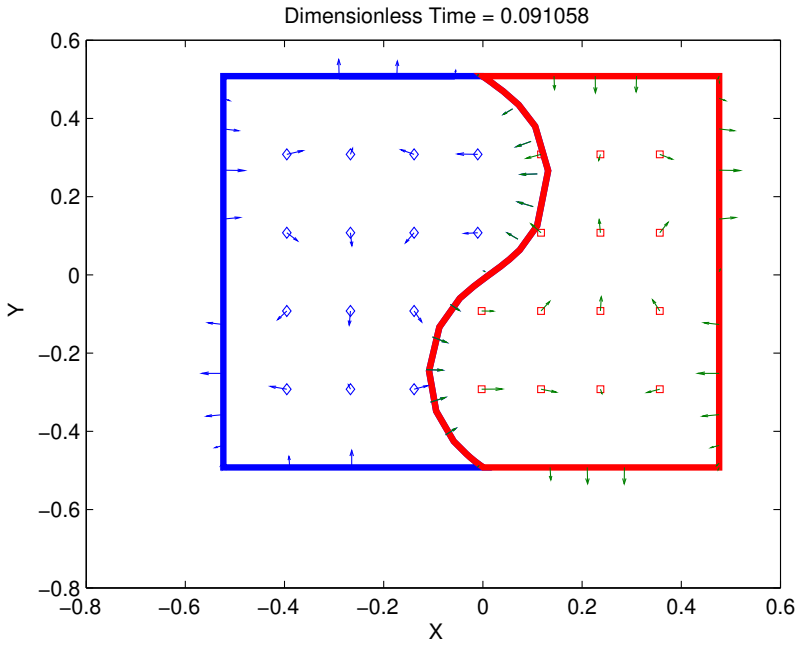

Figure 6.1: Flame front at $t = 0.091058$ with internal center locations and velocity vectors.

4. There is no need to re-initialize to prevent levels from coalescing

5. There are no very small cell sizes or other complications.

The original finite difference level set method required over 14 hours on a main frame computer to execute. The simplified front tracking scheme using the MQ required only 120 seconds on a desktop computer using un-optimized Matlab code. The results were in excellent agreement with prior calculations of Aldredge [1]. Additional details of simulation can be found in [122].

Chapter 7

New Frontiers: High dimensional PDEs

This chapter is distinct from previous chapters because higher dimensional PDEs, whether with mesh-based methods or meshless radial basis functions, is basically unchartered territory. The chapter will present some representative high-dimensional PDE problems, survey what has been done with mesh-based and stochastic methods, and present suggestions that might make such challenging problems workable.

7.1 Physically important problems not addressed by current methods

There are many important applications in biology, finance markets, chemistry and physics that involve PDEs in higher dimensions. Examples of high dimensional partial differential equations in $\mathcal{R}^d$ are:

1. General relativity and black hole formation.
2. Dirac relativistic quantum mechanics.
3. The 6 dimensional Boltzmann's equation in which the independent variables are the three spatial dimensions and the three components of velocity.
4. The Fokker-Planck equation that describes the evolution of the probability density function.

5. The d-different substances in molecular biology that influence a cell's chemical balance.

6. The ab initio quantum mechanical molecular calculations in which each atom of a molecular containing M atoms has 3M degrees of freedom.

7. The d-dimensional Black-Scholes price options of d-stocks or other commodities.

Presently, there are a few papers using RBFs dealing with the Black-Scholes equations, see [111], [70], and [174].

Even with the newly developing supercomputers, both computer memory and execution time become severe limiting factors in solving such multi-dimensional PDEs. One must be aware that mesh generation over irregular 3D domains alone is a very time-consuming task. In higher dimensions, mesh construction, defining the connectivity relations, differentiation, integration, etc. may not be even practical beyond three dimensions.

7.2 Curse of dimensionality

Bellman [17] discussed the curse of dimensionality to describe the computational problem of dealing with higher dimensions on tensor product grids. Assume the computational domain $\Omega \in \mathcal{R}^d$ is a unit hypercube in d dimensions, and along each coordinate, there is a uniform discretization, h; the number of grids is $N \sim \mathcal{O}\left(h^{-d}\right)$. Problems with practical applications are not readily treated on a unit hypercube.

There are three current widespread practices that are used to solve multi-dimensional PDEs: Operator (Dimensional) splitting, Adaptive Sparse Grid (ASG) methods, and Monte Carlo (MC) and the quasi Monte Carlo (QMC). Each method has its limitations in high dimensions.

7.3 Operator Splitting

There are several operator (dimensional) splitting methods in use, see [57]. They conclude that these splitting schemes are either not accurate enough or not parallelizable on operator level or only parallelizable at a high cost. They demonstrated that iterative splitting schemes perform well if the advection is

parallel to a grid direction, but fails to converge when the flow is skew to the grid directions. Reference [172] examined the errors introduced by operator splitting techniques in air quality models, and in general, the results are only first order accurate. Symmetric and non-symmetric operator splitting does not provide significant difference in accuracy.

7.4 Multigrid and Adaptive Sparse Grid methods

For higher dimensions there is an additional disadvantage of regular refinement in each direction, because the number of the degrees of freedom increases very fast when more levels of refinement are introduced. In [88], the authors studied numerical integration in high dimensions and showed that in many multi-dimensional problems, not all dimensions are equally important, and the overhead involved in determining and refining the important dimensions is considerably large. They developed a dimension–adaptive algorithm that tries to find the important dimensions automatically and places more points in those dimensions. The number of indices in the index sets can become very large for difficult high-dimensional problems.

7.5 Monte Carlo and Quasi-Monte Carlo methods

Monte Carlo or pseudo-random methods suffer from one severe shortcoming: they converge inversely proportional to the square root of the number of particles. The quasi- Monte Carlo (QMC) method achieves faster convergence by sampling from carefully chosen deterministic points; hence it is quasi-random. However, [173] showed the lack of randomness in quasi-Monte Carlo methods is a distinct disadvantage, since it causes aliasing and precludes error estimation. One remedy is adaptive sampling; one takes more samples where the integrand has the most variation. The main disadvantage of adaptive sampling is that it can introduce bias. Other problems with adaptive sampling is that it is not very effective for high dimensional problems and this method requires that a substantial number of samples must be taken, in order to estimate the covariance matrix with any reasonable

accuracy. Lastly, there are too many possible dimensions to refine.

The authors in [49] discussed QMC and found surprising results. QMC can be inferred that quasi-random sequences may exhibit cyclic behavior. For example, it might be an efficient method for dimensions 1 to 60, then diverge significantly from the theoretical result between dimensions 80 to 200 and at last converge again. Reference [87] discussed stochastic methods and sparse grid methods. However, the statistical approach becomes quickly intractable for complex problems in multiple random dimensions. The reason is the number of realizations required to acquire good statistics is usually quite large. In the standard sparse grid collocation approach, all the dimensions are treated equally. In many problems usually encountered, not all the dimensions are equally important. That is, the solution varies much more smoothly in some particular dimension than in others. This brings up the possibility of reduction in the computational effort required to solve the stochastic problem by weighting the number of sampling points in the stochastic dimensions according to the solution smoothness in that dimension. But it is not possible to know a priori which dimensions are more important. An obvious disadvantage of this strategy is that the number of points required increases combinatorially as the number of stochastic dimensions is increased.

7.6 Role of the MQ in high dimensional problems

MQ RBFs are very well suited to solve high dimensional PDEs on irregular domains because only the radial distance between any pair of points in $\mathcal{R}^d$ are required. A very irregular domain in $\mathcal{R}^d$ is first discretized by discretizing the boundary surface, then inserting interior points interior. However, both the curse of dimensionality and ll-conditioning are persistent problems that must be addressed. Some recommendations are presented address both issues, since they are inter-related. Because of the large number of discretization points possible in high dimensional PDE solutions, hybrid combinations of various reduction methods along with more complex programming requirements is a fact of life because computers will infinite memory and precision executing instantly will not be available.

7.7 Reduction of discretization points by transformations to different variables or on the dependent variables

The usual approach used by many authors in solving PDEs with RBFs is to cover the domain with a very fine discretization; such an approach is very simple to implement, but increases both CPU time and ill-conditioning. For problems in three dimensions or higher, this simple approach is not practical. The objective in solving higher dimensional problems is to obtain the best accuracy with the least amount of CPU time. One should use discretization points sparingly, to avoid the curse of dimensionality problem. There are several methods used both with RBF methods and other approaches that should be examined before blindly rushing into attempting to solve higher dimensional problems with billions of points. Some of these methods are presented below.

7.8 Test for the effective problem dimensionality

References [87, 88, 173, 49] discussed methods to determine the effective dimensionality of high dimensional problems in mesh-based and Quasi-Monte Carlo methods. Likewise, [89] introduced a functional minimization approach to determine the important independent variables, and any possible degeneracy. He allowed the data centers and expansion coefficients to be unknowns, and minimized a functional containing a transformation matrix, W, of the independent variables. The eigenvalues of W provides information about the computationally relevant independent variables, thus possibly reducing the effective dimensionality of the problem.

7.9 Transformations on the independent variables

Often, waves exhibit a wide range of wavelength scales; a bare minimum of two discretization points are needed for a wave in each direction. For very short wavelength problems, one can work in a transformed frequency space

where the frequencies are inversely proportional to wavelengths by using Fourier transforms. The inverse Fourier transform can be applied to return to regular space. Other commonly used transforms that may have computational advantages are the Laplace and Mellin transforms as well as those based upon the various kinds of Bessel functions such as the Hankel, Meijer, Kontorovich-Lebedev, and Y transforms, see [178]. Such transformations can only transform linear PDEs. Many problems of importance to biophysics, engineering, financial markets, and physics are nonlinear. In order to linearize such PDEs and work in transformed space to reduce the discretization problem in $\mathcal{R}^d$, various methods have been published to linearize nonlinear PDEs (NPDEs). Various authors, such as in [23], [51], [5], and [197] constructed invertible mappings from a system of NPDEs to a linear system of PDEs (LPDEs). The method is based on the existence of an infinite-parameter Lie group of transformations admitted by the nonlinear system. If such a group exists and certain criteria are satisfied , then using the infinite-dimensional symmetry, a transformation may be constructed to link the system of NPDEs to a system of LPDEs. These canonical variables are related to the commuting operators, and represent the linearizing variables. In [55], the author showed that two conditions play an important role: (1) The nonlinearity can be expanded in the same basis functions as the linear part. (2) The linearized parts of the NPDEs have nontrivial solutions.

7.10 Dependent Variable Transformations

An example of flattening rapidly varying functions can be found in the transformations of the time dependent Schrördinger equation for molecular reactions given by:

$$i\hbar\frac{\partial\psi(r,t)}{\partial t} = \left[-\frac{\hbar^2}{2\mu}\nabla^2 + V(r,t)\right]\psi(r,t), \tag{7.1}$$

where $\hbar$ is the Planck constant, μ is the reduced mass, and $V(r,t)$ is the electrostatic potential that includes the nuclear attraction of the electrons and the electrostatic repulsion potentials of the nuclei with each other and the electronic repulsion energies. Because molecular quantum mechanics involve $3N$ spatial dimensions, where N is the number of atoms in a molecule, it is important to reduce the number of discretization points. Quantum fluid dynamics (QFD), see [112], replaces the rapidly varying wavefunction, $\psi(r,t)$

by a slowly varying quantum density and phase space terms:

$$\psi(r,t) = \exp(iS(r,t)/\hbar)\sqrt{\rho(r,t)} \tag{7.2}$$

where ρ is the slowly varying density, and S is phase in configuration space. Define the quantum fluid velocity as:

$$v(r,t) = \nabla S(r,t)/\mu. \tag{7.3}$$

Then the electronic Schrördinger equation is written in two parts:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) = 0, \tag{7.4}$$

and

$$\mu\left(\frac{\partial}{\partial t} + v \cdot \nabla v\right) = -\nabla\left[V(r,t) + V_e(r,t)\right], \tag{7.5}$$

where

$$V_e(r,t) = -\frac{\hbar^2}{2\mu}\left(\nabla^2\sqrt{\rho}/\sqrt{\rho}\right). \tag{7.6}$$

The resulting time dependent QFD equations were solved in the Lagrangian mode using the velocity, v that move the centers to regions of higher gradients [112]. In [175], QFD was used to investigate the propagation of wave packets using adaptive grids and moving boundaries. They found a significant reduction in the number of data centers required to obtain accurate calculations by allowing points and boundaries to move as compared to the Eulerian formulation.

7.11 Translations to a moving node frame

Reference [198] studied cyclone development in tropical waters and used nested moving high resolution grids moving over fixed larger scale grids. With substantial grid motion , the characteristics show a clear asymmetric pattern for cyclone development that had not been seen for fixed grids previously. For hurricane Lili in the Gulf of Mexico, the moving grid approach reduced the number of model grid points by a factor of 6.8, and the model run time by a factor of 4, compared to a conventional fixed grid model. For applications in larger basins, these ratios are expected to favor the moving grid model even more.

In [121] and [122] rotational and translational transformations were used on the PDEs and obtained very high accuracy with a coarse discretization. The physics of the problem moved nodes to regions of steeper gradients more rapidly than the shallower gradients, and effectively provide an automatic form of node refinement. Although there is more complex programming required, there can be a substantial payoff in reducing the number of discretization points.

7.12 Solution Space Enrichment containing both continuous and discontinuous RBFs

One must distinguish between very steep, but continuous structures and a true discontinuous structure. The time-honored way of approximating shocks by rapidly varying continuous functions over a thin region becomes impractical in high dimensional space. Nonlinear hyperbolic PDEs can produce infinitesimally thin shocks and contact surfaces; whereas parabolic PDEs with small dissipative parameters such as viscosity, thermal conductivity, and diffusivity produce very thin regions of finite width. In certain physical situations, the physical parameters are sufficiently small producing very thin-regions that are to a very good approximations negligibly small.

At 1 atm pressure ($1.0133 \cdot 10^5$ Pa) and a temperature of 293 K, the shock thickness at Mach 1 is approximately $5.6 \cdot 10^{-8}$m; at Mach 2, $3 \cdot 10^{-8}$m, at Mach 2, at Mach 3 $2.6 \cdot 10^{-8}$ m, etc. In solids, the shock thickness 5-10 nm.. At 213K, the kinematic viscosity of air = $8.79267 \cdot 10^{-6}$ m^2/s, the thermal diffusivity = $1.22 \cdot 10^{-5}$ m^2/s; at 293 K, the kinematic viscosity of air = $1.5497 \cdot 10^{-5}$ m^2/s, and the thermal diffusivity = $2.3 \cdot 10^{-5}$ m^2/s. To properly model a shock in the fixed Eulerian frame, one requires mesh cell sizes of 1 nm for solids and at least 7 nm along each coordinate for solids and air, respectively. For supersonic jet whose dimensions are order of meters to tens of meters, this is impossible on even present-day supercomputers; on the scale of such an aircraft, a shock is infinitesimally thin. The long-standing approach has been to invoke artificial viscosity to smear shocks over several mesh cells; the physics is necessarily modified to accommodate the numerical schemes.

Since many problems admit solutions with continuous and discontinuous features, the solution space needs to contain both continuous and discontinu-

ous functions. Found in [50], [18], and [58] are a few of the early authors who introduced the extended finite element in which the solution space contained both continuous elements and special discontinuous elements for cracks. In [215] a new RBF approximation was developed in which discontinuous RBFs for cracks. In this method, a jump function that accounts for the displacement discontinuity along the crack faces is added on nodes whose supports are cut by the discontinuity and a branch function accounting for the near-tip crack fields is added on those nodes whose supports are partially cut by the discontinuity. In addition, a special function derivative is discontinuous through the line of discontinuity (material interface for instance) is added into the approximation space.

In [2], the method of fundamental solutions (MFS) was used to solve crack problems. In their work, they use a functional space based on shifted fundamental solutions of the PDE to account for the regular behavior of the solution far away from the crack, and enriched that functional space by adding singular particular solutions that capture the behavior in the vicinity of the cracks. In [20], the solution space was enriched to account for boundary singularities that arise when there is an abrupt change in the boundary conditions (along a smooth boundary) or if there are re-entrant corners. The usual approach to try to overcome these difficulties is to use grid refinement in the vicinity of the singularity. However, adaptive grid refinement schemes cause a significant increase in computational cost and their efficiency is not always satisfactory. Their method is based in enlarging the functional space spanned by the RBF by including new functions that capture the discontinuities in boundary conditions, and yields significant improvements in the accuracy of the solution for the problems considered. In reference [122], the solution space for combustion was enriched by approximating a flame as an infinitesimally thin discontinuous curve separating the burned and unburned regions under vortical flow.

7.13 Domain Decomposition and parallel computer implementation

Any serious calculations of high dimensional PDEs will need to be performed on a parallel computer, especially considering the discretization requirements. A very large problem is decomposed into many smaller manageable problems

each of which is performed in parallel, rather than sequentially. The domain is partitioned into subdomains that may or may not overlap slightly. The goal is to calculate as many sub-problems of nearly equal size in parallel and to minimize inter-processor communication using the message passing interface (MPI). The general goal is to assign each processor an equal amount of work and to minimize the amount of communication between processors by essentially minimizing the surface area of the subdomains, since the communication time among processors is generally orders of magnitude slower than calculations within a processor. Each processor is then responsible for updating the unknowns associated within its subdomain only.

Domain decomposition methods have been successfully implemented in solving both elliptic and time-dependent problems with RBFs (section 4.6). Recently in [202], a dilute incompressible suspension polymer solution in a Newtonian solvent was considered. They solved the Stochastic Differential Equations (SDEs) for the polymer suspension and the PDEs governing the flow, and applied the parallel DDM techniques to both the macroscopic and microscopic components. Hence, the parallel domain decomposition method increases the throughput, and, at the same time, removes the problem of ill-conditioning of the system matrix associated with the RBF method. The convergence of the scheme can be affected by the number of subdomains; the results obtained are very good, judging by the convergence measure. They achieved high efficiency (≥ 60 when using 20 CPUs).

7.14 Summary

High dimensional RBF-PDE calculations will require the smart utilization of computer resources. Such calculations will require more complex hybrid algorithm development in combination with moving nodes, transformations on the independent and dependent variables, enrichment of the solution space, determining those dimensions are most important, using domain decomposition, parallel computer implementation, and high precision arithmetic. Radial basis functions offer a convenient and powerful tool by which to solve high dimensional PDEs on irregular domains.

Chapter 8

Afterword

The focus of this manuscript has been on introducing the MQ RBF method as a tool for solving scattered data interpolation problems and as a tool for solving PDEs. For the majority of the material presented, Matlab code has been included that implements the algorithms and examples. The purpose of including the Matlab code is twofold. One is so that the reader may instantly begin to experiment with and use the algorithms and examples. The other purpose is to foster the idea of ***reproducible research***. Instead of just describing the results of example computations, we have in most cases also included the source code that produced the examples. A listing of all Matlab programs is in appendix A.

Research in RBF methods continues to be very active. The RBF method is relatively new (about 40 years old) and it is far from being fully developed. But at the same time, it is mature enough to have been successfully applied in a large number of applications. In closing, we briefly mention some open problems as well as some areas where the application of RBF methods seems promising.

8.1 Open Problems

The first open problem is how to accurately and efficiently evaluate a RBF method when using a small shape parameter ε, a small minimum separation distance, and with N possibly large. Is there a way, that is applicable and efficient with large N, to bypass the ill-conditioned linear systems and accurately evaluate a RBF approximation for any value of the shape parameter in

a computationally efficient manner. The holy grail would be the discovery of an algorithm for RBF methods that is similar to the Fast Fourier Transform and its $\mathcal{O}(N \log N)$ operation count for pseudospectral methods. Chapter 4 chronicles the progress in this direction that has been made so far.

A second open problem is eigenvalue stability for time-dependent PDEs (section 3.2.6). Can a way be found to apply boundary conditions, or perhaps some other modification made, in order to ensure eigenvalue stability for time-dependent problems? Perhaps penalty boundary conditions that have been successfully used to enhance the stability properties of pseudospectral methods [101] can be developed for RBF methods as well. Without the application of boundary conditions, the MQ differentiation matrix (2.25) is invertible [91]. This is in contrast to other methods such as finite differences and pseudospectral methods whose differentiation matrices are singular. Does the invertibility of the MQ differentiation matrix play a role in the eigenvalue stability issues of the MQ method for time-dependent PDEs?

8.2 Promising Areas for RBF applications

Many problems in science and engineering are not well-posed due to issues of accessibility and cost of measurement. Additionally, the boundary positions and/or boundary conditions may not be known or boundary data may be missing. Inverse problems and their associated ill-posed boundary value problems arise in areas such as non-destructive testing, imaging, electrical impedance tomography, and contaminant transport in groundwater flow. Such ill-posed inverse problems can be classified as those having: (1) one or more unspecified boundary conditions, 2) missing or over-specified boundary data, 3) the boundary not being clearly define, and (4) a semi-infinite domain with Cauchy conditions prescribed only on a part of the boundary. As mentioned in section 8.1, MQ differentiation matrices are invertible without boundary conditions being applied. This indicates that the MQ collocation method may be well-suited for application to ill-posed inverse problems with missing or improperly applied boundary conditions. Some initial applications of RBF methods to ill-posed inverse problems can be found in references [110, 109, 37, 43].

In addition to ill-posed inverse problems, another promising application area for RBFs is high dimensional PDEs as discussed in chapter 7.

8.3 Other Applications and Developments

We have discussed many applications and aspects of RBF methods, but it is impossible to thoroughly discuss every application and development. In closing we briefly mention some other applications and developments of RBF methods: computing eigenmodes of elliptic operators [176], oscillatory RBFs [78, 72], RBF methods on the sphere [62, 170, 73, 79, 80, 74], divergence-free RBFs [152, 153, 171, 86], options pricing approximation [102, 103, 104, 66, 65, 174, 130], RBF quasi-interpolation [14, 105, 131, 142, 141, 40, 41], RBF Hermite interpolation [218] and symmetric collocation for PDEs [61, 180], operator splitting [8], space-time RBFs [169], solution of PDEs [39, 83, 119, 135, 151, 213, 217], compactly supported RBFs [206, 192, 30], periodic RBFs [216], moving least squares [208, 207], problems with moving boundaries [106], fast multipole methods [9, 11, 13, 45] for efficiently evaluating RBF approximations with large N, solution of stochastic differential equations [38], and basis enrichment methods to handle discontinuities and singularities [22].

Appendix B lists a significant number of additional books and papers that develop and apply meshless methods.

Appendix A

Matlab programs

Below is a list of Matlab programs that are available for download at:

http://www.scottsarra.org/math/math.html

The programs implement the examples in the manuscript. The programs that are marked with an asterisk are also included in the text of the manuscript. Each program included in the manuscript is small enough to fit on a page or less. The programs have minimal comments and only on a few occasions are they referred to in the text. This was deliberate, in order to keep the focus of the manuscript on the MQ RBF method. Some time may have to spent with the m-files in order to make them your own. It is hoped that they may be easily be modified in order to solve your own problems.

acbfPreconditioner.m* Implements the Approximate Cardinal Basis Function preconditioner (section 4.5.1)

advectionMQ1d.m* Solves the 1d advection equation in section 3.2.6.

advectionDiffusionMQ.m* Solves a time-dependent advection-diffusion equation (3.16) in one dimension using an explicit fourth-order Runge-Kutta method for time integration.

advectionDiffusionMqTrapezoid.m* Solves a time-dependent advection-diffusion equation (3.16) in one dimension using the implicit Trapezoid method for time integration.

advectionVariableShape.m Solves the advection equation (3.25) with variable shape strategy in section 5.2.

affinceSpaceSvdSolver.m* Implements the affine space solver for the RBF linear system. Section 4.4.

calculateFdStencilWeightsHeatComplex.m* Produces weights for the finite difference mode calculation of section 5.6.

costFunctionShape.m* Cost function for the leave-one-out cross validation algorithm for selecting the shape parameter in section 5.2.

cubicSchrodingerMQ.m* Nonlinear Schrodinger equation from section 3.2.4.

dtvFilter.m Implements the DTV filtering postprocessing method of section 5.5.

dtvFilterMQ.m Applies the DTV filter to postprocess the MQ interpolant of the discontinuous function 5.16.

evaluationMatrixMQ.m Returns 1d the evaluation matrix with entries given by equation (2.6)

evaluationMatrixMQ2d.m* Returns the 2d evaluation matrix with entries given by equation (2.6)

frankeProblemCenters.txt 618 centers on a quarter circle domain. The first 118 centers are on the boundary of the region and the remaining 500 centers are in the interior of the domain. Used in several examples.

frankesFunction.m Franke's function (2.8)

frankeProblemEvaluationPoints.txt 930 evaluation points on the quarter circle domain of example 2.3.

frankeProblemGeneralizedMq.m An example in section 2.10 that uses the generalized MQ (2.28) with various values of the exponent to interpolation the Franke function.

frankeProblemLocalGlobalKappa.m Compares the local and global condition numbers of the system matrix for a interpolation problem in section 2.6.

frankeProblemLsq.m* Approximates the Franke function (2.8) using a least squares method in section 2.11.

generateACBFStencils.m* Finds local stencils for the ACBF preconditioner of section 4.5.1.

generateFdStencils.m* Produces stencils for RBF finite difference mode calculations.

greedyAlgorithm.m* The greedy algorithm of section 4.7.

greedyInterpolation.m Uses the greedy algorithm of section 4.7 to select shape parameters in a 1d interpolation problem in section 4.7.

grpRbf.m* Implements the Gegenbauer post-processing method (section 5.5.3) for removing Gibbs oscillations from the MQ approximations of discontinuous functions.

grpRbfDriver.m* An example of using the Gegenbauer post-processing method.

heatComplexDomain.m* MQ collocation collocation solution of the 2d diffusion equation on an irregularly shape domain (section 3.2.5).

heatComplexDomainFdMode.m Carries of the MQ finite difference example of section 5.6.

hybridInterpolation.m* Implements the hybrid interpolation example in section 2.4. The output is shown in figure 2.6.

improvedTruncatedSvd.m* Implements the improved truncated Singular Value solver. Section 4.3.2.

interpolationFrankeExample.m* 2d interpolation example from section 2.2.

interpolationFrankeExampleLOOCV.m* 2d interpolation example where the shape parameter is selected by the leave-one-out cross validation algorithm in section **??**.

locallyAdaptiveMQ.m Approximates a discontinuous function with the locally adaptive MQ method. Section 5.5.

mq.m Evaluates the MQ and integrated MQ RBFs

mqDerivatives.m Evaluates the first and second derivatives of the MQ and integrated MQ RBFs

mqVariousShapePlots.m Produces the plot in figure 2.1.

nearOptimalCentersAnnulus.m* Example of using the geometric greedy algorithm of section 2.7 to generate near optimal, data independent, center location on an annulus.

nonlinearBvp1d.m* Solves a nonlinear boundary value problem in section 3.1.2.

nonStationaryInterpolation.m Examines the accuracy of a non-stationary interpolation example vs. an error bound. Figure 2.5.

notAKnotInterpolationUnitCircle.m Implements the not-a-knot interpolation example from section 2.8.

notAKnotInterpolationVsStandard.m Implements standard MQ interpolation that is compared to the not-a-knot implementation in section 2.8.

phi.m Evaluates the MQ and integrated MQ (section 5.8) RBFs.

poissonAffine2.m Solves the Poisson problem (4.1) using the affine solver to evaluate the linear system involved in section 4.4.

poissonBoundaryCollocation.m* Solves a Poisson problem using boundary collocation (section 5.9).

poissonSVD.m Solves the Poisson problem (4.1) using the SVD to evaluate the linear system involved in section 4.3.

poissonTSVD.m Solves the Poisson problem (4.1) using the truncated SVD to evaluate the linear system involved in section 4.3.1.

poissonITSVD.m Solves the Poisson problem (4.1) using the improved truncated SVD to evaluate the linear system involved in section 4.3.2.

poissonDomainDecomp.m Solves the PDE boundary value problem (4.1) using the multiplicative Schwarz domain decomposition method.

poissonExampleMQ.m* Solves the Elliptic PDE (4.1) with Dirichlet boundary conditions over an range of the shape parameter. Plots error vs. shape and the condition number of the evaluation matrix vs. the shape parameter.

poissonExampleIMQ7.m An example that uses a integrated MQ (section 5.8) to solve a Poisson problem.

poissonGreedySingleShape.m Uses the greedy algorithm from section 4.7 to select a subset of centers, with the same shape parameter with each center, to use in the solution of a Poisson problem.

poissonLocalGlobalKappa.m Compares the local and global condition numbers of the evaluation matrix for a Poisson problem in section 2.6.

poissonNeumannMQ.m* An example of the MQ collocation method for the Elliptic PDE (4.1) with both Dirichlet and Neumann boundary conditions.

poissonExtendedPrecision.m Extended floating point precision of problem 3.10.

rk4.m "Classical" fourth order explicit Runge-Kutta method that is used to advance in time all time-dependent PDE examples.

setUpHeatComplexFdMode.m Setups up derivative matrix for the MQ finite difference mode calculation of the 2d diffusion example from section 5.6.

stationaryInterpolation.m Examines the accuracy of a stationary interpolation example vs. an error bound. Figure 2.4.

systemMatrixMQ.m* Returns the 1d system or interpolation matrix with entries given by equation (2.5).

systemMatrixMQ2d.m* Returns the 2d system or interpolation matrix with entries given by equation (2.5).

truncatedSvd.m* Implements the truncated Singular Value solver. Section 4.3.1.

vBurgersMQ.m* Burgers' equation example from section 3.2.4.

Appendix B

Additional Meshfree Method References

Books

Sladek, J; Sladek, V (2006): *Advances in Meshless Methods*, Crest Publishers, (2006)

Atluri, S (2004): *The meshless method (MLPG) for domain & bie discretizations*, Crest Publishers.

Journal articles:

Adibi, H; Es'haghi, J (2007): Numerical solution for biharmonic equation using multilevel radial basis functions and domain decomposition methods, *APPLIED MATHEMATICS AND COMPUTATION* Vol. 186 Issue: 1 pp. 246-255.

Ahmed, SG (2006): A collocation method using new combined radial basis functions of thin plate and multiquadraic types, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS* Vol. 30 Issue: 8 pp. 697-701.

Al-Gahtani, H.J.; Naffa'a, M (2009): RBF meshless method for large deflection of thin plates with immovable edges, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENT*S, Vol. 33 Issue: 2 pp. 176-183.

Alipanah, A; Dehghan, M (2007): Numerical solution of the nonlinear Fredholm integral equations by positive definite functions, *APPLIED MATHEMATICS AND COMPUTATION*, Vol. 190 Issue: 2 pp. 1754-1761.

Aminataei, A; Mazarei, MM (2008): Numerical solution of Poisson's equation using radial basis function networks on the polar coordinate, *COM-*

PUTERS & MATHEMATICS WITH APPLICATIONS, Vol. 56 Issue: 11 pp. 2887-2895.

Amaziane, B; Naji, A; Ouazar, D (2004): Radial basis function and genetic algorithms for parameter identification to some groundwater flow problems, *CMC-COMPUTERS MATERIALS & CONTINUA,* Vol. 1 Issue: 2 pp. 117-128.

Baxter, BJC (2006): Scaling radial basis functions via Euclidean distance matrices, *COMPUTERS & MATHEMATICS WITH APPLICATIONS*, Vol. 51 Issue: 8 pp. 1163-1170.

Bernal, F; Gutierrez, G; Kindelan, M (2009): Use of singularity capturing functions in the solution of problems with discontinuous boundary conditions, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 33 Issue: 2 pp. 200-208.

Bernal, F; Kindelan, M (2007): RBF meshless modeling of non-Newtonian Hele-Shaw flow, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 31 Issue: 10 pp. 863-874.

Bini, DA; De Rossi, A; Gabutti, BH (2008): On certain (block) Toeplitz matrices related to radial functions, *LINEAR ALGEBRA AND ITS APPLICATIONS*, Vol. 428 Issue: 2-3 pp. 508-519.

Bishop, NT; Gomez, R; Holvorcem, PR; Matzner, RA; Papadopoulos, P; Winicour, J (1997): Cauchy-characteristic evolution and waveforms, *JOURNAL OF COMPUTATIONAL PHYSICS*, Vol. 136 Issue: 1 pp. 140-167.

Bouhamidi, A; Jbilou, K (2008): Meshless thin plate spline methods for the modified Helmholtz equation, *COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING*, Vol. 197 Issue: 45-48 pp. 3733-3741.

Boztosun, I; Charafi, A (2002): An analysis of the linear advection-diffusion equation using mesh-free and mesh-dependent methods, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 26 Issue: 10 pp. 889-895.

Bozzini, M; Lenarduzzi, L; Rossini, M; Schaback, R (2004): Interpolation by basis functions of different scales and shapes, *CALCOLO*, Vol. 41 Issue: 2 pp. 77-87.

Carlson, RE; Foley, TA (1991): The parameter R^2 in multiquadric interpolation, *COMPUTERS & MATHEMATICS WITH APPLICATIONS*,

Vol. 21 Issue: 9 pp. 29-42.

Chandhini, G; Sanyasiraju, YVSS (2007): Local RBF-FD solutions for steady convection-diffusion problems, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING,* Vol. 72 Issue: 3 pp. 352-378 .

Chantasiriwan, S (2007): Solutions to harmonic and biharmonic problems with discontinuous boundary conditions by collocation methods using multiquadrics as basis functions, *INTERNATIONAL COMMUNICATIONS IN HEAT AND MASS TRANSFER*, Vol. 34 Issue: 3 pp. 313-320.

Chantasiriwan, S (2007): Multiquadric collocation method for time-dependent heat conduction problems with temperature-dependent thermal properties, *JOURNAL OF HEAT TRANSFER-TRANSACTIONS OF THE ASME*, Vol. 129 Issue: 2 pp. 109-113.

Chantasiriwan, S (2007): Collocation methods based on radial basis functions for solving stochastic Poisson problems, *COMMUNICATIONS IN NUMERICAL METHODS IN ENGINEERING*, Vol. 23 Issue: 3 pp. 169-178.

Chantasiriwan, S (2006): An alternative approach for numerical solutions of the Navier-Stokes equations, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, Vol. 69 Issue: 7 pp. 1331-1344.

Chantasiriwan, S (2006): Methods of fundamental solutions for time-dependent heat conduction problems , *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, Vol. 66 Issue: 1 pp. 147-165.

Chantasiriwan, S (2006): Performance of multiquadric collocation method in solving lid-driven cavity flow problem with low Reynolds number, *CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES*, Vol. 15 Issue: 3 pp. 137-146.

Chantasiriwan, S (2006): Error and variance of solution to the stochastic heat conduction problem by multiquadric collocation method, *INTERNATIONAL COMMUNICATIONS IN HEAT AND MASS TRANSFER*, Vol. 33 Issue: 3 pp. 342-349.

Chantasiriwan, S (2005): Solutions of partial differential equations with random Dirichlet boundary conditions by multiquadric collocation method,

ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS, Vol. 29 Issue: 12 pp. 1124-1129.

Chantasiriwan, S (2004): Investigation of the use of radial basis functions in local collocation method for solving diffusion problems, *INTERNATIONAL COMMUNICATIONS IN HEAT AND MASS TRANSFER*, Vol. 31 Issue: 8 pp. 1095-1104.

Chantasiriwan, S (2004): Cartesian grid methods using radial basis functions for solving Poisson, Helmholtz, and diffusion-convection equations, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 28 Issue: 12 pp. 1417-1425.

Chen, CS; Ganesh, M; Golberg, MA; Cheng, AHD (2002): Multilevel compact radial functions based computational schemes for some elliptic problems, *COMPUTERS & MATHEMATICS WITH APPLICATIONS*. Vol. 43 Issue: 3-5 pp. 359-378.

Chen, CS; Kuhn, G; Li, J; Mishuris ,G (2003): Radial basis functions for solving near singular Poisson problems, *COMMUNICATIONS IN NUMERICAL METHODS IN ENGINEERING*, Vol. 19 Issue: 5 pp. 333-347**.**

Chen, CW; Fan, CM; Young, DL, Murugesan K; Tsai ,CC (2005): Eigenanalysis for membranes with stringers using the methods of fundamental solutions and domain decomposition , *CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES,* Vol. 8 Issue: 1 pp. 29-44.

Chen, HQ; Shu, C (2005): An efficient implicit mesh-free method to solve two-dimensional compressible Euler equations, *INTERNATIONAL JOURNAL OF MODERN PHYSICS C,* Vol. 16 Issue: 3 pp. 439-454.

Chen, JS; Hu, W; Hu, HY (2008): Reproducing kernel enhanced local radial basis collocation method, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING,* Vol. 75 Issue: 5 pp. 600-627.

Chen, RH; Wu, ZM (2007): Solving partial differential equation by using multiquadric quasi-interpolation, *APPLIED MATHEMATICS AND COMPUTATION,* Vol. 186 Issue: 2 pp. 1502-1510.

Chen, RH; Wu, ZM (2006): Solving hyperbolic conservation laws using multiquadric quasi-interpolation, *NUMERICAL METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS*, Vol. 22 Issue: 4 pp. 776-796.

Chen, RH; Wu, ZM (2006): Applying multiquadric quasi-interpolation to solve Burgers' equation . *APPLIED MATHEMATICS AND COMPUTATION*, Vol. 172 Issue: 1 pp. 472-484 .

Chen, W; Hon, YC (2003): Numerical investigation on convergence of boundary knot method in the analysis of homogeneous Helmholtz, modified Helmholtz, and convection-diffusion problems, *COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING*, Vol. 192 Issue: 15 pp. 1859-1875.

Chen, W; Tanaka, M (2002): A meshless, integration-free, and boundary-only RBF technique, *COMPUTERS & MATHEMATICS WITH APPLICATIONS*, Vol. 43 Issue: 3-5 pp. 379-391.

Cheng, AHD; Golberg, MA; Kansa, EJ; Zammito G (2003): Exponential convergence and H-c multiquadric collocation method for partial differential equations, *NUMERICAL METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS*, Vol. 19 Issue: 5 pp. 571-594.

Cheng, AHD; Cabral, JJSP (2005): Direct solution of ill-posed boundary value problems by radial basis function collocation method, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, Vol. 64 Issue: 1 pp. 45-64.

Chinchapatnam, PP; Djidjeli. K; Nair PB (2007): Radial basis function meshless method for the steady incompressible Navier-Stokes equations, *INTERNATIONAL JOURNAL OF COMPUTER MATHEMATICS*, Vol. 84 Issue: 10 pp. 1509-1526.

Chinchapatnam, PP; Djidjeli ,K; Nair, PB (2007): Domain decomposition for time-dependent problems using radial based meshless methods, *NUMERICAL METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS*, Vol. 23 Issue: 1 pp. 38-59.

Chinchapatnam, PP; Djidjeli, K; Nair, PB (2006): Unsymmetric and symmetric meshless schemes for the unsteady convection-diffusion equation, *COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING*, Vol. 195 Issue: 19-22 pp. 2432-2453.

Cho, HA; Golberg, MA; Muleshkov, AS; Li, X (2004): Trefftz methods for time dependent partial differential equations, *CMC-COMPUTERS MATERIALS & CONTINUA*, Vol. 1 Issue: 1 pp. 1-37.

CooK, GB; Choptuik, MW; Dubal, MR; Klasky, YS; Matzner, RA; Oliveira, SR (1993): 3-Dimensional intial data for the collision of 2

black holes, *PHYSICAL REVIEW D*, Vol. 47 Issue: 4 pp. 1471-1490.

Dag, I; Dereli, Y (2008): Numerical solutions of KdV equation using radial basis functions, *APPLIED MATHEMATICAL MODELLING*, Vol. 32 Issue: 4 pp. 535-546.

Davi, G; Milazzo, A (2003): A meshfree method for transverse vibrations of anisotropic plates, *INTERNATIONAL JOURNAL OF SOLIDS AND STRUCTURES*, Vol. 40 Issue: 20 pp. 5229-5249.

Dehghan, M; Shokri, A (2008): A numerical method for solving the hyperbolic telegraph equation, *NUMERICAL METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS*, Vol. 24 Issue: 4 pp. 1080-1093.

Dehghan M; Shokri A (2008): A numerical method for solution of the two-dimensional sine-Gordon equation using the radial basis functions, *MATHEMATICS AND COMPUTERS IN SIMULATION*, Vol. 79 Issue: 3 pp. 700-715 .

Dehghan, M; Tatari, M (2008): Use of radial basis functions for solving the second-order parabolic equation with nonlocal boundary conditions, *NUMERICAL METHODS FOR PARTIAL DIFFERENTIAL EQUATION*S, Vol. 24 Issue: 3 pp. 924-938.

Dehghan, M; Shokri, A (2008): A numerical method for one-dimensional nonlinear Sine-Gordon equation using collocation and radial basis functions, *NUMERICAL METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS*, Vol. 24 Issue: 2 pp. 687-698 .

Dehghan, M; Shokri, A (2007): A numerical method for KdV equation using collocation and radial basis functions, *NONLINEAR DYNAMICS*, Vol. 50 Issue: 1-2 pp. 111-120.

Dehghan, M; Tatari, M (2007): The radial basis functions method for identifying an unknown parameter in a parabolic equation with overspecified data, *NUMERICAL METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS*, Vol. 23 Issue: 5 pp. 984-997.

Dehghan, M; Shokri, A (2007): A numerical method for two-dimensional Schrodinger equation using collocation and radial basis functions, *COMPUTERS & MATHEMATICS WITH APPLICATIONS*, Vol. 54 Issue: 1 pp. 136-146.

Dehghan, M; Tatari, M (2007): The radial basis functions method for identifying an unknown parameter in a parabolic equation with overspecified

data, *NUMERICAL METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS,* Vol. 23 Issue: 5 pp. 984-997.

Dehghan, M; Tatari, M (2006): Determination of a control parameter in a one-dimensional parabolic equation using the method of radial basis functions, *MATHEMATICAL AND COMPUTER MODELLING,* Vol. 44 Issue: 11-12 pp. 1160-1168.

Dehghan, M; Shokri, A (2009): A Meshless Method for Numerical Solution of a Linear Hyperbolic Equation with Variable Coefficients in Two Space Dimensions, *NUMERICAL METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS*, Vol. 25 Issue: 2 pp. 494-506 .

Demirkaya, G; Soh, CW; Ilegbusi, OJ (2008): Direct solution of Navier-Stokes equations by radial basis functions, *APPLIED MATHEMATICAL MODELLING*, Vol. 32 Issue: 9 pp. 1848-1858.

Ding, H; Shu, C; Tang, DB (2005): Error estimates of local multiquadric-based differential quadrature (LMQDQ) method through numerical experiments, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, Vol. 63 Issue: 11 pp. 1513-1529.

Dinis, LMJS; Jorge, RMN; Belinha, J (2007): Analysis of 3D solids using the natural neighbour radial point interpolation method, *COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING*, Vol. 196 Issue: 13-16 pp. 2009-2028.

Divo, E; Kassab, AJ *(2007): An efficient localized radial basis function meshless method for fluid flow and conjugate heat transfer, JOURNAL OF HEAT TRANSFER-TRANSACTIONS OF THE ASME*, Vol. 129 Issue: 2 pp. 124-136.

Divo, E; Kassab, A (2006): Iterative domain decomposition meshless method modeling of incompressible viscous flows and conjugate heat transfer ,*ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*,Vol. 30 Issue: 6 pp. 465-478 .

Divo, E; Kassab, AJ (2005): A meshless method for conjugate heat transfer problems, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 29 Issue: 2 pp. 136-149.

Duan, Y (2008): A note on the meshless method using radial basis functions, *COMPUTERS & MATHEMATICS WITH APPLICATIONS,* Vol. 55 Issue: 1 ;pp 66-75.

Duan, Y; Tang, PF; Huang, TZ; Lai, SJ (2009):, Coupling projection domain decomposition method and Kansa's method in electrostatic problems, *COMPUTER PHYSICS COMMUNICATIONS*, Vol. 180 Issue: 2, pp. 209-214.

Dubal, MR (1992): Construction of 3-dimensional black-hole initial data via multiquadrics, *PHYSICAL REVIEW D*, Vol. 45 Issue: 4 pp. 1178-1187.

El Zahab, Z; Divo. E; Kassab A (2009): A meshless CFD approach for evolutionary shape optimization of bypass grafts anastomoses, *INVERSE PROBLEMS IN SCIENCE AND ENGINEERING*, Vol. 17 Issue: 3 pp. 411-435.

Emdadi, A; Kansa, EJ; Libre NA; Rahimian M; Shekarchi M (2008): Stable PDE solution methods for large multiquadric shape parameters, *CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES,* Vol. 25 Issue: 1 pp. 23-41.

Fasshauer, GE; Zhang, JG (2007): On choosing "optimal" shape parameters for RBF approximation *NUMERICAL ALGORITHMS*, Vol. 45 Issue: 1-4 pp. 345-368.

Fasshauer, GE (2002): Newton iteration with multiquadrics for the solution of nonlinear PDEs, *COMPUTERS & MATHEMATICS WITH APPLICATIONS*, Vol. 43 Issue: 3-5 pp. 423-438.

Fasshauer, GE (1999): Solving differential equations with radial basis functions: multilevel methods and smoothing , *ADVANCES IN COMPUTATIONAL MATHEMATICS*, Vol. 11 Issue: 2-3 pp. 139-159.

Fedoseyev, AL; Friedman, MJ; Kansa, EJ (2002): Improved multiquadric method for elliptic partial differential equations via PDE collocation on the boundary, *COMPUTERS & MATHEMATICS WITH APPLICATIONS*, Vol. 43 Issue: 3-5 pp. 439-455.

Ferreira, AJM; Fasshauer, GE; Batra, RC; Rodrigues, JD (2008): Static deformations and vibration analysis of composite and sandwich plates using a layerwise theory and RBF-PS discretizations with optimal shape parameter, *COMPOSITE STRUCTURES*, Vol. 86 Issue: 4 pp. 328-343.

Ferreira, AJM; Roque, CMC; Jorge, RMN; Fasshauer, GE; Batra, RC (2007): Analysis of functionally graded plates by a robust meshless method, *MECHANICS OF ADVANCED MATERIALS AND STRUCTURES*, Vol. 14 Issue: 8 pp. 577-587.

Ferreira, AJM; Roque, CMC; Jorge, RMN (2007): Natural frequencies of FSDT cross-ply composite shells by multiquadrics, *COMPOSITE STRUCTURES*, Vol. 77 Issue: 3 pp. 296-305.

Ferreira, AJM; Roque ,CMC; Jorge, RMN (2006): Static and free vibration analysis of composite shells by radial basis functions, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 30 Issue: 9 pp. 719-733.

Ferreira, AJM; Batra, RC; Roque, CMC; Qian, LF; Jorge, RMN (2006): Natural frequencies of functionally graded plates by a meshless method , *COMPOSITE STRUCTURES*, Vol. 75 Issue: 1-4 Special Issue: pp. 593-600.

Ferreira, AJM; Roque, CMC; Jorge, RMN (2006): Modelling cross-ply laminated elastic shells by a higher-order theory and multiquadrics, *COMPUTERS & STRUCTURES*, Vol. 84 Issue: 19-20 pp. 1288-1299.

Ferreira, AJM (2005): Analysis of composite plates using a layerwise theory and multiquadrics discretization , *MECHANICS OF ADVANCED MATERIALS AND STRUCTURES*, Vol. 12 Issue: 2 pp. 99-112 .

Ferreira, AJM; Roque, CMC; Jorge ,RMN; Kansa, EJ (2005): Static deformations and vibration analysis of composite and sandwich plates using a layerwise theory and multiquadrics discretizations, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 29 Issue: 12 pp. 1104-1114.

Ferreira, AJM; Batra, RC (2005): Natural frequencies of orthotropic, monoctinic and hexagonal plates by a meshless method , *JOURNAL OF SOUND AND VIBRATION,* Vol. 285 Issue: 3 pp. 734-742.

Ferreira, AJM; Roque, CMC; Jorge, RMN (2005): Analysis of composite plates by trigonometric shear deformation theory and multiquadrics, *COMPUTERS & STRUCTURES*, Vol. 83 Issue: 27 pp. 2225-2237.

Ferreira, AJM; Martins PALS, Roque CMC (2005): Solving time-dependent engineering problems with multiquadrics, *JOURNAL OF SOUND AND VIBRATION,* Vol. 280 Issue: 3-5 pp. 595-610.

Ferreira, AJM; Roque, CMC; Martins, PALS (2004): Radial basis functions and higher-order shear deformation theories in the analysis of laminated composite beams and plates, *COMPOSITE STRUCTURES*, Vol. 66 Issue: 1-4 pp. 287-293.

Ferreira, AJM (2004): Polyharmonic (thin-plate) splines in the analysis of composite plates,*INTERNATIONAL JOURNAL OF MECHANICAL SCIENCES*, Vol. 46 Issue: 10 pp. 1549-1569.

Ferreira, AJM; Roque, CMC; Martins, PALS (2003): Analysis of composite plates using higher-order shear deformation theory and a finite point formulation based on the multiquadric radial basis function method, *COMPOSITES PART B-ENGINEERING*, Vol. 34 Issue: 7 pp. 627-636.

Ferreira, AJM (2003): Thick composite beam analysis using a global meshless approximation based on radial basis functions, *MECHANICS OF ADVANCED MATERIALS AND STRUCTURES*, Vol. 10 Issue: 3 pp. 271-284.

Ferreira, AJM (2003): A formulation of the multiquadric radial basis function method for the analysis of laminated composite plates, *COMPOSITE STRUCTURES*, Vol. 59 Issue: 3 pp. 385-392.

Firoozjaee, AR; Afshar, MH (2009): Discrete least squares meshless method with sampling points for the solution of elliptic partial differential equations, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 33 Issue: 1 pp. 83-92.

Flyer, N; Wright, GB (2009): A radial basis function method for the shallow water equations on a sphere, *PROCEEDINGS OF THE ROYAL SOCIETY A-MATHEMATICAL PHYSICAL AND ENGINEERING SCIENCES,* Vol. 465 Issue 2106 , pp. 1949-1976.

Flyer, N; Wright, GB (2007): Transport schemes on a sphere using radial basis functions, *JOURNAL OF COMPUTATIONAL PHYSICS*, Vol. 226 Issue: 1 pp. 1059-1084.

Fornberg, B; Piret, C (2008): On choosing a radial basis function and a shape parameter when solving a convective PDE on a sphere, *JOURNAL OF COMPUTATIONAL PHYSICS*, Vol. 227 Issue: 5 pp. 2758-2780.

Fornberg, B; Piret, C (2008): A stable algorithm for flat radial basis functions on a sphere, *SIAM JOURNAL ON SCIENTIFIC COMPUTING*, Vol. 30 Issue: 1 pp. 60-80.

Fornberg, B; Zuev, J (2007): The Runge phenomenon and spatially variable shape parameters in RBF interpolation, *COMPUTERS & MATHEMATICS WITH APPLICATIONS*, Vol. 54 Issue: 3 pp. 379-398.

Fornberg, B; Driscoll, TA; Wright, G; Charles, R (2002): Observations on the behavior of radial basis function approximations near bound-

aries, *COMPUTERS & MATHEMATICS WITH APPLICATIONS*, Vol. 43 Issue: 3-5 pp. 473-490.

Fuselier, EJ (2008): Sobolev-type approximation rates for divergence-free and curl-free RBF interpolants, *MATHEMATICS OF COMPUTATION*, Vol. 77 Issue: 263 pp. 1407-1423.

Galperin, EA; Zheng, Q (1993): Solution and control of PDE via global optimization methods, *COMPUTERS & MATHEMATICS WITH APPLICATIONS*, Vol. 25 Issue: 10-11 pp. 103-118.

Galperin, EA; Zheng, Q (1993): Application of global optimization to implicit solutions of partial differential equations, *COMPUTERS & MATHEMATICS WITH APPLICATIONS*, Vol. 25 Issue: 10-11 pp. 119-124.

Gaspar, C (2009): Multi-level meshless methods based on direct multi-elliptic interpolation, *JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS*, Vol. 226 Issue: 2 Special Issue: Sp. Iss. SI pp. 259-267.

Gelas, A; Bernard, O; Friboulet, D; Prost, R (2007): Compactly supported radial basis functions based collocation method for level-set evolution in image segmentation, *IEEE TRANSACTIONS ON IMAGE PROCESSING*, Vol. 16 Issue: 7 pp. 1873-1887.

Ghorbani, M; Soheili, AR (2006): Corrected fundamental solution for numerical solution of elliptic PDEs, *APPLIED MATHEMATICS AND COMPUTATION,* Vol. 181 Issue: 1 pp. 175-184.

Giesl, P; Wendland, H (2007): Meshless collocation: Error estimates with application to dynamical systems, *SIAM JOURNAL ON NUMERICAL ANALYSIS*, Vol. 45 Issue: 4 pp. 1723-1741.

Gilhooley, DF; Xiao, JR; Batra, RC; McCarthy, MA; Gillespie, JW (2008): Two-dimensional stress analysis of functionally graded solids using the MLPG method with radial basis functions, *COMPUTATIONAL MATERIALS SCIENCE*, Vol. 41 Issue: 4 pp. 467-481.

Girosi, F; Jones, M; Poggio,T (1995): Regularization theory and neural networks architectures, *NEURAL COMPUTATION*, Vol. 7 Issue: 2 pp. 219-269.

Girosi, F (1992): Some extensions of radial basis functions and their applications in artificial-intelligence, *COMPUTERS & MATHEMATICS WITH APPLICATIONS*, Vol. 24, Issue: 12 pp. 61-80.

Godinho, L; Tadeu, A; Mendes, PA (2007): Wave propagation around thin structures using the MFS,. *CMC-COMPUTERS MATERIALS*

& CONTINUA, Vol. 5 Issue: 2 pp. 117-127.

Godinho, L; Mendes, PA; Tadeu, A; Cadena-Isaza, A; Smerzini, C; Sanchez-Sesma, FJ; Madec, R; Komatitsch, D (2009): Numerical Simulation of Ground Rotations along 2D Topographical Profiles under the Incidence of Elastic Plane Waves, *BULLETIN OF THE SEISMOLOGICAL SOCIETY OF AMERICA*, Vol. 99 Issue: 2B pp. 1147-1161.

Golberg, MA; Chen, CS; Bowman, H (1999): Some recent results and proposals for the use of radial basis functions in the BEM, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 23 Issue: 4 pp. 285-296.

Gonzalez, M; Goldschmit, MB (2006): Inverse geometry heat transfer problem based on a radial basis functions geometry representation, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, Vol. 65 Issue: 8 pp. 1243-1268.

Guimaraes, FG; Saldanha, RR; Mesquita, RC; Lowther, DA; Ramirez, JA (2007): A meshless method for electromagnetic field computation based on the multiquadric technique, *IEEE TRANSACTIONS ON MAGNETICS*, Vol. 43 Issue: 4 Special Issue: Sp. Iss. SI pp. 1281-1284.

Gutierrez, G; Florez, W (2008): Comparison between global, classical domain decomposition and local, single and double collocation methods based on RBF interpolation for solving convection-diffusion equation, *INTERNATIONAL JOURNAL OF MODERN PHYSICS C,* Vol. 19 Issue: 11 pp. 1737-1751 .

Haq, S; Siraj-ul-Islam; Ali, A (2009): A Numerical Meshfree Technique for the Solution of the MEW Equation, *CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES,* Vol. 38 Issue: 1 pp. 1-23.

Hart, EE; Cox, SJ; Djidjeli, K; Kubytskyi, VO (2009): Solving an eigenvalue problem with a periodic domain using radial basis functions, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 33 Issue: 2 pp. 258-262.

Hon, YC; Yang, ZH (2009): Meshless collocation method by Delta-shaped basis functions for default barrier model ,*ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 33 Issue: 7 pp. 951-958.

Hon, YC; Ling, L; Liew, KM (2005): Numerical analysis of parameters in a laminated beam model by radial basis functions , *CMC-COMPUTERS MATERIALS & CONTINUA*, Vol. 2 Issue: 1 pp. 39-49.

Hon, YC; Chen, W (2003): Boundary knot method for 2D and 3D Helmholtz and convection-diffusion problems under complicated geometry, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, Vol. 56 Issue: 13 pp. 1931-1948.

Hon, YC; Schaback, R; Zhou, X (2003): An adaptive greedy algorithm for solving large RBF collocation problems, *NUMERICAL ALGORITHMS,* Vol. 32 Issue: 1 pp. 13-25.

Hon, YC; Lu, MW; Xue, WM; Zhou, X (2002): Numerical algorithm for triphasic model of charged and hydrated soft tissues, *COMPUTATIONAL MECHANICS*, Vol. 29 Issue: 1 pp. 1-15.

Hon, YC (2002): A quasi-radial basis functions method for American options pricing, *COMPUTERS & MATHEMATICS WITH APPLICATIONS*, Vol. 43 Issue: 3-5 pp. 513-524.

Hon, YC; Wu, ZM (2000): A numerical computation for inverse boundary determination problem, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 24 Issue: 7-8 pp. 599-606.

Hon, YC; Wu, ZM (2000): A quasi-interpolation method for solving stiff ordinary differential equations, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, Vol. 48 Issue: 8 pp. 1187-1197.

Hon, YC; Cheung, KF; Mao, XZ; Kansa, EJ (1999): Multiquadric solution for shallow water equations, *JOURNAL OF HYDRAULIC ENGINEERING-ASCE,* Vol. 125 Issue: 5 pp. 524-533.

Hon, YC; Mao, XZ (1998): An efficient numerical scheme for Burgers' equation, *APPLIED MATHEMATICS AND COMPUTATION*, Vol. 95 Issue: 1 pp. 37-50.

Hon ,YC; Lu, MW; Xue, WM; Zhu,YM (1997): Multiquadric method for the numerical solution of a biphasic mixture model, *APPLIED MATHEMATICS AND COMPUTATION*, Vol. 88 Issue: 2-3 pp. 153-175.

Hu, HY; Chen, JS; Hu, W (2007): Weighted radial basis collocation method for boundary value problems, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, Vol. 69 Issue: 13 pp. 2736-2757.

Hu, HY; Chen, JS (2008): Radial basis collocation method and quasi-Newton iteration for nonlinear elliptic problems, *NUMERICAL METHODS*

FOR PARTIAL DIFFERENTIAL EQUATIONS, Vol. 24, Issue: 3 pp. 991-1017.

Hu, SP; Young, DL; Fan, CM (2008): FDMFS for diffusion equation with unsteady forcing function, *CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES*, Vol. 24, Issue: 1 pp. 1-20.

Hu, XG; Ho, TS; Rabitz, H; Askar, A (2000): Solution of the quantum fluid dynamical equations with radial basis function interpolation, *PHYSICAL REVIEW E-PART B*, Vol. 61 Issue: 5 pp. 5967-5976.

Hu, XG; Ho, TS; Rabitz, H; Askar, A (2002): Multivariate radial basis interpolation for solving quantum fluid dynamical equations, *COMPUTERS & MATHEMATICS WITH APPLICATIONS*, Vol. 43 Issue: 3-5 pp. 525-537.

Jin, BT; Chen, W (2006): Boundary knot method based on geodesic distance for anisotropic problems *, JOURNAL OF COMPUTATIONAL PHYSICS*, Vol. 215 Issue: 2 pp. 614-629.

Jin, BT; Zheng, Y (2005): Boundary knot method for some inverse problems associated with the Helmholtz equation, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, Vol. 62 Issue: 12 pp. 1636-1651.

Jin, BT; Zheng , Y (2006): A meshless method for some inverse problems associated with the Helmholtz equation, *COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING*, Vol. 195 Issue: 19-22 pp. 2270-2288 .

Jin, BT; Zheng, Y(2005): Boundary knot method for the Cauchy problem associated with the inhomogeneous Helmholtz equation, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 29 Issue: 10 pp. 925-935.

Jin, BT; Zheng, Y (2005): Boundary knot method for some inverse problems associated with the Helmholtz equation, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, Vol. 62 Issue: 12 pp. 1636-1651.

Jung , JH (2007): A note on the Gibbs phenomenon with multiquadric radial basis functions. *APPLIED NUMERICAL MATHEMATICS*, Vol. 57 Issue: 2 pp. 213-229.

Jung, JH, Durante, VR (2009): An iterative adaptive multiquadric radial basis function method for the detection of local jump discontinuities,

APPLIED NUMERICAL MATHEMATICS, Vol. 59 , pp: 1449-1466 .

Karageorghis, A; Chen, CS; Smyrlis, YS (2007): A matrix decomposition RBF algorithm: Approximation of functions and their derivatives, *APPLIED NUMERICAL MATHEMATICS*, Vol. 57 Issue: 3 pp. 304-319.

Katsikadelis, JT (2009): The meshless analog equation method: I. Solution of elliptic partial differential equations, *ARCHIVE OF APPLIED MECHANICS,* Vol. 79 Issue: 6-7 pp. 557-578.

Katsikadelis, JT (2008): The 2D elastostatic problem in inhomogeneous anisotropic bodies by the meshless analog equation method (MAEM), *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 32 Issue: 12 Special Issue: Sp. Iss. SI pp. 997-1005.

Katsikadelis, JT (2008): A generalized Ritz method for partial differential equations in domains of arbitrary geometry using global shape functions, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 32 Issue: 5 pp. 353-367.

Katsikadelis, JT; Tsiatas, GC (2003): Nonlinear dynamic analysis of heterogeneous orthotropic membranes by the analog equation method, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 27 Issue: 2 pp. 115-124.

Kee, BBT; Liu, GR; Zhang, GY; Lu, C (2008): A residual based error estimator using radial basis functions, *FINITE ELEMENTS IN ANALYSIS AND DESIGN*, Vol. 44 Issue: 9-10 pp. 631-645.

Kee, BBT; Liu, GR; Lu, C (2007): A regularized least-squares radial point collocation method (RLS-RPCM) for adaptive analysis, *COMPUTATIONAL MECHANICS*, Vol. 40 Issue: 5 pp. 837-853.

Khattak, AJ; Tirmizi, SIA; Siraj-ul-Islam (2009): Application of meshfree collocation method to a class of nonlinear partial differential equations, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 33 Issue: 5 pp. 661-667.

Khattak, AJ; Siraj-ul-Islam (2008): A comparative study of numerical solutions of a class of KdV equation, *APPLIED MATHEMATICS AND COMPUTATION*, Vol. 199 Issue: 2 pp. 425-434.

Khattak, AJ (2009): A computational meshless method for the generalized Burger's-Huxley equation, *APPLIED MATHEMATICAL MODELLING*, Vol. 33, pp. 3718-3729.

Kolodziej, JA; Mierzwiczak, M (2008): The optimal shape parameter of multiquadric collocation method for solution of nonlinear steady-state heat conduction in multilayered plate , *JOURNAL OF MECHANICS OF MATERIALS AND STRUCTURES*, Vol. 3 Issue: 6 pp. 1077-1086.

Kosec, G; Sarler, B (2008): Local RBF Collocation Method for Darcy flow, *CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES,* Vol. 25 Issue: 3 pp. 197-207.

Kosec, G; Sarler, B (2008): Solution of thermo-fluid problems by collocation with local pressure correction, *INTERNATIONAL JOURNAL OF NUMERICAL METHODS FOR HEAT & FLUID FLOW*, Vol. 18 Issue: 7-8 pp. 868-882.

Kovacevic, I; Poredos, A; Sarler, B (2003): Solving the Stefan problem with the radial basis function collocation method, **NUMERICAL HEAT TRANSFER PART B-FUNDAMENTALS**, Vol. 44 Issue: 6 pp. 575-599.

Kozulic, V; Gotovac, H; Gotovac, B (2007): An adaptive multiresolution method for solving PDE's, *CMC-COMPUTERS MATERIALS & CONTINUA*, Vol. 6 Issue: 2 pp. 51-70.

Kumar, S; Misra, RK (2006): Analysis of banana fibers reinforced low-density polyethylene/poly(epsilon-caprolactone) composites, *SOFT MATERIALS,* Vol. 4 Issue: 1 pp. 1-13.

Lai, SJ; Wang, BZ; Duan, Y (2008): Meshless radial basis function method for transient electromagnetic computations, *IEEE TRANSACTIONS ON MAGNETICS,* Vol. 44 Issue: 10 pp. 2288-2295.

La Rocca, A; Rosales, AH; Power, H (2005): Radial basis function Hermite collocation approach for the solution of time dependent convection-diffusion problem, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 29 Issue: 4 pp. 359-370 .

La Rocca, A; Power, H (2005): Free mesh radial basis function collocation approach for the numerical solution of system of multi-ion electrolytes, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, Vol. 64 Issue: 13 pp. 1699-1734.

La Rocca, A; Power, H (2008): A double boundary collocation Hermitian approach for the solution of steady state convection-diffusion problems, *COMPUTERS & MATHEMATICS WITH APPLICATIONS*, Vol. 55 Issue: 9 pp. 1950-1960.

La Rocca, A; Rosales, AH; Power, H (2005): Radial basis function Hermite collocation approach for the solution of time dependent convection-diffusion problems. *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 29 Issue: 4 pp. 359-370.

La Rocca, A; Power, H (2006): A Hermite radial basis function collocation approach for the numerical simulation of crystallization processes in a channel, *COMMUNICATIONS IN NUMERICAL METHODS IN ENGINEERING*, Vol. 22 Issue: 2 pp. 119-135.

Le, P; Mai-Duy, N; Tran-Cong, T; Baker, G (2007): A numerical study of strain localization in elasto-thermo-viscoplastic materials using radial basis function networks, *CMC-COMPUTERS MATERIALS & CONTINUA*, Vol. 5 Issue: 2 pp. 129-150.

Le, P; Mai-Duy, N; Tran-Cong, T; Baker, G (2008): A meshless modeling of dynamic strain localization in quasi-brittle materials using radial basis function networks, *CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES*, Vol. 25 Issue: 1 pp. 43-67.

Lee, CF; Ling, LV; Schaback, R (2009): On convergent numerical algorithms for unsymmetric collocation, *ADVANCES IN COMPUTATIONAL MATHEMATICS*, Vol. 30 Issue: 4 pp. 339-354.

Lee, CK; Liu, X; Fan, SC (2003): Local multiquadric approximation for solving boundary value problems, *COMPUTATIONAL MECHANICS*, Vol. 30 Issue: 5-6 pp. 396-409.

Li, CS; Xiong, SW; Rodrigues, JMC; Martins, PAF (2005): Simulation of bulk metal forming by means of finite integral mesh free methods , *PROCEEDINGS OF THE INSTITUTION OF MECHANICAL ENGINEERS PART C-JOURNAL OF MECHANICAL ENGINEERING SCIENCE*, Vol. 219 Issue: 9 pp. 949-963.

Li, J; Chen, Y; Pepper, D (2003): Radial basis function method for 1-D and 2-D groundwater contaminant transport modeling, *COMPUTATIONAL MECHANICS*, Vol. 32 Issue: 1-2 pp. 10-15.

Li, JC (2001): Mathematical justification for RBF-MFS, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 25 Issue: 10 pp. 897-901.

Li, JC (2004): A radial basis meshless method for solving inverse boundary value problems, *COMMUNICATIONS IN NUMERICAL METHODS IN ENGINEERING*, Vol. 20 Issue: 1 pp. 51-61.

Li, JC; Cheng, AHD; Chen, CS (2003): A comparison of efficiency and error convergence of multiquadric collocation method and finite element method, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 27 Issue: 3 pp. 251-257.

Li, JC; Chen, CS (2003): Some observations on unsymmetric radial basis function collocation methods for convection-diffusion problems, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING,* Vol. 57 Issue: 8 pp. 1085-1094.

Li, JC (2005): Application of radial basis meshless methods to direct and inverse biharmonic boundary value problems , *COMMUNICATIONS IN NUMERICAL METHODS IN ENGINEERING*, Vol. 21 Issue: 4 pp. 169-182.

Li, JC; Hon, YC; Chen, CS (2002): Numerical comparisons of two meshless methods using radial basis functions, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 26 Issue: 3 pp. 205-225.

Li, JC; Hon, YC (2004): Domain decomposition for radial basis meshless methods, *NUMERICAL METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS* , Vol. 20 Issue: 3 pp. 450-462.

Li, JY; Luo, SW; Qi, YJ; Huang ,YP (2003): Numerical solution of elliptic partial differential equation using radial basis function neural networks, *NEURAL NETWORKS,* Vol. 16 Issue: 5-6 pp. 729-734.

Libre, NA, Emdadi, A; Kansa. EJ, Shekarchi, M; Rahimian, M (2008) A Fast Adaptive Wavelet scheme in RBF Collocation for nearly singular potential PDE, *CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES*, Vol. 38 Issue: 3 pp. 263-284.

Liew, KM; Chen, XL (2004): Mesh-free radial point interpolation method for the buckling analysis of Mindlin plates subjected to in-plane point loads, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING* , Vol. 60 Issue: 11 pp. 1861-1877

Ling, L; Schaback, R (2008): Stable and convergent unsymmetric meshless collocation methods, *SIAM JOURNAL ON NUMERICAL ANALYSIS*, Vol. 46 Issue: 3 pp. 1097-1115.

Ling, L; Hon, YC (2005): Improved numerical solver for Kansa's method based on affine space decomposition, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 29 Issue: 12 pp. 1077-1085.

Ling, L (2004): A univariate quasi-multiquadric interpolation with better smoothness, *COMPUTERS & MATHEMATICS WITH APPLICATIONS*, Vol. 48 Issue: 5-6 pp. 897-912.

Ling, LV; Hon, YC; Yamamoto, M (2005): Inverse source identification for Poisson equation, *INVERSE PROBLEMS IN SCIENCE AND ENGINEERING*, Vol. 13 Issue: 4 pp. 433-447.

Ling, L; Trummer, MR (2004): Multiquadric collocation method with integral formulation for boundary layer problems, *COMPUTERS & MATHEMATICS WITH APPLICATIONS*, Vol. 48 Issue: 5-6 pp. 927-941.

Ling, L; Trummer ,MR (2006): Adaptive multiquadric collocation for boundary layer problems, *JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS*, Vol. 188 Issue: 2 pp. 265-282.

Liu, L; Chua, LP; Ghista, DN (2006): Conforming radial point interpolation method for spatial shell structures on the stress-resultant shell theory, *ARCHIVE OF APPLIED MECHANICS*, Vol. 75 Issue: 4-5 pp. 248-267.

Liu, L; Chua, LP; Ghista, DN (2007): Mesh-free radial basis function method for static, free vibration and buckling analysis of shear deformable composite laminates, *COMPOSITE STRUCTURES,* Vol. 78 Issue: 1 pp. 58-69.

Liu, X; Liu, GR; Tai, K; Lam, KY (2005): Radial point interpolation collocation method (RPICM) for partial differential equations, *COMPUTERS & MATHEMATICS WITH APPLICATIONS,* Vol. 50 Issue: 8-9 pp. 1425-1442.

Livne, OE; Wright, GB (2006): Fast multilevel evaluation of smooth radial basis function expansions, *ELECTRONIC TRANSACTIONS ON NUMERICAL ANALYSIS*, Vol. 23 pp. 263-287.

Long, SY; Xiong, YB (2002): Meshless method of a thin plate, *ACTA MECHANICA SOLIDA SINICA*, Vol. 15 Issue: 3 pp. 244-258.

Long, SY; Zhang, Q (2002): Analysis of thin plates by the local boundary integral equation (LBIE) method, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 26 Issue: 8 pp. 707-718.

Lorentz, RA; Narcowich, FJ; Ward, JD (2003): Collocation discretizations of the transport equation with radial basis functions, *APPLIED MATHEMATICS AND COMPUTATION,* Vol. 145 Issue: 1 pp. 97-116.

Mahadevan, P; Dixit, US; Robi, PS (2007): Analysis of cold rigid-plastic axisymmetric forging problem by radial basis function collocation method, *International Journal of Manufacturing Technology,* Vol. 34, pp.464-473.

Mai-Cao, L; Tran-Cong ,T (2008): A meshless approach to capturing moving interfaces in passive transport problems. *CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES,* Vol. 31 Issue: 3 pp. 157-188.

Mai-Cao, L; Tran-Cong, T (2005): A meshless IRBFN-based method for transient problems, *CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES* Vol. 7 Issue: 2 pp. 149-171.

Mai-Duy, N; Mai-Cao, L, Tran-Cong, T (2007): Computation of transient viscous flows using indirect radial basis function networks, *CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES*, Vol. 18 Issue: 1 pp. 59-77 .

Mai-Duy, N; Khennane, A; Tran-Cong, T (2007): Computation of laminated composite plates using integrated radial basis function networks, *CMC-COMPUTERS MATERIALS & CONTINUA*, Vol. 5 Issue: 1 pp. 63-77.

Mai-Duy, N; Tran-Cong, T (2006): Solving biharmonic problems with scattered-point discretization using indirect radial-basis-function networks, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 30 Issue: 2 pp. 77-87.

Mai-Duy, N; Tanner, R (2005): Solving high-order partial differential equations with indirect radial basis function networks, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, Vol. 63 Issue: 11 pp. 1636-1654.

Mai-Duy, N; Tran-Cong, T (2005): An efficient indirect RBFN-based method for numerical solution of PDEs , *NUMERICAL METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS*, Vol. 21 Issue: 4 pp. 770-790.

Mai-Duy, N; Tanner, RI (2007): A collocation method based on one-dimensional RBF interpolation scheme for solving PDEs, *INTERNATIONAL JOURNAL OF NUMERICAL METHODS FOR HEAT & FLUID FLOW*, Vol. 17 Issue: 2 pp. 165-186.

Mai-Duy, N; Tran-Cong, T (2009): Integrated radial-basis-function networks for computing Newtonian and non-Newtonian fluid flows, *COM-*

PUTERS & STRUCTURES, Vol. 87 Issue: 11-12 Special Issue: Sp. Iss. SI pp. 642-650.

Mai-Duy, N; Le-Cao, K; Tran-Cong, T (2008): Cartesian grid technique based on one-dimensional integrated radial basis function networks for natural convection in concentric annuli, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS*, Vol. 57 Issue: 12 pp. 1709-1730.

Mai-Duy, N; Tran-Cong, T (2007): A Cartesian-grid collocation method based on radial-basis-function networks for solving PDEs in irregular domains, *NUMERICAL METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS,* Vol. 23 Issue: 5 pp. 1192-1210.

Mai-Duy, N; Tran-Cong, T (2008): A multidomain integrated-radial-basis-function collocation method for elliptic problems, *NUMERICAL METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS*, Vol. 24 Issue: 5 pp. 1301-1320.

Mai-Duy, N; Tran-Cong, T (2007): A Cartesian-grid collocation method based on radial-basis-function networks for solving PDEs in irregular domains, *NUMERICAL METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS,* Vol. 23 Issue: 5 pp. 1192-1210.

Mai-Duy, N; Tarmer, RI (2005): Computing non-Newtonian fluid flow with radial basis function networks, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS*, Vol. 48 Issue: 12 pp. 1309-1336.

Mai-Duy, N; Tran-Cong, T (2003): Indirect RBFN method with thin plate splines for numerical solution of differential equations, *CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES*, Vol. 4 Issue: 1 pp. 85-102.

Mai-Duy, N; Tran-Cong, T (2003): Neural networks for BEM analysis of steady viscous flows, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS*, Vol. 41 Issue: 7 pp. 743-763.

Mai-Duy, N; Tran-Cong ,T (2002): Mesh-free radial basis function network methods with domain decomposition for approximation of functions and numerical solution of Poisson's equations, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 26 Issue: 2 pp. 133-156.

Mai-Duy, N; Tran-Cong, T (2001): Numerical solution of Navier-Stokes equations using multiquadric radial basis function networks, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS*, Vol.

37 Issue: 1 pp. 65-86.

Mai-Duy, N; Tran-Cong, T .(2004): An effective RBFN-boundary integral approach for the analysis of natural convection flow, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS*, Vol. 46 Issue: 5 pp. 545-568.

Mai-Duy, N; Tran-Cong ,T (2003): RBF interpolation of boundary values in the BEM for heat transfer problems, *INTERNATIONAL JOURNAL OF NUMERICAL METHODS FOR HEAT & FLUID FLOW*, Vol. 13 Issue: 5-6 pp. 611-632.

Misra, RK; Datta, C (2008): Mechanical behavior of unidirectional glass fibers reinforced Resol/VAC-EHA Composites at different Vol. fraction of fibers, *SOFT MATERIALS,* Vol. 6 Issue: 3-4 pp. 99-118.

Misra, RK; Datta, C (2009): Mechanical Behavior of Polyethylene Fibers Reinforced Resol/VAC-EHA, *JOURNAL OF MACROMOLECULAR SCIENCE PART A-PURE AND APPLIED CHEMISTRY*, Vol. 46 pp. 425-437.

Misra, RK; Sandeep, K; Misra, A (2007): Analysis of anisotropic plate using multiquadric radial basis function, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 31 Issue: 1 pp. 28-34.

Moroney, TJ; Turner, IW (2007): A three-dimensional finite volume method based on radial basis functions for the accurate computational modelling of nonlinear diffusion equations, *JOURNAL OF COMPUTATIONAL PHYSICS*, Vol. 225 Issue: 2 pp. 1409-1426.

Morton, TM (2004): Two approaches to solving pseudodifferential equations on spheres using zonal kernels, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 28 Issue: 10 pp. 1227-1232.

Naffa, M; Al-Gahtani ,HJ (2007): RBF-based meshless method for large deflection of thin plates, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 31 Issue: 4 pp. 311-317.

Nam, MD (2004): Indirect RBFN method with scattered points for numerical solution of PDEs, *CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES*, Vol. 6 Issue: 2 pp. 209-226.

Nam, MD; Tranh, TC (2001): Numerical solution of differential equations using multiquadric radial basis function networks, *NEURAL NETWORKS,* Vol. 14 Issue: 2 pp. 185-199.

Narcowich, FJ; Ward, JD (1994): Generalized Hermite interpolation via matrix-valued conditionally positive-definite functions, *MATHEMATICS OF COMPUTATION,* Vol. 63 Issue: 208 pp. 661-687.

Orsini, P ;Power, H; Morvan, H (2008): Improving volume element methods by meshless radial basis function techniques , *CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES*, Vol. 23 Issue: 3 pp. 187-207.

Palaseanu, M; Pearlstine, L (2008): Estimation of water surface elevations for the Everglades, Florida, *COMPUTERS & GEOSCIENCES*, Vol. 34, Issue: 7 pp. 815-826.

Pepper, DW; Sarler, B (2005): Application of meshless methods for thermal analysis, *STROJNISKI VESTNIK-JOURNAL OF MECHANICAL ENGINEERING,* Vol. 51, Issue: 7-8 pp. 476-483.

Perko, J; Sarler, B (2007): Weight function shape parameter optimization in meshless methods for non-uniform grids, *CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES,* Vol. 19 Issue: 1 pp. 55-68.

Platte, RB; Driscoll, TA (2004): Computing eigenmodes of elliptic operators using radial basis functions, *COMPUTERS & MATHEMATICS WITH APPLICATION*S, Vol. 48 Issue: 3-4 pp. 561-576.

Platte, RB; Driscoll, TA (2005): Polynomials and potential theory for Gaussian radial basis function interpolation, *SIAM JOURNAL ON NUMERICAL ANALYSIS,* Vol. 43 Issue: 2 pp. 750-766.

Platte, RB; Driscoll, TA (2006): Eigenvalue stability of radial basis function discretizations for time-dependent problems, *COMPUTERS & MATHEMATICS WITH APPLICATIONS*, Vol. 51 Issue: 8 pp. 1251-1268.

Power, H; Barraco, V (2002): A comparison analysis between unsymmetric and symmetric radial basis function collocation methods for the numerical solution of partial differential equations, *COMPUTERS & MATHEMATICS WITH APPLICATIONS*, Vol. 43 Issue: 3-5 pp. 551-583.

Ramachandran, PA; Karur, SR (1998): Multidimensional interpolation using osculatory radial basis functions, *COMPUTERS & MATHEMATICS WITH APPLICATIONS*, Vol. 35 Issue: 11 pp. 63-73.

Recio, DP; Jorge, RMN; Dinis, LMS (2007): On the use of element-free Galerkin Method for problems involving incompressibility, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 31 Issue: 2 pp. 103-115.

Recio, DP; Jorge, RMN; Dinis, LMS (2006): Locking and hourglass phenomena in an element-free Galerkin context: the B-bar method with stabilization and an enhanced strain method, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, Vol. 68 Issue: 13 pp. 1329-1357.

Reutskiy, SY (2008): A Meshless Method for Nonlinear, Singular and Generalized Sturm-Liouville Problems, *CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES*, Vol. 34 Issue: 3 pp. 227-252 .

Reutskiy, SY (2009): The method of external excitation for solving Laplace singular eigenvalue problems, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 33 Issue: 2 pp. 209-214

Reutskiy, SY (2007): The methods of external and internal excitation for problems of free vibrations of non-homogeneous membranes, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 31 Issue: 11 pp. 906-918.

Roque, CMC; Ferreira, AJM; Jorge, RMN (2006): Free vibration analysis of composite and sandwich plates by a trigonometric layerwise deformation theory and radial basis functions, *JOURNAL OF SANDWICH STRUCTURES & MATERIALS*, Vol. 8 Issue: 6 pp. 497-515.

Roque, CMC; Ferreira, AJM; Jorge, RMN (2007): A radial basis function approach for the free vibration analysis of functionally graded plates using a refined theory, *JOURNAL OF SOUND AND VIBRATION*, Vol. 300 Issue: 3-5 pp. 1048-1070.

Roque, CMC; Ferreira, AJM (2009): New developments in the radial basis functions analysis of composite shells, *COMPOSITE STRUCTURES*, Vol. 87 Issue: 2 Special Issue: Sp. Iss. SI pp. 141-150.

Saeedpanah, I; Jabbari , E (2009): A Novel Approach to Local Radial Point Interpolation Meshless Method (LRPIM) for Numerical Solution of Groundwater Flow Response to Tidal Fluctuation in a Coastal Confined Aquifer , *JOURNAL OF COASTAL RESEARCH* , Vol. 2 Sp. Iss. 56 pp. 1050-1054.

Sanyasiraju, YVSS; Chandhini, G (2008): Local radial basis function based gridfree scheme for unsteady incompressible viscous flows, *JOURNAL OF COMPUTATIONAL PHYSICS*, Vol. 227 Issue: 20 pp. 8922-8948.

Sarler, B; Vertnik, R (2006): Meshfree explicit local radial basis function collocation method for diffusion problems , *COMPUTERS & MATHE-*

MATICS WITH APPLICATIONS, Vol. 51 Issue: 8 pp. 1269-1282.

Sarler, B; Jelic, N; Kovacevic, I; Lakner, M; Perko, J (2006): Axisymmetric multiquadrics, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 30 Issue: 2 pp. 137-142.

Sarler, B (2005): A radial basis function collocation approach in computational fluid dynamics,S*CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES*, Vol. 7 Issue: 2 pp. 185-193.

Sarler, B; Perko, J; Chen, CS (2004): Radial basis function collocation method solution of natural convection in porous media,*INTERNATIONAL JOURNAL OF NUMERICAL METHODS FOR HEAT & FLUID FLOW*, Vol. 14 Issue: 2 pp. 187-212.

Shanazari, K; Rabie, N (2009): A three dimensional adaptive nodes technique applied to meshless-type methods , *APPLIED NUMERICAL MATHEMATICS,* Vol. 59 6 1187-1197.

Shank, YY; Shu C; Lu, ZL (2008): Application of local MQ-DQ method to solve 3D incompressible viscous flows with curved boundary, *CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES*, Vol. 25 Issue: 2 pp. 99-113.

Shu, C; Wu, YL (2007): Integrated radial basis functions-based differential quadrature method and its performance. *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS,* Vol. 53 Issue: 6 pp. 969-984.

Shu, C; Ding , H; Zhao, N (2006): Numerical comparison of least square-based finite-difference (LSFD) and radial basis function-based finite-difference (RBFFD) methods, *COMPUTERS & MATHEMATICS WITH APPLICATIONS*, Vol. 51 Issue: 8 pp. 1297-1310.

Shu, C; Ding, H; Chen, HQ; Wang, TG (2005): An upwind local RBF-DQ method for simulation of inviscid compressible flows, *COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERIN*G, Vol. 194 Issue: 18-20 pp. 2001-2017.

Shu, C; Ding, H; Yeo, KS (2005): Computation of incompressible Navier-Stokes equations by local RBF-based differential quadrature method, *CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES*, Vol. 7 Issue: 2 pp. 195-205.

Shu, C; Ding, H; Yeo, KS (2004): Solution of partial differential equations by a global radial basis function-based differential quadrature method,

ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS, Vol. 28 Issue: 10 pp. 1217-1226.

Shu, C; Ding, H; Yeo, KS (2003): Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier-Stokes equations, *COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING*, Vol. 192 Issue: 7-8 pp. 941-954.

Singh, A; Singh, IV; Prakash, R (2007): Numerical analysis of fluid squeezed between two parallel plates by meshless method, *COMPUTERS & FLUIDS*, Vol. 36 Issue: 9 pp. 1460-1480.

Siraj-ul-Islam; Haq, S; Uddin, M (2009): A meshfree interpolation method for the numerical solution of the coupled nonlinear partial differential equations, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 33 Issue: 3 pp. 399-409.

Sobester, A; Nair, PB; Keane, AJ (2008): Genetic programming approaches for solving elliptic partial differential equations, *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, Vol. 12 Issue: 4 pp. 469-478.

Sterk, M; Trobec, R (2008): Meshless solution of a diffusion equation with parameter optimization and error analysis, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 32 Issue: 7 pp. 567-577.

Stevens, D; Power, H; Lees, M; Morvan, H (2009): The use of PDE centers in the local RBF Hermitian method for 3D convective-diffusion problems, *JOURNAL OF COMPUTATIONAL PHYSICS*, Vol. 228 Issue: 12 pp. 4606-4624.

Stevens, D; Power, H; Morvan, H (2009): An order-N complexity meshless algorithm for transport-type PDEs, based on local Hermitian interpolation, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS* , Vol. 33 Issue: 4 pp. 425-441.

Sukumar, N; Wright, RW (2007): Overview and construction of meshfree basis functions: From moving least squares to entropy approximants, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, Vol. 70 Issue: 2 pp. 181-205.

Takeuchi, T; Yamamoto, M (2008): Tikhonov regularization by a reproducing kernel Hilbert space for the Cauchy problem for an elliptic equation, *SIAM JOURNAL ON SCIENTIFIC COMPUTING*, Vol. 31 Issue: 1

pp. 112-142 .

Tiago, CM; Leitao, VMA (2006): Application of radial basis functions to linear and nonlinear structural analysis problems, *COMPUTERS & MATHEMATICS WITH APPLICATIONS,* Vol. 51 Issue: 8 pp. 1311-1334.

Tian, HY; Reutskiy, S; Chen, CS (2008): A basis function for approximation and the solutions of partial differential equations, *NUMERICAL METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS*, Vol. 24 Issue: 3 pp. 1018-1036

Tian, R; Yagawa, G (2007): Non-matching mesh gluing by meshless interpolation - An alternative to Lagrange multipliers, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, Vol. 71 Issue: 4 pp. 473-503

Tolstykh, AI; Shirobokov, DA (2005): Using radial basis functions in a "finite difference mode", *CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES*, Vol. 7 Issue: 2 pp. 207-222.

Tolstykh, AI; Lipavskii, MV; Shirobokov, DA (2003): High-accuracy discretization methods for solid mechanics, *ARCHIVES OF MECHANICS*, Vol. 55 Issue: 5-6 pp. 531-553.

Tolstykh, AI; Shirobokov ,DA (2003): On using radial basis functions in a "finite difference mode" with applications to elasticity problems, *COMPUTATIONAL MECHANICS*, Vol. 33 Issue: 1 pp. 68-79.

Trahan, CJ; Wyatt, RE (2003): Radial basis function interpolation in the quantum trajectory method: optimization of the multi-quadric shape parameter, *JOURNAL OF COMPUTATIONAL PHYSICS*, Vol. 185 Issue: 1 pp. 27-49.

Tran-Canh, D;Tran-Cong, T (2004): Element-free simulation of dilute polymeric flows using Brownian configuration fields, *KOREA-AUSTRALIA RHEOLOGY JOURNA*L, Vol. 16 Issue: 1 pp. 1-15.

Tran-Canh, D; Tran-Cong, T (2002): Computation of viscoelastic flow using neural networks and stochastic simulation, *KOREA-AUSTRALIA RHEOLOGY JOURNA*L, Vol. 14 Issue: 4 pp. 161-174.

Tran, CD; Phillips, DG (2007): Predicting torque of worsted singles yarn using an efficient radial basis function network-based method, *JOURNAL OF THE TEXTILE INSTITUTE*, Vol. 98 Issue: 5 pp. 387-396.

Tsai, CC (2007): The method of fundamental solutions for three-dimensional

elastostatic problems of transversely isotropic solids, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 31 Issue: 7 pp. 586-594.

Tsai, CC; Lin, YC; Young, DL; Aturi, SN (2006): Investigations on the accuracy and condition number for the method of fundamental solutions, *CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES*, Vol. 16 Issue: 2 pp. 103-114.

Tsai, CC; Young, D; Chen, CW; Fan, CM (2006): The method of fundamental solutions for eigenproblems in domains with and without interior holes, *PROCEEDINGS OF THE ROYAL SOCIETY A- MATHEMATICAL PHYSICAL AND ENGINEERING SCIENCES*, Vol. 462 Issue: 2069 pp. 1443-1466.

Tsai, CC; Young, DL; Cheng, AHD (2002): Meshless BEM for three-dimensional Stokes flows, *CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES*, Vol. 3 Issue: 1 pp. 117-128.

Uddin, M; Haq, S; Siraj-ul-Islam (2009): A mesh-free numerical method for solution of the family of Kuramoto-Sivashinsky equations, *APPLIED MATHEMATICS AND COMPUTATION*, Vol. 212 Issue: 2 pp. 458-469.

Valtchev, SS; Roberty, NC (2008): A time-marching MFS scheme for heat conduction problems, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 32 Issue: 6 pp. 480-493.

Vanani, SK; Aminatae, A. (2008): On the numerical solution of neutral delay differential equations using multiquadric approximation scheme, *BULLETIN OF THE KOREAN MATHEMATICAL SOCIETY*, Vol. 45 Issue: 4 pp. 663-670 .

Vanani, SK; Aminataei, A (2009): Multiquadric approximation scheme on the numerical solution of delay differential systems of neutral type, *MATHEMATICAL AND COMPUTER MODELLING,* Vol. 49 Issue: 1-2 pp. 234-241.

Vertnik, R; Zaloznik, M; Sarler, B (2006): Solution of transient direct-chill aluminium billet casting problem with simultaneous material and interphase moving boundaries by a meshless method, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 30 Issue: 10 pp. 847-855.

Vertnik, R; Sarler, B (2006): Meshless local radial basis function collocation method for convective-diffusive solid-liquid phase change problems,

INTERNATIONAL JOURNAL OF NUMERICAL METHODS FOR HEAT & FLUID FLOW, Vol. 16 Issue: 5 pp. 617-640.

Vrankar, L; Runovc, F; Turk; G (2007): The use of the mesh free methods (radial basis functions) in the modeling of radionuclide migration and moving boundary value problems, *ACTA GEOTECHNICA SLOVENICA,* Vol. 4 Issue: 1 pp. 42-53.

Vrankar, L; Turk, G; Runovc, F (2004): Combining the radial basis function Eulerian and Lagrangian schemes with geostatistics for modeling of radionuclide migration through the geosphere ,*COMPUTERS & MATHEMATICS WITH APPLICATIONS*, Vol. 48 Issue: 10-11 pp. 1517-1529.

Vrankar, L; Turk, G; Runovc, F (2004): Modelling of radionuclide migration through the geosphere with radial basis function method and geostatistics, *JOURNAL OF THE CHINESE INSTITUTE OF ENGINEERS,* Vol. 27 Issue: 4 pp. 455-462.

Wang, H; Qin, QH (2007): Some problems with the method of fundamental solution using radial basis functions , *ACTA MECHANICA SOLIDA SINICA*, Vol. 20 Issue: 1 pp. 21-29.

Wang, SY; Wang, MY (2006): Radial basis functions and level set method for structural topology optimization, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, Vol. 65 Issue: 12 pp. 2060-2090.

Wen, PH; Hon, YC (2007): Geometrically nonlinear analysis of Reissner-Mindlin plate by meshless computation, *CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES*, Vol. 21 Issue: 3 pp. 177-191.

Wendland, H (2007): On the stability of meshless symmetric collocation for boundary value problems, *BIT NUMERICAL MATHEMATICS,* Vol. 47 Issue: 2 pp. 455-468.

Wertz, J; Kansa, EJ; Ling, L (2006): The role of the multiquadric shape parameters in solving elliptic partial differential equations, *COMPUTERS & MATHEMATICS WITH APPLICATIONS*, Vol. 51 Issue: 8 pp. 1335-1348.

Wong, ASM; Hon ,YC; Li, TS; Chung, SL; Kansa, EJ (1999): Multizone decomposition for simulation of time-dependent problems using the multiquadric scheme, *COMPUTERS & MATHEMATICS WITH APPLICATIONS,* Vol. 37 Issue: 8 pp. 23-43.

Wong, SM; Li, TS (2006): Domain decomposition with radial basis functions for solving singularity problem, *DYNAMICS OF CONTINUOUS DISCRETE AND IMPULSIVE SYSTEMS-SERIES A-MATHEMATICAL ANALYSIS*, Vol. 13 pp. 501-513.

Wright, GB; Fornberg, B (2006): Scattered node compact finite difference-type formulas generated from radial basis functions, *JOURNAL OF COMPUTATIONAL PHYSICS*, Vol. 212 Issue: 1 pp. 99-123.

Wu, LY; Chung, LL; Huang, HH (2008): Radial spline collocation method for static analysis of beams, *APPLIED MATHEMATICS AND COMPUTATION*, Vol. 201 Issue: 1-2 pp. 184-199.

Wu, NJ; Tsay, TK; Young, DL (2008): Computation of nonlinear free-surface flows by a meshless numerical method, *JOURNAL OF WATERWAY PORT COASTAL AND OCEAN ENGINEERING-ASCE,* Vol. 134 Issue: 2 pp. 97-103.

Wu, NJ; Tsay, TK; Young, D (2006): Meshless numerical simulation for fully nonlinear water waves, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS*, Vol. 50 Issue: 2 pp. 219-234.

Wu, WX; Shu, C; Wang, CM (2007): Vibration analysis of arbitrarily shaped membranes using local radial basis function-based differential quadrature method, *JOURNAL OF SOUND AND VIBRATION*, Vol. 306 Issue: 1-2 pp. 252-270.

Wu, YL; Shu. C; Chen, HQ. (2004): Radial basis function-enhanced domain-free discretization method and its applications, *NUMERICAL HEAT TRANSFER PART B-FUNDAMENTALS,* Vol. 46 Issue: 3 pp. 269-282.

Wu, ZM; Hon, YC (2003): Convergence error estimate in solving free boundary diffusion problem by radial basis functions method, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 27 Issue: 1 pp. 73-79.

Wu, YL; Shu, C (2002): Development of RBF-DQ method for derivative approximation and its application to simulate natural convection in concentric annuli, *COMPUTATIONAL MECHANICS* , Vol. 29 Issue: 6 pp. 477-485.

Wu, YL; Liu, GR (2003): A meshfree formulation of local radial point interpolation method (LRPIM) for incompressible flow simulation, *COMPUTATIONAL MECHANICS,* Vol. 30 Issue: 5-6 pp. 355-365.

Wu, ZM (2004): Dynamically knots setting in meshless method for solving time dependent propagations equation, *COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING*, Vol. 193 Issue: 12-14 pp. 1221-1229.

Xiang, S; Wang, KM (2009):, Free vibration analysis of symmetric laminated composite plates by trigonometric shear deformation theory and inverse multiquadric RBF, *THIN-WALLED STRUCTURES*, Vol. 47 Issue: 3 pp. 304-310 .

Xiao, JR; Gama, BA; Gillespie, JW; Kansa, EJ (2005): Meshless solutions of 2D contact problems by subdomain variational inequality and MLPG method with radial basis functions, *ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 29 Issue: 2 pp. 95-106.

Xiong, SW; Liu, WK; Cao, H; Li, CS; Rodrigues, JMC; Martins, PAF (2005): Simulation of bulk metal forming processes using the reproducing kernel particle method, *COMPUTERS & STRUCTURES*, Vol. 83 Issue: 8-9 pp. 574-587.

Xiao, JR; Gilhooley, DF; Batra, RC; Gillespie, JW; McCarthy, MA (2008): Analysis of thick composite laminates using a higher-order shear and normal deformable plate theory (HOSNDPT) and a meshless method, *COMPOSITES PART B-ENGINEERING*, Vol. 39 Issue: 2 pp. 414-427.

Yang, JG; Guo, R; Tian, YW (2008): Hybrid radial basis function/finite element modelling of journal bearing, *TRIBOLOGY INTERNATIONAL*, Vol. 41 Issue: 12 pp. 1169-1175.

Young, DL; Chen, CS; Wong, TK (2005): Solution of Maxwell's equations using the MQ method, *CMC-COMPUTERS MATERIALS & CONTINUA*, Vol. 2 Issue: 4 pp. 267-276.

Young, DL; Jane, SC; Lin, CY; Chiu,CL; Chen,KC (2004): Solutions of 2D and 3D Stokes laws using multiquadrics method ,*ENGINEERING ANALYSIS WITH BOUNDARY ELEMENTS*, Vol. 28, Issue: 10 pp. 1233-1243.

Young, DL; Tsai ,CC; Fan, CM (2004): Direct approach to solve nonhomogeneous diffusion problems using fundamental solutions and dual reciprocity methods, *JOURNAL OF THE CHINESE INSTITUTE OF ENGINEERS*, Vol. 27 Issue: 4 pp. 597-609.

Zerroukat, M; Power, H; Chen, CS (1998): A numerical method for heat transfer problems using collocation and radial basis functions, *INTER-*

NATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING, Vol. 42 Issue: 7 pp. 1263-1278.

Zerroukat, M; Djidjeli, K; Charafi, A (2000): Explicit and implicit meshless methods for linear advection-diffusion-type partial differential equations , *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, Vol. 48 Issue: 1 pp. 19-35.

Zhang, X; Song, KZ; Lu, MW; Liu, X (2000): Meshless methods based on collocation with radial basis functions, *COMPUTATIONAL MECHANICS,* Vol. 26 Issue: 4 pp. 333-343.

Zhang, YX; Tan ,YJ (2006): Solve partial differential equations by meshless subdomains method combined with RBFs, *APPLIED MATHEMATICS AND COMPUTATION* , Vol. 174 Issue: 1 pp. 700-709.

Zhang, YX; Tan, YJ (2005): Solving partial differential equations by BKM combined with DDM, *APPLIED MATHEMATICS AND COMPUTATION*, Vol. 171 Issue: 2 pp. 1004-1015.

Zhang, XK; Kwon, KC; Youn, SK (2004): Least-squares meshfree method for incompressible Navier-Stokes problems, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS*, Vol. 46 Issue: 3 pp. 263-288.

Zhou, X; Hon, YC; Sun, S; Mak, AFT (2002): Numerical simulation of the steady-state deformation of a smart hydrogel under an external electric field, *SMART MATERIALS & STRUCTURES*, Vol. 11 Issue: 3 pp. 459-467.

Zhou, X; Hon, YC; Li, JC (2003): Overlapping domain decomposition method by radial basis functions, *APPLIED NUMERICAL MATHEMATICS*, Vol. 44 Issue: 1-2 pp. 241-255.

Zuppa,C; Cardona, A (2003): A collocation meshless method based on local optimal point interpolation, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, Vol. 57 Issue: 4 pp. 509-536.

Bibliography

[1] R. C. Aldredge. Premixed flame propagation in a high-intensity, large-scale vortical flow. *Combustion and Flame*, 106:29–40, 1996. 6.3, 6.3

[2] C. Alves and H. Leita. Crack analysis using an enriched mfs domain decomposition technique. *Engineering Analysis with Boundary Elements*, 30:160–166, 2006. 7.12

[3] S. N. Atluri. *The Meshless Method (MLPG) for Domain & BIE Discretizations*. Tech Science Press, 2004. (document)

[4] S. N. Atluri and S. Shen. *The Meshless Local Petrov-Galerkin (MLPG) Method*. Tech Science Press, 2002. (document)

[5] V. A. Baikov and K. R. Khusnutdinova. Formal linearization and exact solutions of some nonlinear partial differential equations. *Nonlinear Mathematical Physics*, 3:139–146, 1996. 7.9

[6] D. H. Bailey. High-precision arithmetic in scientific computation. *Computing in Science and Engineering*, pages 54–61, 2005. 4.1

[7] D. H. Bailey and Jonathan M. Borwein. High-precision computation and mathematical physics. *To appear in XII Advanced Computing and Analysis Techniques in Physics Research*, 2008. 4.1

[8] K. Balakrishnan, R. Sureshkumar, and P. A. Ramachandran. An operator splitting radial basis function method for the solution of transient nonlinear poisson problems. *Computers and Mathematics with Applications*, 43:289–304, 2002. 8.3

[9] R. Beatson and G. Newsam. Fast evaluation of radial basis functions: I. *Computers and Mathematics with Applications*, 24(12):7–19, 1992. 8.3

[10] R. Beatson and G. Newsam. Fast evaluation of radial basis functions: Moment-based methods. *SIAM Journal of Scientific Computing*, 19(5):1428–1449, 1998. 1.1

[11] R. K. Beatson and E. Chacko. Fast evaluation of radial basis functions: A multivariate momentary evaluation scheme. In A. Cohen, C. Rabut, and L. L. Schumaker, editors, *Curve and Surface Fitting: Saint Malo 1999*, pages 37–46. Vanderbilt University Press, Nashville, 2000. 8.3

[12] R. K. Beatson, J. B. Cherrie, and C. T. Mouat. Fast fitting of radial basis functions: Methods based on preconditioned gmres iteration. *Advances in Computational Mathematics*, 11:253–270, 1999. 4.5, 4.5.1, 4.5.1

[13] R. K. Beatson, W. A. Light, and S. Billings. Fast solution of the radial basis function interpolation equations: Domain decomposition methods. *SIAM Journal of Scientific Computing*, 22(5):1717–1740, 2000. 8.3

[14] R. K. Beatson and M. J. D. Powell. Univariate multiquadric approximation: Quasi-interpolation to scattered data. *Constructive Approximation*, 8:275–288, 1992. 8.3

[15] H. Behrens and A. Iske. Grid-free adaptive semi-lagrangian advection using radial basis functions. *Computers and Mathematics with Applications*, 43:391–327, 2002. 5.7

[16] H. Behrens, A. Iske, and M. Kaser. Adaptive meshfree method of backward characteristics for nonlinear transport equations. In *Meshfree methods for partial differential equations*, volume 26 of *Lecture Notes in Computer Science and Engineering*, pages 21–36. Springer, Berlin, 2003. 5.7

[17] R. E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957. 7.2

[18] T. Belytschko, N. Mőes, S. Usui, and C. Parimi. Arbitrary discontinuities in finite elements. *International Journal for Numerical Methods in Engineering*, 50:993–1013, 2001. 7.12

[19] F. Bernal. *Meshless Methods for Elliptic and Free-Boundary Problems*. PhD thesis, University of Carlos III De Madrid, 2008. 3.1.2

[20] F. Bernal, G. Gutierrez, and M. Kindelan. Use of singularity capturing functions in the solution of problems with discontinuous boundary conditions. *Engineering Analysis with Boundary Elements*, 33:200–208, 2009. 7.12

[21] F. Bernal and M. Kindelan. Rbf meshless modeling of non-newtonian hele-shaw flow. *Engineering Analysis with Boundary Elements*, 31:863–874, 2007. 3.1.2

[22] F. Bernal and M. Kindelan. On the enriched rbf method for singular potential problems. *To appear in Engineering Analysis with Boundary Elements*, 2009. 8.3

[23] G. W. Bluman and S. Kumei. *Symmetries and Differential Equations*. Springer, New York, 1989. 7.9

[24] M. Bozzini, L. Lenarduzzi, M. Rossini, and R. Schaback. Interpolation by basis functions of different scales and shapes. *Calcolo*, 41(2):77–87, 2004. 5.2

[25] M. Bozzini, L. Lenarduzzi, and R. Schaback. Adaptive interpolation by scaled multiquadrics. *Advances in Computational Mathematics*, 16:375–387, 2002. 5.7

[26] C. Bresten, S. Gottlieb, D. Higgs, and J.-H. Jung. Recovery of high order accuracty in radial basis function approximaton for discontinuous problems. *Preprint*, 2008. 5.5.3

[27] D. Brown, L. Ling, E. J. Kansa, and J. Levesley. On approximate cardinal preconditioning methods for solving pdes with radial basis functions. *Engineering Analysis with Boundary Elements*, 29(4):343–353, 2005. 4.5.1

[28] M. D. Buhmann. Spectral convergence of multiquadric interpolation. *Proceedings of the Edinburgh Mathematical Society*, 36:319–333, 1993. 2.4

[29] M. D. Buhmann. Multiquadric pre-wavelets on non-equally spaced knots in one dimension. *Mathematics of Computation*, 64:1611–1625, 1995. 5.4

[30] M. D. Buhmann. A new class of radial basis functions with compact support. *Mathematics of Computation*, 70(233):307–318, 2000. 8.3

[31] M. D. Buhmann. *Radial Basis Functions*. Cambridge University Press, 2003. (document), 2

[32] M. D. Buhmann and S. Dinew. Limites of radial basis function interpolants. *Communication on Pure and Applied Analysis*, 6(3):569–585, 2007. 5.3

[33] M. D. Buhmann, S. Dinew, and E. Larsson. A note on radial basis function interpolant limits. *To appear in IMA Journal of Numerical Analysis*, 2009. 5.3

[34] J. Butcher. *Numerical Methods for Ordinary Differential Equations*. Wiley, 2003. 3.2.1

[35] S. Campbell, I. Ipsen, C. Kelley, and C. Meyer. Gmres and the minimal polynomial. *BIT*, pages 664–675, 1996. 4.5

[36] Matlab Central. http://www.mathworks.com/matlabcentral/fileexchange/. 5.7

[37] S. Chantasiriwan. Solutions of partial differential equations with random dirichlet boundary conditions by multiquadric collocation method. *Engineering Analysis with Boundary Elements*, 29:1124–1129, 2005. 8.2

[38] S. Chantasiriwan. Error and variance of solution to the stochastic heat conduction problem by multiquadric collocation method. *International Communications in Heat and Mass Transfer*, 33:342349, 2006. 8.3

[39] C. S. Chen, Y. F. Rashed, and M. A. Golberg. A mesh-free method for linear diffusion equations. *Numerical Heat Transfer Part B*, 33:469–486, 1998. 8.3

[40] R. Chen and Z. Wu. Applying multiquadric quasi-interpolation to solve burgers' equation. *Applied Mathematics and Computation*, 172:472–484, 2006. 8.3

[41] R. Chen and Z. Wu. Solving partial differential equation by using multiquadric quasi-interpolation. *Applied Mathematics and Computation*, 186:1502–1510, 2007. 8.3

[42] Y. Chen, J. Lee, and A. Eskandarian. *Meshless Methods in Solid Mechanics*. Springer, 2006. (document)

[43] A. H.-D. Cheng and J. Cabral. Direct solution of ill-posed boundary value problems by radial basis function collocation method. *International Journal of Numerical Methods in Engineering*, 64:45–64, 2005. 8.2

[44] M. Chenoweth. A numerical study of generalized multiquadric radial basis function interpolation. *Submitted to SIAM Undergraduate Research Online*, 2009. 2.10

[45] J. B. Cherrie, R. K. Beatson, and G. N. Newsam. Fast evaluation of radial basis functions: Methods for generalized multiquadrics in $\mathcal{R}^n$. *SIAM Journal of Scientific Computing*, 23(5):1549–1571, 2002. 8.3

[46] P. Chinchapatnam, k. Djidjeli, and P. Nair. Domain decomposition for time-dependent problems using radial based meshless methods. *Numerical Methods for Partial Differential Equations*, 2006. 4.6

[47] C. Chui. Wavelets and spline interpolation. *Advances in numerical analysis volume II: Wavelets, subdivision algorithms, and radial basis functions*, pages 1–35, 1992. 5.4

[48] C. K. Chui, J. Stockler, and J. Ward. Analytic wavelets generated by radial functions. *Advances in Computational Mathematics*, 5:95–123, 1996. 5.4

[49] M. E. da Silva and T. Barbe. Quasi-monte carlo in finance: extending for problems of high effective dimension. *Economia Aplicada*, 9:577–594, 2005. 7.5, 7.8

[50] N. Daux, N. Sukumar, and T. Belytschko. Arbitrary branched and intersecting cracks with the extended finite element method. *International Journal for Numerical Methods in Engineering*, 48:1741, 2000. 7.12

[51] A. Donato and F. Oliveri. Linearization procedure of nonlinear first order systems of partial differential equations by means of canonical variables related to lie groups of point transformations. *Journal of Mathematical Analysis. and Applications*, 188:552–568, 1994. 7.9

[52] T. A. Driscoll and B. Fornberg. Interpolation in the limit of increasingly flat radial basis functions. *Computers and Mathematics with Applications*, 43:413–422, 2002. 1, 5.3

[53] T. A. Driscoll and A. Heryudono. Adaptive residual subsampling methods for radial basis function interpolation and collocation problems. *Computers and Mathematics with Applications*, 53:927–939, 2007. 3.1, 5.7

[54] V. Durante and J.-H. Jung. An iterative adaptive multiquadric radia basis function method for the detection of local jump discontinuities. *To appear in Applied Numerical Mathematics*, 2008. 5.5.1, 5.5.3

[55] O. H. El-Kalaawy. Exact soliton solutions for some nonlinear partial differential equations. *Chaos, Solitons and Fractals*, 14:547–552, 2002. 7.9

[56] A. Emdadi, E. J. Kansa, N. Ali Libre, M. Rahimian, and M. Shekarchi. Stable pde solution methods for large multiquadric shape parameters. *Computer Modeling In Engineering And Sciences*, 25(1):23–42, 2008. 4.3.2, 4.3.2

[57] I. Faragó and B. Gnandt. Additive and iterative operator splitting methods and their numerical investigation. *Computers and Mathematics with Applications*, 55(10):2266–2279, 2008. 7.3

[58] C. Farhat, I. Harari, and L. Franca. The discontinuous enrichment method. *Computer Methods in Applied Mechanics and Engineering*, 190:6455–6479, 2001. 7.12

[59] S. Farlow. *Partial Differential Equations for Scientists and Engineers*. Dover, 1982. 3

[60] G. E. Fasshauer. Newton iteration with multiquadrics for the solution of nonlinear pdes. *Computer and Mathematics with Applications*. 3.1.2

[61] G. E. Fasshauer. Solving partial differential equations by collocation with radial basis functions. In C. Rabut A. Le Mehaute and L. L. Schumaker, editors, *Surface Fitting and Multiresolution Methods*, pages 131–138. Vanderbilt University Press, 1997. 8.3

[62] G. E. Fasshauer. Hermite interpolation with radial basis functions on spheres. *Advances in Computational Mathematics*, 10:81–96, 1999. 8.3

[63] G. E. Fasshauer. *Meshfree Approximation Methods with Matlab*. World Scientific, 2007. (document), 2, 2.11

[64] G. E. Fasshauer, C. Gartland, and J. Jerome. Newton iteration for partial differential equations and the approximation of the identity. *Numerical Algorithms*. 3.1.2

[65] G. E. Fasshauer, A. Khaliq, and D. Voss. A parallel time stepping approach using meshfree approximations for pricing options with non-smooth payoffs. In *Proceedings of Third World Congress of the Bachelier Finance Society*, 2004. 8.3

[66] G. E. Fasshauer, A. Khaliq, and D. Voss. Using meshfree approximation for multi-asset american option problems. *Journal of the Chinese Institute of Engineers*, 27(4):563–571, 2004. 8.3

[67] G. E. Fasshauer and J. Zhang. On choosing "optimal" shape parameters for rbf approximation. *Numerical Algorithms*. 5.1

[68] A. I. Fedoseyev, M. J. Friedman, and E. J. Kansa. Continuation for nonlinear elliptic partial differential equations discretized by the multiquadric method. *International Journal of Bifurcation and Chaos*, 10(2):481–492, 2000. 3.1.2

[69] A. I. Fedoseyev, M. J. Friedman, and E. J. Kansa. Improved multiquadric method for elliptic partial differential equations via pde collocation on the boundary. *Computers and Mathematics with Applications*, 43(3-5):491–500, 2002. 3.1, 5.9, 5.9

[70] Z. Fei, Y. Goto, and E. Kita. Solution of black-scholes equation by using rbf approximation. In Y. Kaneda, H. Kawamura, and M. Sasai, editors, *Frontiers of Computational Science*, pages 339–343. Springer, 2007. 7.1

[71] A. Ferreira, E. J. Kansa, G. E. Fasshauer, and V. Leito, editors. *Progress on Meshless Methods*. Springer, 2008. (document)

[72] N. Flyer. Exact polynomial reproduction for oscillaory radial basis functions on infinite lattices. *Computers and Mathematics with Applications*, 51:1199–1208, 2006. 8.3

[73] N. Flyer and G. Wright. Transport schemes on a sphere using radial basis functions. *Journal of Computational Physics*, 226:1059–1084, 2007. 8.3

[74] N. Flyer and G. Wright. A radial basis function method for the shallow water equations on a sphere. *Proceedings of the Royal Society A*, 465:1949–1976, 2009. 8.3

[75] T. A. Foley. Near optimal parameter selection for multiquadric interpolation. *Journal of Applied Science and Computation*, 1:54–69, 1994. 5.1

[76] B. Fornberg, T. Dirscol, G. Wright, and R. Charles. Observations on the behavior of radial basis function approximations near boundaries. *Computers and Mathematics with Applications*, 43:473–490, 2002. 2.8

[77] B. Fornberg and N. Flyer. *The Gibbs Phenomenon in Various Representations and Applications*, chapter The Gibbs phenomenon for radial basis functions. Sampling Publishing, Potsdam, NY, 2008. 5.5

[78] B. Fornberg, E. Larsson, and G. Wright. A new class of oscillatory radial basis functions. *Computers and Mathematics with Applications*, 51:1209–1222, 2006. 8.3

[79] B. Fornberg and C. Piret. A stable algorithm for flat radial basis functions on a sphere. *SIAM Journal of Scientific Computing*, 30:60–80, 2007. 8.3

[80] B. Fornberg and C. Piret. On choosing a radial basis function and a shape parameter when solving a convective pde on a sphere. *Journal of Computational Physics*, 227:2758–2780, 2008. 8.3

[81] B. Fornberg and G. Wright. Stable computation of multiquadric interpolants for all values of the shape parameter. *Computers and Mathematics with applications*, 48:853–867, 2004. 4.2, 4.2

[82] B. Fornberg, G. Wright, and E. Larsson. Some observations regarding interpolants in the limit of flat radial basis functions. *Computers and Mathematics with Applications*, 43:37–55, 2004. 5.3

[83] C. Franke and R. Schaback. Solving partial differential equations by collocation using radial basis functions. *Applied Mathematics and Computation*, pages 73–82, 1998. 8.3

[84] R. Franke. A critical comparison of some methods for the interpolation of scattered data. Technical Report NPS-53-79-03, Naval Postgraduate School, 1979. 1, 5.1

[85] R. Franke. Scattered data interpolation: Tests of some methods. *Mathematics of Computation*, pages 181–200, 1982. 1, 2.2

[86] E. J. Fuselier, F. J. Narcowich, J. D. Ward, and G.B. Wright. Error and stability estimates for surface-divergence free rbf interpolants on the sphere. *To appear in Mathematics of Computation*, 2009. 8.3

[87] B. Ganapathysubramanian and N. Zabaras. Sparse grid collocation schemes for stochastic natural convection problems. *Journal of Computational. Physics*, 225:652–685, 2007. 7.5, 7.8

[88] T. Gerstner and M. Griebel. Dimension-adaptive tensor-product quadrature. *Computing*, (71):65–87, 2003. 7.4, 7.8

[89] F. Girosi. Some extenstions of radial basis functions and their applications in artificial intellegence. 7.8

[90] D. Gottlieb and C.-W. Shu. On the Gibbs phenomenon and its resolution. *SIAM Review*, 39(4):644–668, 1997. 5.5.3

[91] S. Gottlieb. (Private Communication), 2008. 8.1

[92] M. Griebel and M. Schweitzer, editors. *Meshfree Methods for Partial Differential Equations*. Springer, 2002. (document)

[93] M. Griebel and M. Schweitzer, editors. *Meshfree Methods for Partial Differential Equations II*. Springer, 2005. (document)

[94] M. Griebel and M. Schweitzer, editors. *Meshfree Methods for Partial Differential Equations III*. Springer, 2006. (document)

[95] M. Griebel and M. Schweitzer, editors. *Meshfree Methods for Partial Differential Equations IV*. Springer, 2008. (document)

[96] P. C. Hansen. *Rank-deficient an discrete ill-posed problems: Numerical Aspects of Linear Inversion*. SIAM Publications, 1997. 4.3.1

[97] R. Hardy. Multiquadric equations of topography and other irregular surfaces. *Journal of Geophysical Research*, 76(8). 1, 2.11, 5.1

[98] R. Hardy. Least square prediction. *Photogramm. Engng. Rem. Sens.*, 43:475–492, 1977. 4.6

[99] R. L. Hardy. Theory and applications of the multiquadric-biharmonic method. *Computers and Mathematics with Applications*, 19(8/9):163–208, 1990. 1, 2.11

[100] J. Hesthaven, S. Gottlieb, and D. Gottlieb. *Spectral Methods for Time-Dependent Problems*. Cambridge University Press, 2007. 1.1, 5.3

[101] J. S. Hesthaven. Spectral penalty methods. *Applied Numerical Mathematics*, 33:23–41, 2000. 8.1

[102] Y. C. Hon. A rbfs method for solving options pricing model. In *Proceedings of Advances in Scientific Computing & Modeling*, 1998. 8.3

[103] Y. C. Hon. A radial basis function method for solving options pricing model. *Financial Engineering*, 8(1):31–49, 1999. 8.3

[104] Y. C. Hon. A comparison on using various radial basis functions for options pricing. *International Journal of Applied Science & Computations*, 7(1):29–47, 2000. 8.3

[105] Y. C. Hon. A quasi-interpolation method for solving stiff ordinary differential equations. *International Journal of Numerical Methods in Engineering*, 48:1187–1197, 2000. 8.3

[106] Y. C. Hon. A grid-free, nonlinear shallow-water model with moving boundary. *Engineering Analysis with Boundary Elements*, 88:967–973, 2004. 8.3

[107] Y. C. Hon and X. Mao. An efficient numerical scheme for Burger's equation. *Appl. Math. Comput.*, pages 37–50, 1998. 2.6, 5.7

[108] Y. C. Hon and R. Schaback. On unsymmetric collocation by radial basis function. *Applied Mathematics and Computations*, 119:177–186, 2001. 3

[109] Y. C. Hon and T. Wei. Backus-gilbert algorithm for a cauchy problem of laplace equation. *Inverse Problems*, 17:261–271, 2001. 8.2

[110] Y. C. Hon and Z. Wu. A numerical computation for inverse boundary determination problem. *Engineering Analysis with Boundary Elements*, 24:599–606, 2000. 8.2

[111] Y. C. Hon and Z. Wu. Additive schwarz domain decomposition with radial basis approximation. *International Journal of Applied Mathematics and Statistics*, 4:81–98, 2002. 7.1

[112] X. G. Hu, T. S. Ho, H. Rabnitz, and A. Askar. Multivariate radial basis function interpolation for solving quantum fluid dynamical systems. *Computers and Mathematics with Applications*, 43(3):525–537, 2002. 7.10, 7.10

[113] C.-S. Huang, C.-F. Leeb, and A.H.-D. Cheng. Error estimate, optimal shape factor, and high precision computation of multiquadric collocation method. *Engineering Analysis with Boundary Elements*, 31:614–623, 2007. 4.1

[114] M. S. Ingber, C. S. Chen, and J. A. Tanski. A mesh free approach using radial basis functions and parallel domain decomposition for solving three-dimensional diffusion equations. *International Journal For Numerical Methods In Engineering*, 60:2183–2201, 2004. 4.6, 4.6

[115] L. Jameson and T. Miyama. Wavelet analysis and ocean modeling: A dynamically adaptive numerical method "wofd-aho". *Monthly Weather Review*, pages 1536–1548, 2000. 5.4.1

[116] J. H. Jung. A note on the Gibbs phenomenon with multiquadric radial basis functions. *Applied Numerical Mathematics*, 57:213–219, 2007. 5.2, 5.5.1

[117] E. J. Kansa. Multiquadrics - a scattered data approximation scheme with applications to computational fluid dynamics I: Surface approximations and partial derivative estimates. *Computers and Mathematics with Applications*, 19(8/9):127–145, 1990. 1, 5.2, 5.2

[118] 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. *Computers and Mathematics with Applications*, 19(8/9):147–161, 1990. 1, 3.2.1, 3.2.3

[119] E. J. Kansa. A strictly conservative spatial approximation scheme for the governing engineering and physics equations over irregular regions and inhomogeneously scattered nodes. *Computers and Mathematics with Applications*, 24(5/6):169–190, 1992. 8.3

[120] E. J. Kansa. Local, point-wise rotational transformations of the conservation equations into stream-wise coordinates. *Computers and Mathematics with Applications*, 43:501–511, 2002. 6.1.2, 6.1.3

[121] E. J. Kansa. Exact explicit time integration of hyperbolic partial differential equations with mesh free radial basis functions. *Engineering Analysis with Boundary Elements*, 31:577–585, 2007. 6.1.3, 7.11

[122] E. J. Kansa, R.C. Aldredge, and L. Ling. Numerical simulation of two-dimensional combustion using mesh-free methods. *To appear in Engineering Analysis with Boundary Elements*, 2009. 6.1.3, 6.2, 6.3, 6.3, 6.3, 7.11, 7.12

[123] E. J. Kansa and R. Carlson. Improved accuracy of multiquadric interpolation using variable shape parameters. *Computers and Mathematics with Applications*, 24(12):99–120, 1992. 5.2

[124] E. J. Kansa and Y.C. Hon. Circumventing the ill-conditioning problem with multiquadric radial basis fuctions: Applications to elliptic partial differential equations. *Computers and Mathematics with Applications*, 39(7/8):123–137, 2000. 3.1

[125] E. J. Kansa, H. Power, G. E. Fasshauer, and L. Ling. A volumetric integral radial basis function method for time-dependent partial differential equation: I. formulation. *Journal of Engineering Analysis with Boundary Elements*, 28:1191–1206, 2004. 5.8, 6.1.3, 6.1.3

[126] D. N. Kontinos. *Rotated upwind algorithms for solution of the two and three dimensional Euler and Navier-Stokes equations*. PhD thesis, North Carolina State University, 1994. 6.1.2

[127] C. B. Laney. *Computational Gas Dynamics*. Cambridge University Press, 1998. 6.1.3, 6.2

[128] E. Larrson and B. Fornberg. Theoretical and computational aspects of multivariate interpolation with increasing flat radial basis functions. *Computers and Mathematics with Applications*, 49(1):103–130, 2005. 1, 5.3

[129] E. Larsson and B. Fornberg. A numerical study of some radial basis function based solution methods for elliptic pdes. *Computers and Mathematics with Applications*, 46:891–902, 2003. 2.8, 3.1, 5.9

[130] E. Larsson, K. hlander, and A. Hall. Multi-dimensional option pricing using radial basis functions and the generalized fourier transform. *Journal of Computational and Applied Mathematics*, 222:175–192, 2008. 8.3

[131] D. Lazzaro and L. Montefusco. Radial basis functions for the multivariate interpolation of large scattered data sets. *Journal of Computationa and Applied Mathematics*, 140:521–536, 2002. 8.3

[132] Y. J. Lee, G. J. Yoon, and J. Yoon. Convergence property of increasingly flat radial basis function interpolation to polynomial interpolation. *To appear in SIAM Journal of Mathematical Analysis*. 1, 5.3

[133] V. Leito, C. Alves, and C. Durate, editors. *Advances in Meshfree Techniques*. Springer, 2007. (document)

[134] D. W. Levey, K. G. Powell, and B. VanLeer. Use of a rotated riemann solver for the 2-dimensional euler equations. *Journal of Computational Physics*, 106:201–214, 1993. 6.1.2

[135] J. Li and C. S. Chen. Some observations on unsymmetric radial basis function collocation methods for convection-diffusion problems. *International Journal for Numerical Methods in Engineering*, 57:1085–1094, 2003. 8.3

[136] J. Li, H.-D. Cheng, and C.-S. Chen. A comparison of efficiency and error convergence of multiquadric collocation method and finite element method. *Engineering Analysis with Boundary Elements*, 27:251–257, 2003. (document)

[137] J. Li and Y. C. Hon. Domain decomposition for radial basis meshless methods. *Numerical Methods for Partial Differential Equations*, 20(3):450–462, 2004. 4.6

[138] Z.-C. Li and H.-T. Huang. Effective condition number for numerical partial differential equations. *Numerical Linear Algebra with Applications*, 15:575–594, 2008. 2.6

[139] N. Ali Libre, A. Emdadi, E. J. Kansa, M. Rahimian, and M. Shekarchi. A stabilized rbf collocation scheme for neumann type boundary value problems. *Computer Modeling In Engineering And Sciences*, 24(1):61–80, 2008. 4.3.2

[140] N. Ali Libre, A. Emdadi, E. J. Kansa, M. Shekarchi, M. Rahimian, and M. Shekarchi. Multiresolution wavelet based adaptive distributions for rbf approximation of near singular problems. *Preprint*, pages 1–27, 2009. 5.4.1, 5.7

[141] L. Ling. Multidimensional quasi-interpolation formula with dimension-splitting multiquadric basis. *Applied Mathematics and Computation*, 161:195–209, 2005. 8.3

[142] L. Ling. A univariate quasi multiquadric interpolation with better smoothness. *Computers and Mathematics with Applications*, 48:897–912, 2005. 8.3

[143] L. Ling and Y. C. Hon. Improved numerical solver for kansa's method based on affine space decomposition. *Engineering Analysis with Boundary Elements*, 29:1077–1085, 2005. 4.4, 4.4

[144] L. Ling and E. J. Kansa. Preconditioning for radial basis functions with domain decomposition methods. *Mathematical and Computer Modelling*, 40:1413–1427, 2004. 4.5.1, 4.5.1, 4.6

[145] L. Ling and E. J. Kansa. A least squares preconditioner for radial basis functions collocation methods. *Advances in Computational Mathematics*, 23:31–54, 2005. 4.5.1, 4.5.1

[146] L. Ling and R. Schaback. Stable and convergent unsymmetric meshless collocation methods. *SIAM Journal of Numerical Analysis*, 46(3):1097–1115, 2008. 4.7, 4.7

[147] L. Ling and R. Schaback. An improved subspace selection algorithm for meshless collocation methods. *To appear in International Journal For Numerical Methods In Engineering*, 2009. 4.7

[148] L. Ling and M. R. Trummer. Multiquadric collocation method with integral formulation for boundary layer problems. *Computers and Mathematics with Applications*, 48:927–941, 2004. 5.8

[149] G. R. Liu. *Mesh Free Methods: Moving Beyond the Finite Element Method*. CRC Press, 2002. (document)

[150] G. R. Liu and Y. T. Gu. *An Introduction to Meshfree Methods and Their Programming*. Springer, 2005. (document)

[151] R. A Lorentz, F. Narcowich, and J. D. Ward. Collocation discritizations of the transport equation with radial basis functions. *Applied Mathematics and Computation*, 145, 2003. 8.3

[152] S. Lowitzsch. *Interpolation and approximation employing divergence-free radial basis functions with applications*. PhD thesis, Texas A&M University, 2002. 8.3

[153] S. Lowitzsch. Error estimates for matrix-valued radial basis function interpolation. *Journal of Approximation Theory*, 137(2):238–249, 2005. 8.3

[154] W. R. Madych and S. A. Nelson. Error bounds for multiquadric interpolation. In C. Chui, L. Schumaker, and J. Ward, editors, *Approximation Therory VI*, pages 413–416. Academic Press, 1989. 2.4

[155] W. R. Madych and S. A. Nelson. Multivariate interpolation and conditionally positive definite functions ii. *Mathematics of Computation*, 4(189):211–230, 1990. 5.8

[156] W. R. Madych and S. A. Nelson. Bounds on multivariate interpolation and exponential error estimates for multiquadric interpolation. *Journal of Approximation Theory*, 70:94–114, 1992. 1

[157] W. R. Madych and S. A. Nelson. Bounds on multivariate polynomials and exponential error estimates for multiquadric interpolation. *Journal of Approximation Theory*, 70:94–114, 1992. 2.4

[158] N. Mai-Duy and T. Tran-Cong. Approximation of function and its derivatives using radial basis functions networks. *Applied Mathematical Modelling*, 27:197–220, 2003. 5.8

[159] N. Mai-Duy and T. Tran-Cong. A multidomain integrated-radial-basis-function collocation method for elliptic problems. *Applied Mathematical Modelling*, 27:197–220, 2003. 4.6, 5.8

[160] N. Mai-Duy and T. Tran-Cong. Solving high order ordinary differential equations with radial basis function networks. *Internation Journal of Numerical Methods in Engineering*, 62:824–852, 2005. 5.8

[161] N. Mai-Duy and T. Tran-Cong. Solving high order partial differential equations with indirect radial basis function networks. *Internation Journal of Numerical Methods in Engineering*, 63:1636–1654, 2006. 5.8

[162] J. C. Maihuber. On Haar's theorem concerning chebyshev approximation problems having unique solutions. *Proceedings of the American Mathematical Society*, 7:609–615, 1956. 1

[163] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1999. 5.4

[164] S. De Marchi, R. Schaback, and H. Wendland. Near-optimal data-independent point locations for radial basis function interpolation. *Advances in Computational Mathematics*, pages 1–14, 2004. 2.7

[165] C. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, first edition, 2004. 2.5

[166] C. Micchelli. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2:11–22, 1986. 1, 2.3

[167] G. J. Moridis and E. J. Kansa. The Laplace transform multiquadrics method: A highly accurate scheme for the numerical solution of linear partial differential equatins. *Journal of Applied Science and Computations*, 1:375–475, 1994. 3.2.3

[168] J. Munoz-Gomez, P. Gonzalez-Casanova, and G. Rodriguez-Gomez. Domain decomposition by radial basis functions for time dependent

partial differential equations. In *Advances in Computer Science and Technology*, pages 105–109. Proceedings of the IASTED International Conference, 2006. 4.6

[169] D. E. Myers, S. DeIaco, D. Posa, and L. DeCesare. Space-Time radial basis functions. *Computers and Mathematics with Applications*, 43:539–549, 2002. 8.3

[170] F. J. Narcowich and J. D. Ward. Scattered-data interpolation on the sphere: Error estimates and locally supported basis functions. *SIAM Journal of Mathematical Analysis*, 33:1393–1410, 2002. 8.3

[171] F. J. Narcowich, J. D. Ward, and G. B. Wright. Divergence-free rbfs on surfaces. *Journal of Fourier Analysis and Applications*, 13:643–663, 2007. 8.3

[172] K. Nguyen and D. Dabdub. Development and analysis of a non-splitting solution for three-dimensional air quality models. *Atmospheric Environment*, 37:3741–3748, 2003. 7.3

[173] A. B. Owen. Monte carlo extension of quasi monte carlo. In D. J. Medeiros, E. F. Watson, J.S. Carson, and M. S. Manivannan, editors, *Proceedings of the 1998 Winter Simulation Conference*, pages 571–577, 1998. 7.5, 7.8

[174] U. Pettersson, E. Larsson, G. Marcusson, and J. Persson. Improved radial basis function methods for multi-dimensional option pricing. *Journal of Computational and Applied Mathematics*, 222:82–93, 2008. 7.1, 8.3

[175] L. R. Pettey and R. E. Wyatt. Wave packet dynamics with adaptive grids: The moving boundary truncation method. *Chemical Physics Letters*, 424:443–448, 2006. 7.10

[176] R. Platte and T. Driscoll. Computing eigenmodes of elliptic operators using radial basis functions. *Computers and Mathematics with Applications*, 48:561–576, 2004. 8.3

[177] R. Platte and T. Driscoll. Polynomials and potential theory for gaussian radial basis function interpolation. *SIAM Journal on Numerical Analysis*, 43(2):750–766, 2005. 2.8, 3.2.2

[178] A. D. Polyanin and A. V. Manzhirov. *Handbook of Mathematics for Engineers and Scientists*. Chapin and Hall, CRC, Boco Raton, 2007. 7.9

[179] M. Powell. The theory of radial basis function approximation in 1990. In W. Light, editor, *Advances in Numerical Analysis, Vol. II: Wavelets, Subdivision Algorithms and Radial Functions*. 1990. (document)

[180] H. Power and V. Barraco. A comparison analysis between unsymmetric and symmetric radial basis function collocation methods for the numerical solution of partial differential equations. *International Journal for Numerical Methods in Engineering*, 43:551–583, 2002. 8.3

[181] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations*. Oxford, first edition, 1999. 4.6

[182] S. Rippa. An algorithm for selecting a good parameter c in radial basis function interpolation. *Advances in Computational Mathematics*, 11:193–210, 1999. 5.1

[183] S. A. Sarra. Adaptive radial basis function methods for time dependent partial differential equations. *Applied Numerical Mathematics*, 54(1):79–94, 2005. 5.7

[184] S. A. Sarra. Digital total variation filtering as postprocessing for radial basis function approximation methods. *Computers and Mathematics with Applications*, 52:1119–1130, 2006. 5.5, 5.5.2, 5.5.2

[185] S. A. Sarra. Integrated multiquadric radial basis function approximation methods. *Computers and Mathematics with Applications*, 51(8):1283–1296, 2006. 3.1, 5.8

[186] S. A. Sarra. The matlab postprocessing toolkit. *www.scottsarra.org/mpt/mpt.html*, 2008. 5.5.3

[187] S. A. Sarra. A numerical study of the accuracy and stability of symmetric and asymmetric rbf collocation methods for hyperbolic pdes. *Numerical Methods for Partial Differential Equations*, 24(2):670 – 686, 2008. 3.2.2

[188] S. A. Sarra and D. Sturgill. A random variable shape parameter strategy for radial basis function approximation methods. *Submitted to Engineering Analysis with Boundary Elements*, 2009. 5.2

[189] R. Schaback. Error estimates and condition numbers for radial basis function interpolation. *Advances in Computational Mathematics*, 3:251–264, 1995. 2.5

[190] R. Schaback. Limit problems for interpolation by analytic radial basis functions. *Journal of Computational and Applied Mathematics*, 212:127–149, 2008. 1, 5.3

[191] R. Schaback. The meshless kernel-based method of lines for solving nonlinear evolution equations. *Goettingen University preprint*, 2008. 3.2.2

[192] R. Schaback and H. Wendland. Using compactly supported radial basis functions to solve partial differential equations. In C.S. Chen, C. A. Brebbia, and D.W. Pepper, editors, *Boundary Element Technology XIII*, pages 311–324. WitPress, Southampton, Boston, 1999. 8.3

[193] J.A. Sethian and P. Smeraka. Level set methods for fluid interfaces. *Annual Review of Fluid Mechanics*, 35:341–372, 2003. 6.3, 6.3

[194] C. Shu, H. Ding, H. Q. Chen, and T. G. Wang. An upwind local RBF-DQ method for simulation of inviscid compressible flows. *Computers in applied mechanics and engineering*, 194:2001–2017, 2005. 5.6, 5.6

[195] C. Shu, H. Ding, and K. Yeo. Local radial basis function based differential quadrature method and its application to solve two-dimensional incompressible navier-stokes equations. *Computers in applied mechanics and engineering*, 192:941–954, 2003. 5.6, 5.6

[196] J. Sladek and V. Sladek. *Advances in Meshless Methods*. Tech Science Press, 2006. (document)

[197] C. Sophocleous. Linearizing mappings for certain nonlinear diffusion equations. *Journal of Physics A: Mathematical and Theoretical*, 31:6293–6307, 1998. 7.9

[198] H. L. Tolman and J. H. Alves. Numerical modeling of wind waves generated by tropical cyclones using moving grids. *Ocean Modelling*, 9:305–323, 2005. 7.11

[199] A. I. Tolstykh, M. V. Lipavskii, and D. A. Shirobokov. High-accuracy discretization methods for solid mechanics. *Archives of Mechanics*, 55:531–553, 2003. 3.1, 5.6

[200] A. I. Tolstykh and D. A. Shirobokov. On using radial basis functions in finite difference mode with applications to elasticity problems. *Computational Mechanics*, 33:68–79, 2003. 5.6

[201] C. J. Trahan and R. E. Wyatt. Radial basis function interpolation in the quantum trajectory method: optimization of the multiquadric shape parameter. *Journal of Computational Physics*, 185:27–49, 2003. 5.1

[202] C. D. Tran, D. G. Phillips, and T. Tran-Cong. Computation of dilute polymer solution flows using bcf-rbfn based method and domain decomposition technique. *Korea-Australia Rheology Journal*, 21:1–12, 2009. 7.13

[203] L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM, first edition, 1997. 2.5, 4.3, 4.3, 4.5, 4.5.1

[204] K. Volokh and O. Vilnay. Pin-pointing solution of ill-condition squar systems of linear equations. *Applied Mathematics Letters*, 13:119–124, 2000. 4.3.2

[205] J. G. Wand and G. R. Liu. On the optimal shape parameters of radial basis functions used for 2-d meshless methods. *Computers Methods in Applied Mechanics and Engineering*, 191:2611–2630, 2002. 2.10

[206] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial basis functions of minimal degree. *Advances in Computational Mathematics*, 4:389–396, 1995. 8.3

[207] H. Wendland. Local polynomial reproduction and moving least squares approximation. *IMA Journal of Numerical Analysis*, 21:285–300, 2001. 8.3

[208] H. Wendland. Moving least squares approximation on the sphere. In T. Lyche and L. L. Schumaker, editors, *Mathematical Methods in CAGD: Oslo 2000*, pages 517–526. Vanderbilt University Press, Nashville, 2001. 8.3

[209] H. Wendland. *Scattered Data Approximation*. Cambridge University Press, 2005. (document), 2

[210] J. Wertz, E. J. Kansa, and L. Ling. The role of the multiquadric shape parameter in solving elliptic partial differential equations. *Computers and Mathematics with Applications*, 51(8):1335–1348, 2006. 5.2

[211] G. Wright. *Radial Basis Function Interpolation: Numerical and Analytical Developments*. PhD thesis, University of Colorado at Boulder, 2003. 4.2

[212] G. Wright and B. Fornberg. Scattered node compact finite difference-type formulas generated from radial basis functions. *Journal of Computational Physics*, 212:99–123, 2006. 5.6

[213] Y. L. Wu and C. Shu. Development of rbf-dq for derivative approximation and its application to simulate natural convection in concentric annuli. *Computational Mechanics*, 29:477–485, 2002. 8.3

[214] J. R. Xiao and M. A. McCarthy. A local heaviside weighted meshless method for two-dimensional solids using radia basis functions. *Computational Mechanics*, 31:301–315, 2003. 2.10

[215] J. Xu and T. Belytschko. *Lecture Notes in Computational Science and Engineering*, chapter Discontinuous Radial Basis Function Approximations for Meshfree Methods, pages 231–253. Springer Berlin Heidelberg, 2005. 7.12

[216] Y. Xu and E. W. Cheney. Interpolation by periodic radial functions. *Computers and Mathematics with Applications*, pages 201–215, 1992. 8.3

[217] M. Zerroukat, H. Power, and C. S. Chen. A numerical method for heat transfer problems using collocation and radial basis functions. *International Journal for Numerical Methods in Engineering*, 42:1263–1278, 1998. 8.3

[218] Wu Zongmin. Hermite-birkhoff interpolation of scattered data by radial basis functions. *Approximation Theory and its Applications*, 8(2):1–10, 1992. 8.3