

University of Canterbury
Department of Mathematics and Statistics



Fast algorithms and preconditioning techniques
for fitting radial basis functions

A thesis presented
for the Degree of
Doctor of Philosophy in Mathematics
at the
University of Canterbury
by
Cameron T. Mouat

Supervisor: Assoc. Prof. Rick Beatson

2001

QA
281
.M924
2001

Contents

Abstract	iv
Acknowledgements	v
1 Introduction	1
1.1 Interpolation methods	2
1.1.1 Finite elements	2
1.1.2 Radial basis functions	5
1.2 Conditionally positive definite functions	7
1.3 Condition numbers	11
1.4 Choosing a basic function	17
1.5 Preconditioning techniques	21
1.5.1 Preconditioning by forming decaying functions	23
1.5.2 Powell's QR method	24
1.6 Fast matrix-vector product algorithms	26
1.6.1 Moment method	29
2 Preconditioned GMRES iteration	31
2.1 Introduction	31
2.2 The GMRES iterative method	34
2.3 Preconditioning: Good basis versus bad basis	36
2.4 Preconditioning: Approximate cardinal functions	37

2.4.1	Approximate cardinal functions based on purely local centres	39
2.4.2	Approximate cardinal functions based on local centres and special points	41
2.4.3	Decay element approximate cardinal functions with r^{-3} growth at infinity	42
2.5	Numerical Results	46
3	Fast Kriging	54
3.1	Introduction	54
3.2	Surface fitting by universal Kriging	57
3.3	A fast fitting method for large N	61
3.4	Numerical Results	65
3.5	Prediction errors	66
3.6	Geophysical application	67
3.7	Discussion	69
4	On the boundary over distance preconditioner	73
4.1	Introduction	73
4.2	A preconditioning method	75
4.3	Properties of the matrix Q	81
4.4	Scaleability	84
4.5	Decay	87
4.6	Numerical results	97
4.7	Preconditioning in \mathcal{R}^3	101
4.8	Roundoff error and fast computation of the action of the precondi- tioned matrix	102
5	An algebraic multigrid algorithm	108
5.1	Fine level approximation	109
5.2	Coarse grid approximation	110

5.3	Numerical results	112
6	RBF collocation	116
6.1	Introduction	116
6.2	RBF collocation	117
6.3	Collocation with Matern basic functions	121
6.4	Solving the collocation system for large N	125
6.4.1	Domain decomposition	125
6.4.2	Approximate cardinal functions	128
6.5	Numerical results	131
	Bibliography	134

Abstract

Radial basis functions are excellent interpolators for scattered data in \mathcal{R}^d . Previously the use of RBFs had been restricted to small or medium sized data sets due to the high computational cost of solving the interpolation equations when using global basic functions. The construction of fast multipole methods, which reduce the cost of finding a matrix-vector product to $\mathcal{O}(N \log N)$ or $\mathcal{O}(N)$ operations, has created the opportunity to dramatically reduce the cost of solving RBF equations. This thesis presents preconditioners which in conjunction with matrix iterative methods reduce the cost of solving these systems from $\mathcal{O}(N^3)$ operations to $\mathcal{O}(N \log N)$ operations.

The usual formulation of the radial basis function interpolation equations are generally badly conditioned for large N . Thus the accuracy of the solution is less certain. However, it is not the problem that is badly conditioned but instead the basis built from the Φ functions. Preconditioners in this thesis improve the conditioning of the system by converting to a better basis.

Acknowledgements

Firstly, I would like to thank my supervisor Rick Beatson for his help and guidance over the course of writing this thesis. His expertise has been invaluable.

I would like to acknowledge the support of the New Zealand Math Society and the Department of Mathematics and Statistics at the University of Canterbury for financial support allowing me to attend a conference in Canberra. Thanks also to the University of Canterbury for financial support in the form of a Doctoral Scholarship and to Applied Research Associates NZ Ltd for funding through subcontract DRF601.

There are many people within the Department of Mathematics and Statistics that have played a part in my education and to this thesis. Thanks to all the lecturers and tutors for teaching me, to the technical staff for solving my numerous problems, and to the important people that brewed the coffee for morning and afternoon tea. Also, the other postgraduate students, past and present, who have helped make the experience an enjoyable one. In particular Ben Allen, Jon Cherrie, Chris Hann, Andy McKenzie, Paul Shorten, Tim McLennen, Tim Evans and Tim Mitchell.

Last but not least, I would like to say a big thank you to my family for always being supportive and to my friends for providing me with excuses to not do work.

Chapter 1

Introduction

This dissertation considers radial basis function (RBF) interpolation and other related problems. Solving these problems by traditional techniques involves the solution of a full matrix system which is prohibitive when the number of interpolation nodes is large. The fast algorithms and preconditioning techniques presented here greatly increase the size of problem that can be tackled with RBF techniques.

This introduction presents selected pieces of the known theory related to the work in the later chapters. Section 1.1 discusses two common interpolation methods and the advantages and disadvantages of both, Section 1.2 gives existence and uniqueness conditions for radial basis function interpolants, and Section 1.3 considers bounds on the condition numbers of the interpolation matrix. Some examples of how to choose a basic function are given in Section 1.4 and examples of two preconditioning techniques are given in Section 1.5. Finally, Section 1.6, presents a brief overview of fast matrix-vector algorithms which are essential in any fast technique for fitting radial basis functions. Throughout this thesis all matrix and vector norms are l^2 norms unless specified otherwise. We also employ the notation $\Phi(x)$ for $\phi(\|x\|)$.

1.1 Interpolation methods

Consider interpolation by a function, $s : \mathcal{R}^d \rightarrow \mathcal{R}$, to data values $\{f_i\}$, given at points $X = \{x_1, \dots, x_N\} \subset \mathcal{R}^d$. The interpolation conditions are

$$s(x_i) = f_i, \quad 1 \leq i \leq N, \quad (1.1)$$

where f_i is a known value corresponding to spatial location x_i . The elements of X will be referred to interchangeably as centres of the RBF or nodes of interpolation. This interpolation problem can be solved using one of a number of methods with varying degrees of success [39]. Often an interpolation method will be limited in its application if the centres, X , have to be distributed according to specific requirements. For example, interpolation by tensor products requires the centres to be on a grid. This section briefly reviews two interpolation techniques that require at worst minor restrictions on the geometry of X . The first, finite elements, is classed as a local method because the value of the interpolant at x , $s(x)$, usually only depends on the data at points near x . The second, radial basis functions, can be global or local depending on the support of the basic function.

1.1.1 Finite elements

Finite element methods involve triangulating the convex hull of the interpolation nodes. Let T be such a set of triangles. An interpolant is found on each triangle, so that s is a piecewise function. The template for interpolation on a triangle is referred to as a macroelement. Although obtaining a suitable triangulation is important the discussion here is restricted to macroelements.

If $X \subset \mathcal{R}^2$ then write $T_{ijk} \in T$ for the triangle with vertices at x_i, x_j, x_k . Then for $x \in T_{ijk}$ the interpolant takes the form

$$s(x) = p_{ijk}(x),$$

where p_{ijk} is a (piecewise) polynomial. Akima [1] gives a macroelement where p_{ijk} is a quintic polynomial on each triangle and is determined by partial derivatives at the

vertices and normal derivatives on the edges. Such a construction renders a piecewise surface that is C^1 . However, Franke [39] noted that this method sometimes performs poorly when the derivatives are not estimated accurately and that to find accurate derivative estimates results in a “considerable time penalty in the preprocessing phase”.

A more common approach is to refine the triangulation by dividing each triangle into subtriangles. This section considers macroelements given by Powell and Sabin [70]. Each triangle is divided into either six or twelve subtriangles in the manner of Figure 1.1(a) and 1.1(b) respectively. Define \mathcal{E}_{ijk} as the edges of the subtriangles in T_{ijk} that are not part of an edge of T_{ijk} . Theorem 1.1.1 below is needed before we discuss how to determine p_{ijk} . For a more general form of Theorem 1.1.1 see [24, 85].

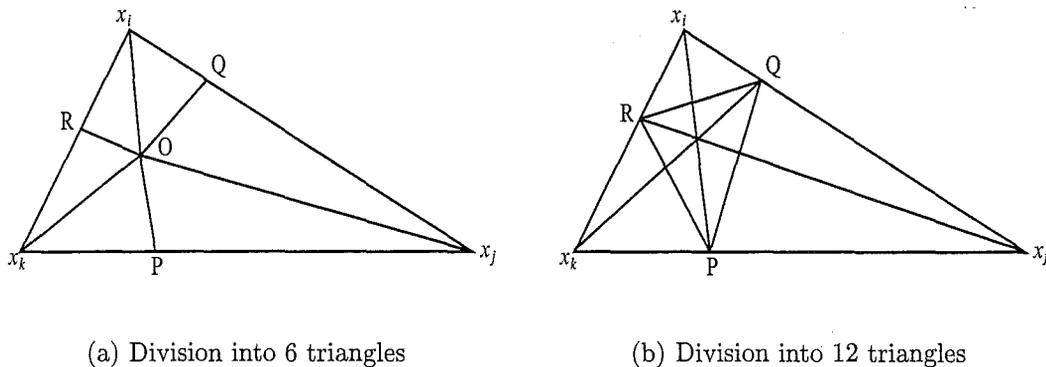


Figure 1.1: Refinement of triangles in the Powell-Sabin finite element.

Theorem 1.1.1. *Let $\Psi(x, y)$ be a piecewise function*

$$\Psi(x, y) = \begin{cases} q_1(x, y), & lx + my + n \geq 0, \\ q_2(x, y), & lx + my + n < 0. \end{cases} \quad (1.2)$$

where q_1 and q_2 are quadratics and let $L_1 = \{(x, y) : lx + my + n = 0\}$. Then $\Psi(x, y)$ is C^1 across L_1 if and only if

$$q_2(x, y) = q_1(x, y) + \lambda(lx + my + n)^2, \quad (1.3)$$

holds for some value λ .

Proof. Let $\hat{q}_2 = q_2 - q_1$. Factor \hat{q}_2 into the form

$$\hat{q}_2(x, y) = \lambda(lx + my + n)^2 + \beta(lx + my + n) + r(x, y), \quad (1.4)$$

where $r(x, y)$ cannot be factored by $lx + my + n$. If Ψ is continuous on L_1 then $\hat{q}_2|_{L_1} = 0$. This implies, for all $(x, y) \in L_1$,

$$\lambda(lx + my + n)^2 + \beta(lx + my + n) + r(x, y) = 0, \quad (1.5)$$

and therefore $r(x, y) = 0$, $(x, y) \in L_1$. So either r is identically zero or $(lx + my + n)$ is a factor of r . The latter contradicts our assumptions so it must be that $r = 0$. Furthermore the first derivatives of Ψ will be continuous only if $\nabla \hat{q}_2|_{L_1} = 0$. This leads to $\beta = 0$ in equation (1.4). \square

Consider a triangle $T_{ijk} \in T$ refined as in Figure 1.1(a). Let $s = \{1, 2, 3, 4, 5, 6, 1\}$, and the subtriangles of T_{ijk} be t_1, \dots, t_6 such that there is an edge in \mathcal{E}_{ijk} between t_{s_i} and $t_{s_{i+1}}$, $i = 1, \dots, 6$. Let q_i be the quadratic on t_i . From Theorem 1.1.1, $q_{s_{i+1}}$ will then be of the form,

$$q_{s_{i+1}}(x, y) = q_{s_i}(x, y) + \lambda_i(l_i x + m_i y + n_i)^2,$$

where λ_i is unknown and $l_i x + m_i y + n_i$ is the line which contains the edge between t_{s_i} and $t_{s_{i+1}}$. The unknown coefficients of q_1 plus $\lambda_1, \dots, \lambda_6$ determine p_{ijk} .

Interpolating to the function values and first order partial derivatives at x_i , x_j and x_k takes away nine of these degrees of freedom. Let the vertex O in Figure 1.1(a) have coordinates (x_0, y_0) then $n_i = -(l_i x_0 + m_i y_0)$. Noticing that the quadratic q_1 is determined twice gives

$$q_1 = q_1 + \sum_{i=1}^6 \lambda_i \left(l_i(x - x_0) + m_i(y - y_0) \right)^2,$$

and thus the final three conditions

$$\sum_{i=1}^6 \lambda_i l_i^2 = 0, \quad \sum_{i=1}^6 \lambda_i m_i^2 = 0, \quad \sum_{i=1}^6 \lambda_i l_i m_i = 0.$$

If O is the circumcentre of the triangle in Figure 1.1(a) and R, P, Q are midpoints of their respective edges, then, the normal derivative along an edge of T_{ijk} is linear and determined by the data at the endpoints. For example, the normal derivative along $\overrightarrow{x_j x_k}$ would be determined by the data at x_j and x_k . Thus if the entire triangulation was refined as in Figure 1.1(a) the surface would be C^1 across all triangles. It should be noted though that the circumcentre will be outside of the triangle if there is an interior angle $> 90^\circ$. Powell and Sabin then propose, for such triangles, to refine them as in Figure 1.1(b). Again quadratics are found on each subtriangle. Using these two types of macroelements appropriately on the triangulation renders a surface that is C^1 .

A disadvantage of finite element methods is that the interpolant depends on the triangulation. Often triangles will be long and thin near the edge of the convex hull and as Franke noted this can result in surfaces that look discontinuous. Evaluating at a point not in X also involves location of the appropriate triangle. For points in general position this requires more than $\mathcal{O}(1)$ operations. One interesting method of locating the relevant triangle is the walking triangle algorithm of Lawson [54, page 171]. An advantage of many finite element methods is that if an additional interpolation point is added then it is easy to update the surface with little additional cost. Often an interpolant is found using a global method like radial basis functions which is then evaluated on a suitably fine grid so that a triangulation of this fine grid will give regular triangles.

1.1.2 Radial basis functions

In view of the Mairhuber counter-example a fixed set of functions cannot be used for global interpolation at N centres if $d \geq 2$. Instead functions of the form

$$s(\cdot) = \sum_{i=1}^N \lambda_i \Phi(\cdot - x_i), \quad (1.6)$$

are used. The Mairhuber counter-example does not apply to such interpolants as the interpolation space, $\text{span}\{\Phi(\cdot - x_i) : 1 \leq i \leq N\}$, varies with the nodes of

interpolation. In (1.6), $\Phi : \mathcal{R}^d \rightarrow \mathcal{R}$ will be called the basic function and $\lambda = [\lambda_1, \dots, \lambda_N]^T$ is the vector of coefficients to be determined. If Φ is a radial function, i.e. $\Phi(x) = \Phi(y)$ for all x, y such that $\|x\| = \|y\|$, then (1.6) is called a radial basis function. These functions are often supplemented by a low degree polynomial, $q \in \pi_{m-1}^d$, where π_{m-1}^d is the space of $(m-1)^{\text{th}}$ degree polynomials in d variables, to obtain

$$s(\cdot) = q(\cdot) + \sum_{i=1}^N \lambda_i \Phi(\cdot - x_i). \quad (1.7)$$

The problem of finding an interpolant of the form (1.6) has a unique solution if Φ , X and λ satisfy certain conditions, see Section 1.2. The system,

$$\begin{bmatrix} A_\Phi & P \\ P^T & O \end{bmatrix} \begin{bmatrix} \lambda \\ a \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}, \quad (1.8)$$

is solved for $[\lambda^T \ a^T]^T$ to ensure s satisfies the interpolation conditions (1.1). In (1.8), $(A_\Phi)_{ij} = \Phi(x_i - x_j)$, and $P_{ij} = p_j(x_i)$, $j = 1, \dots, \dim(\pi_{m-1}^d)$, where $\{p_j\}$ is a basis for π_{m-1}^d . The vector a is a vector of coefficients with respect to this basis. The equations

$$\sum_{i=1}^N \lambda_i p_j(x_i) = 0, \quad j = 1, \dots, \dim(\pi_{m-1}^d), \quad (1.9)$$

ensure that the growth of s is controlled at infinity. For example, if $\Phi(\cdot) = \sqrt{|\cdot|^2 + c^2}$, $m = 1$ and $s(x)$ is of the form (1.7), then $s(x)$ is $\mathcal{O}(1)$ as $|x| \rightarrow \infty$. These ‘‘side conditions’’ (1.9) also ensure the system is invertible by taking away the extra degrees of freedom that were added by appending a polynomial.

One advantage of RBFs is that they result in smooth interpolants which are excellent approximations [38, 39] to the unknown surface. Unfortunately for globally supported basic functions traditional techniques require $\mathcal{O}(N^3)$ operations and $\mathcal{O}(N^2)$ storage to solve the system (1.8). Evaluation at a single extra point is also costly requiring $\mathcal{O}(N)$ operations. Coupled with the poor conditioning of the interpolation matrix this can result in the system being unsolvable for large N . This thesis studies techniques for solving (1.8) which reduce both the floating point

operations and storage requirements by orders of magnitude. Furthermore, preconditioning strategies will be discussed which greatly improve the conditioning of the interpolation equations.

1.2 Conditionally positive definite functions

The system of interpolation equations (1.8) has a unique solution provided the interpolation matrix

$$\begin{bmatrix} A_\phi & P \\ P^T & O \end{bmatrix}, \quad (1.10)$$

is nonsingular. As we show in this section this will occur whenever ϕ is strictly positive definite or strictly conditionally positive definite of order m and X satisfies certain weak conditions on its geometry.

Definition 1.2.1. *A function $\phi : \mathcal{R} \rightarrow \mathcal{R}$ which is continuous on $[0, \infty)$ is strictly conditionally positive definite of order m (SCPD m) on \mathcal{R}^d if for every set of distinct points $\{x_1, \dots, x_N\} \subset \mathcal{R}^d$*

$$\sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j \phi(\|x_i - x_j\|) > 0, \quad (1.11)$$

for all non-zero α satisfying,

$$\sum_{i=1}^N \alpha_i q(x_i) = 0, \quad (1.12)$$

for all polynomials q in π_{m-1}^d .

Note that a strictly positive definite function is SCPD0, and if a function is SCPD m , then it is also SCPD n for all $n > m$. Examples include the thin-plate spline and the negative of the multiquadric which are SCPD2 and SCPD1 respectively. In fact all covariograms used in ordinary Kriging are SCPD1. The table below lists the order of strict conditional positive definiteness of various basic functions. Throughout the table the constant c is positive and the strict conditional positive definiteness is on \mathcal{R}^d for every positive integer d .

linear	$\phi(r) = -r$	SCPD1
power	$\phi(r) = -r^\beta, \quad \beta \in (0, 2)$	SCPD1
thin-plate spline	$\phi(r) = r^2 \log r$	SCPD2
negative multiquadric	$\phi(r) = -\sqrt{r^2 + c^2}$	SCPD1
inverse multiquadric	$\phi(r) = 1/\sqrt{r^2 + c^2}$	SCPD0
Matern	$\phi(r) = r^\nu K_\nu(r), \quad \nu > 0$	SCPD0
exponential	$\phi(r) = \exp(-cr)$	SCPD0
Gaussian	$\phi(r) = \exp(-cr^2)$	SCPD0

In later chapters we consider fast methods for fitting RBF interpolants built upon most of these basic functions.

Theorem 1.2.2. *If $X = \{x_1, \dots, x_N\} \subset \mathcal{R}^d$ is unisolvent for π_{m-1}^d and ϕ is SCPD m then there exists a unique RBF interpolant of the form (1.7).*

Proof. It suffices to show that the system (1.8) has a unique solution. If X is unisolvent for π_{m-1}^d then P has rank $p_{\text{dim}} := \dim(\pi_{m-1}^d)$. Because λ is in the space P^\perp write $\lambda = Q\mu$, where Q is an $N \times (N - p_{\text{dim}})$ matrix with columns that span P^\perp . Then the first part of (1.8) can be written as $A_\phi Q\mu + Pa = f$. Premultiplying by Q^T gives

$$Q^T A_\phi Q \mu = Q^T f. \quad (1.13)$$

The matrix $Q^T A_\phi Q$ is positive definite as for any non-zero vector $y \in \mathcal{R}^{N-p_{\text{dim}}}$, $\hat{y} = Qy$ is non-zero and in P^\perp and

$$y^T Q^T A_\phi Q y = \hat{y}^T A_\phi \hat{y} > 0,$$

since ϕ is SCPD m . Consequently μ and thus $\lambda = Q\mu$ are uniquely determined. Equation (1.13) rearranges to $Q^T(f - A_\phi \lambda) = 0$ which shows $f - A_\phi \lambda$ is in the column space of P . Therefore the coefficients a are uniquely determined by the equations $Pa = f - A_\phi \lambda$. \square

The rest of this section is devoted to giving conditions for a basic function to be SCPD m . Much of the treatment follows Chapters 14 and 15 of Cheney and Light [22].

Definition 1.2.3. *A function f is said to be completely monotone on $[0, \infty)$ if*

- $f \in C[0, \infty)$
- $f \in C^\infty(0, \infty)$
- $(-1)^k f^{(k)}(t) \geq 0$ for $t > 0$ and $k = 0, 1, \dots$

One example of a completely monotone function is $f(\cdot) = (1 + \cdot)^{-1/2}$. It is easy to verify that this function satisfies the criteria in Definition 1.2.3. We will see in the Schoenberg and Michelli interpolation theorems below that completely monotone functions are important in characterizing classes of functions that are SCPD0 and SCPD1 respectively. A characterization of a completely monotone function is given by the Bernstein-Widder Theorem [90].

Lemma 1.2.4 (Bernstein-Widder). *A function $f : [0, \infty) \rightarrow [0, \infty)$ is completely monotone if and only if there is a nondecreasing bounded function μ such that*

$$f(t) = \int_0^\infty \exp(-st) d\mu(s).$$

The following result is well known and can be proved via Bochner's Theorem (see for example [78, 22]) or see Powell [67, Corollary 3.2].

Lemma 1.2.5. *The Gaussian $\Phi(\cdot) = \exp(-c\|\cdot\|^2)$, $c > 0$ is strictly positive definite on \mathcal{R}^d for every positive integer d .*

Theorem 1.2.6 (Schoenberg [76]). *If f is completely monotone but not constant on $[0, \infty)$, then the function $\phi(\cdot) = f(\cdot^2)$ is a strictly positive definite function.*

Proof. Let $\alpha \in \mathcal{R}^N$, $\alpha \neq 0$ and x_1, \dots, x_N be distinct points in \mathcal{R}^d . If f is completely monotone, then from Lemma 1.2.4, we obtain

$$\begin{aligned} \sum_{i,j=1}^N \alpha_i \alpha_j \phi(\|x_i - x_j\|) &= \sum_{i,j=1}^N \alpha_i \alpha_j \int_0^\infty \exp(-s\|x_i - x_j\|^2) d\mu(s), \\ &= \int_0^\infty \sum_{i,j=1}^N \alpha_i \alpha_j \exp(-s\|x_i - x_j\|^2) d\mu(s), \end{aligned} \quad (1.14)$$

where bringing the summation inside the integral is valid because the integral is finite. Because f is not constant, $d\mu(s)$ is not concentrated at zero, and thus μ is not concentrated at zero. From Lemma 1.2.5 the integrand is positive, for $s > 0$. From Lemma 1.2.4, $d\mu$ is positive and thus the integral is positive. \square

The Schoenberg interpolation theorem can be applied to show that the inverse multiquadric and the Matern class of functions are positive definite. For the inverse multiquadric we have $f(\cdot) = (c^2 + \cdot)^{-1/2}$. f is completely monotone and thus from Theorem 1.2.6 the inverse multiquadric is strictly positive definite.

The Michelli interpolation theorem requires the following representation of a SCPD1 function. This can be found in [22].

Lemma 1.2.7. *Let $f : [0, \infty) \rightarrow [0, \infty)$ be such that $f \in C[0, \infty)$ and $f \in C^\infty(0, \infty)$ and f' is completely monotone but not constant on $(0, \infty)$. Then f has the representation*

$$f(t) = f(0) + \int_0^\infty \frac{1 - \exp(-st)}{s} d\mu(s),$$

where μ is a nondecreasing measure.

Theorem 1.2.8 (Michelli [60]). *Let $f : [0, \infty) \rightarrow [0, \infty)$. Assume f is continuous on $[0, \infty)$ and that f' is completely monotone but not constant on $(0, \infty)$. Then $\phi(\cdot) = -f(\cdot^2)$ is SCPD1.*

Proof. Applying Lemma 1.2.7 to f gives the representation,

$$f(t) = f(0) + \int_0^\infty \frac{1 - \exp(-st)}{s} d\mu(s).$$

Let $\alpha \in \mathcal{R}^N$, $\alpha \neq 0$ with $\sum \alpha_i = 0$ and x_1, \dots, x_N be distinct points in \mathcal{R}^d . Now the quadratic form becomes

$$\begin{aligned} \sum_{i,j=1}^N \alpha_i \alpha_j \phi(\|x_i - x_j\|) &= - \sum_{i,j=1}^N \alpha_i \alpha_j \left(f(0) + \int_0^\infty \frac{1 - \exp(-s\|x_i - x_j\|^2)}{s} d\mu(s) \right), \\ &= \int_0^\infty \sum_{i,j=1}^N \alpha_i \alpha_j \frac{\exp(-s\|x_i - x_j\|^2)}{s} d\mu(s), \end{aligned} \quad (1.15)$$

which is positive because the integrand is positive from again using Lemma 1.2.5. \square

Theorems 1.2.6 and 1.2.8 along with Theorem 1.2.2 show the existence and uniqueness of RBF interpolants for a large class of important basic functions. These include the inverse multiquadric, the multiquadric, and the linear basic functions. The following theorem from Michelli [60] generalizes the previous result to give sufficient conditions for ϕ to be SCPD m .

Theorem 1.2.9 (Michelli [60]). *Let $f : [0, \infty) \rightarrow [0, \infty)$. Assume f is continuous on $[0, \infty)$, and that the derivative $f^{(j)}(t)$ exists for all positive integers j and for all $t \in (0, \infty)$. If $(-1)^m f^{(m)}$ is completely monotone on $(0, \infty)$ then $\phi(\cdot) = f(\cdot^2)$ is SCPD m .*

For example, consider the thin-plate spline $\phi(r) = r^2 \log r$ and the corresponding function $f(r) = (1/2)r \log r$. The second derivative of f is $f'' = (2r)^{-1}$ which is completely monotone on $(0, \infty)$. Thus the thin-plate spline is SCPD2.

1.3 Condition numbers

The matrix of the usual formulation (1.10) of a radial basis function problem is generally badly conditioned for large values of N . Solving a linear system with a large condition number can result in a less accurate solution if a small perturbation enters the system. Such a perturbation can occur, for example, by roundoff error due to finite precision arithmetic. The well known relationship between relative

error and relative residual for the system $Ax = b$ is

$$\frac{1}{\text{cond}(A)} \frac{\|\delta b\|}{\|b\|} \leq \frac{\|\delta x\|}{\|x\|} \leq \text{cond}(A) \frac{\|\delta b\|}{\|b\|}.$$

Thus if the condition number of A is large, the relative residual $\|\delta b\|/\|b\|$ gives little information about the relative error $\|\delta x\|/\|x\|$. That is a small residual may correspond to a large error or vice versa. On the other hand, if the condition number is small then the relative residual and relative error will be of a similar order of magnitude. All condition numbers and norms in this section are for l_2 unless otherwise specified.

This section reviews some of the bounds on condition numbers given by various authors [2, 63, 64, 73] for the matrix A_ϕ in (1.8). These results generally show that upper bounds on the condition number depend on the minimum separation distance between two centres in X and get larger as this separation distance decreases. So as N increases and X is within a fixed domain the interpolation matrix is likely to become more ill-conditioned and the accuracy of the solution is less certain. Define the function $\theta_\phi(h)$ as an upper bound of the form

$$\|A_\phi^{-1}\| \leq \theta_\phi(h), \quad (1.16)$$

where the bound holds for any set of distinct nodes, $X \in \mathcal{R}^d$, with

$$h := \min_{(x_i, x_j) \in X, i \neq j} \|x_i - x_j\|.$$

In [3] it was shown that for a uniform grid of centres, with separation distance h , that a lower bound to $\|A_\phi^{-1}\|$ could also be obtained. For the linear basic function the lower bound only differed from the upper bound obtained in [2] by a multiplicative constant, showing that this upper bound is optimal up to a constant. The same technique was also applied to the multiquadric and inverse multiquadric basic functions and established that the upper bounds were close to optimal for these functions.

The following list gives upper bounds on $\|A_\phi^{-1}\|$ for some common radial functions. In the list h is the separation distance, d the dimension, and c a ϕ -specific

parameter. Note also that the values in the table are the dominant power of h as $h \rightarrow 0$ and that constants have been omitted.

Basic function ϕ		Upper bound on $\ A_\phi^{-1}\ $
linear	$\phi(r) = -r$	h^{-1}
power	$\phi(r) = -r^\beta, \quad \beta \in (0, 2)$	$h^{-\beta}$
multiquadric	$\phi(r) = \sqrt{r^2 + c^2}$	$h^{-1} \exp(12.76cd/h)$
inverse multiquadric	$\phi(r) = 1/\sqrt{r^2 + c^2}$	$h \exp(12.76cd/h)$
Matern	$\phi(r) = r^\nu K_\nu(r), \quad \nu > 0$	$h^{-2\nu}$
exponential	$\phi(r) = \exp(-cr)$	h^{-1}
Gaussian	$\phi(r) = \exp(-cr^2)$	$h^d \exp(40.71d^2/(ch^2))$

Some of the first research on condition numbers for RBF systems was by Ball [2]. He established an upper bound on the norm of A_ϕ^{-1} when ϕ is the linear basic function. Narcowich and Ward [63, 64] with a more general technique give upper bounds on the norm of A_ϕ^{-1} for various SCPD1 basic functions including the multiquadric.

The approach reviewed here for finding upper bounds on $\|A_\phi^{-1}\|$ is from Schaback [73]. Although many results given by Schaback were already known the approach is simpler and more general than the method of Narcowich and Ward. Firstly, two well known theorems which are important for studies on condition numbers.

Theorem 1.3.1. *Let $X = \{x_1, \dots, x_N\}$ be distinct points in \mathcal{R}^d . If Φ is a strictly positive definite function and*

$$\sum_{i,j=1}^N \alpha_i \alpha_j \Phi(x_i - x_j) \geq \theta_\Phi \|\alpha\|^2,$$

for all $\alpha \in \mathcal{R}^N$, then,

$$\|A_\Phi^{-1}\| \leq \theta_\Phi^{-1}.$$

The proof comes directly from the Rayleigh-Ritz theorem which can be found in, for example, [48, Theorem 4.2.2].

Theorem 1.3.2 (Ball [2]). *Let $X = \{x_1, \dots, x_N\}$ be distinct points in \mathcal{R}^d . If Φ is a strictly conditionally positive definite function of order 1 with $\Phi(0) \leq 0$ and*

$$\sum_{i,j=1}^N \alpha_i \alpha_j \Phi(x_i - x_j) \geq \theta_\Phi \|\alpha\|^2,$$

for all $\alpha \in \mathcal{R}^N$ satisfying $\sum_i \alpha_i = 0$, then,

$$\|A_\Phi^{-1}\| \leq \theta_\Phi^{-1}.$$

The proof requires the use of the Courant-Fischer theorem below which can be found in, for example, [48, Theorem 4.2.11].

Lemma 1.3.3 (Courant-Fischer). *Let A be an $N \times N$ symmetric matrix with eigenvalues $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N$, and let k be a given integer with $1 \leq k \leq N$. Then*

$$\max_{w_1, \dots, w_{k-1} \in \mathcal{R}^N} \min_{\substack{\alpha \neq 0, \alpha \in \mathcal{R}^N \\ \alpha \perp w_1, \dots, w_{k-1}}} \frac{\alpha^T A \alpha}{\alpha^T \alpha} = \lambda_k.$$

Proof of Theorem 1.3.2. Define $\mathcal{D} = \{\alpha \in \mathcal{R}^N \setminus \{0\} : \sum \alpha_i = 0\}$. Then applying Lemma 1.3.3 with $k = 2$ gives

$$\lambda_2 \geq \min_{\alpha \in \mathcal{D}} \frac{\alpha^T A_\Phi \alpha}{\|\alpha\|^2} \geq \theta_\Phi.$$

The sum of the eigenvalues equals the trace of A_Φ which implies

$$\lambda_1 = \text{trace}(A_\Phi) - \sum_{i=2}^N \lambda_i \leq N\Phi(0) - (N-1)\theta_\Phi.$$

It follows that

$$\min_{i=1, \dots, N} |\lambda_i| \geq \theta_\Phi,$$

and $\|A_\Phi^{-1}\| \leq \theta_\Phi^{-1}$. □

Given a SCPD m function Φ with Fourier transform $\hat{\phi}$, and $\alpha \in \mathcal{R}^N$ such that $\sum_i \alpha_i q(x_i) = 0$, for all $q \in \pi_{m-1}^d$, then Madych and Nelson [57] show,

$$\sum_{i,j=1}^N \alpha_i \alpha_j \Phi(x_i - x_j) = (2\pi)^{-d} \int_{\mathcal{R}^d} \hat{\phi}(\omega) \left| \sum_{i=1}^N \alpha_i \exp(ix_i^T \omega) \right|^2 d\omega. \quad (1.17)$$

Now, let $\hat{\psi}$ be the Fourier transform of a radial function Ψ such that

$$\hat{\phi} \geq \hat{\psi} \geq 0. \quad (1.18)$$

Then by substituting $\hat{\psi}$ into (1.17) and using (1.18) we obtain

$$\begin{aligned} \sum_{i,j=1}^N \alpha_i \alpha_j \Phi(x_i - x_j) &= (2\pi)^{-d} \int_{\mathcal{R}^d} \hat{\phi}(\omega) \left| \sum_i \alpha_i \exp(ix_i^T \omega) \right|^2 d\omega, \\ &\geq (2\pi)^{-d} \int_{\mathcal{R}^d} \hat{\psi}(\omega) \left| \sum_i \alpha_i \exp(ix_i^T \omega) \right|^2 d\omega, \\ &= \sum_{i,j=1}^N \alpha_i \alpha_j \Psi(x_i - x_j). \end{aligned} \quad (1.19)$$

Define B as the matrix $B_{ij} = \Psi(x_i - x_j)$. By choosing Ψ so that B is diagonally dominant a bound of the form

$$\alpha^T B \alpha \geq \theta_\Phi \|\alpha\|^2, \quad \theta_\Phi > 0,$$

can be obtained. For SCPD0 or SCPD1 basic functions this will imply $\|A_\Phi^{-1}\| \leq \theta_\Phi^{-1}$ by Theorems 1.3.1 and 1.3.2.

To construct a θ_Φ Narcowich and Ward [64] first find a function $\omega(s)$ so that

$$Q_s := \sum_{i,j=1}^N \alpha_i \alpha_j \exp(-s|x_i - x_j|^2) \geq \omega(s) |\alpha|^2,$$

where $s > 0$ and $\alpha \in \mathcal{R}^N$ is such that $\sum_i \alpha_i q(x_i) = 0$, for all $q \in \pi_{m-1}^d$. They then use the representation

$$\sum_{i,j=1}^N \alpha_i \alpha_j \Phi(x_i - x_j) = \int_0^\infty \frac{Q_s}{s^m} d\mu(s),$$

which comes from Michelli [60] and the constraints on α . Now θ_Φ is defined as

$$\theta_\Phi := \int_0^\infty \frac{\omega(s)}{s^m} d\mu(s). \quad (1.20)$$

Evaluating the integral (1.20) for a given Φ leads to the upper bound on $\|A_\Phi^{-1}\|$.

The following result, from Schaback [73, Theorem 3.1], gives another estimate of θ_Φ .

Theorem 1.3.4. *If $\hat{\phi}$ is the Fourier transform of ϕ then define,*

$$\hat{\phi}_0(r) := \inf_{\|\omega\| \leq 2r} \hat{\phi}(\omega).$$

A bound on θ_ϕ is

$$\theta_\phi \geq \frac{1}{2} \frac{\hat{\phi}_0(M)}{\Gamma(d/2 + 1)} \left(\frac{M}{4\sqrt{\pi}} \right)^d,$$

for any $M > 0$ satisfying

$$M \geq \frac{12}{h} \left(\frac{\pi \Gamma^2(d/2 + 1)}{9} \right)^{\frac{1}{d+1}},$$

or alternatively, $M \geq \frac{6.38d}{h}$.

Schaback goes on to show the bound for the multiquadric. We proceed to apply Schaback's machinery to the Matern class (see e.g. (1.26)) which have Fourier transforms given by

$$\hat{\phi}(\omega) = (c^2 + \|\omega\|^2)^{-\nu-d/2},$$

where c is a positive constant. $\hat{\phi}$ is decreasing so,

$$\hat{\phi}_0(r) = (c^2 + 4r^2)^{-\nu-d/2},$$

and therefore,

$$\theta_\phi \geq \frac{1}{2} \frac{(c^2 + 4M^2)^{-\nu-d/2}}{\Gamma(d/2 + 1)} \left(\frac{M}{4\sqrt{\pi}} \right)^d.$$

Let $C_1 = (2(4\sqrt{\pi})^d \Gamma(d/2 + 1))^{-1}$ then substituting in the bound on M and simplifying gives

$$\begin{aligned} \theta_\phi &\geq \frac{C_1 M^d}{(c + 4M^2)^{\nu+d/2}}, \\ &= \frac{C_2 h^{2\nu}}{(ch^2 + 162.8d^2)^{\nu+d/2}}, \end{aligned} \tag{1.21}$$

where $C_2 = C_1(6.38d)^d$. So for the Matern class of basic functions, as the smoothness of the basic function increases then so does the upper bound θ_ϕ .

Upper bounds on $\|A_\phi\|$ are found by Narcowich and Ward [63] and Ball [2]. Noticing that the maximum absolute row sum is bounded by $N \max |\Phi(x_i - x_j)|$ leads to the inequality,

$$\|A_\phi\| \leq N \max |\Phi(x_i - x_j)|.$$

Now each centre is surrounded by a ball of volume $C_d(2h)^d$ which contains no other centre. The sum of these volumes will be less than the total volume $C_d(D + 2h)^d$ where D is the diameter of a region of \mathcal{R}^d containing X . Therefore, $NC_d(2h)^d \leq C_d(D + 2h)^d$ and

$$N \leq \left(\frac{D + 2h}{2h} \right)^d.$$

The bound on $\|A_\phi\|$ becomes

$$\|A_\phi\| \leq \max |\Phi(x_i - x_j)| \left(\frac{D + 2h}{2h} \right)^d.$$

Combining the upper bounds on $\|A_\phi\|$ and $\|A_\phi^{-1}\|$ gives a bound on the condition number.

1.4 Choosing a basic function

The equations for finding the coefficients of an RBF or of a Kriging function in its “dual” form are identical once the basic function, Φ , is known. The heuristic behind finding this basic function is where the two methods differ. In RBF interpolation the basic function is specified by minimizing a chosen seminorm over a subspace of functions. A well known example from Duchon [29] is the thin-plate spline in \mathcal{R}^2 which minimizes

$$\int_{x=(\xi,\eta) \in \mathcal{R}^2} \left[\frac{\partial^2 s(x)}{\partial \xi^2} \right]^2 + 2 \left[\frac{\partial^2 s(x)}{\partial \xi \partial \eta} \right]^2 + \left[\frac{\partial^2 s(x)}{\partial \eta^2} \right]^2 dx, \quad (1.22)$$

over all interpolants $s : \mathcal{R}^2 \rightarrow \mathcal{R}$ that have bounded first derivatives and square integrable second derivatives. This can be thought of as minimizing the bending energy or curvature of the interpolant. Clearly if you wanted an interpolant that was as smooth as possible in the sense of (1.22) then you would fit an RBF with a thin-plate spline basic function. If, however, the aim was to fit a smooth surface that does not necessarily interpolate the data then a different criteria could be used. For example, minimizing

$$\|y - f\|_2 + \rho \int_{x=(\xi,\eta) \in \mathcal{R}^2} \left[\frac{\partial^2 s(x)}{\partial \xi^2} \right]^2 + 2 \left[\frac{\partial^2 s(x)}{\partial \xi \partial \eta} \right]^2 + \left[\frac{\partial^2 s(x)}{\partial \eta^2} \right]^2 dx, \quad (1.23)$$

where y is a vector with $y_i = s(x_i)$ and ρ sets the level of compromise between fidelity to the initial data f and smoothness. Large values of ρ correspond to a smooth surface with little resemblance to the original data at the centres. As $\rho \rightarrow 0$ the surface becomes closer to the interpolating surface. RBFs that minimize (1.23) are called smoothing splines and finding the parameter ρ is often achieved by generalized cross validation (GCV) (see for example [83]). This leads to a system

$$\begin{bmatrix} A_\Phi + \beta I & P \\ P^T & O \end{bmatrix} \begin{bmatrix} \lambda \\ a \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix},$$

to solve for λ and a where β is proportional to ρ . The smoothing spline does not interpolate the data but often reflects the actual underlying surface if the data is noisy. Thin-plate spline basic functions have been shown to be very accurate in medical imaging [21], surface reconstruction [20], as well as other applications. The inverse multiquadric and multiquadric have been used with success in geophysics and for solving elliptic PDEs.

In the Kriging context the basic function is chosen to minimize

$$E\left((z(x) - s(x))^2\right), \quad (1.24)$$

at a point $x \notin X$. In (1.24), $z(x)$ is the actual surface value, $s(x)$ is the Kriged approximation, and $E(\cdot)$ is the expected value. Minimizing (1.24) ensures that the surface is as accurate as possible at the point x . Often this can lead to the surface not being as aesthetically pleasing as a surface produced using an RBF interpolant or smoothing spline.

Some classes of basic functions used in Kriging contain a smoothness parameter which sets the smoothness of the fitted surface. One such class of functions is the Matern class [42, 58, 71]. These basic functions are given by

$$\Phi_\nu(\cdot) = \frac{2^{1-\nu}\alpha_1}{\Gamma(\nu)}(\alpha_2|\cdot|)^\nu K_\nu(\alpha_2|\cdot|), \quad (1.25)$$

where K_ν is a modified Bessel function of order $\nu > 0$, and $\alpha_1, \alpha_2 > 0$ are constants to be determined. If n is a positive integer then using equation 12 on page 80 of [86]

gives

$$\phi_{n+1/2}(r) = \frac{\exp(-\alpha_2 r) \alpha_1 (\alpha_2 r)^n}{(2n-1)!!} \sum_{k=0}^n \frac{(n+k)!}{k!(n-k)!(2\alpha_2 r)^k}.$$

Listed below are examples of Matern basic functions for specific values of ν and $\alpha_1 = \alpha_2 = 1$.

$$\begin{aligned} \nu = 1/2, \quad \phi(r) &= \exp(-r), \\ \nu = 3/2, \quad \phi(r) &= (1+r) \exp(-r), \\ \nu = 5/2, \quad \phi(r) &= (1+r+r^2/3) \exp(-r), \\ \nu = 7/2, \quad \phi(r) &= (1+r+2r^2/5+r^3/15) \exp(-r). \end{aligned} \tag{1.26}$$

An advantage of the Matern class of functions is that it allows the “degree of smoothness to be estimated from the data rather than restricted a priori” [78]. If the data is determined to be smooth a higher value of ν would be appropriate. Likewise if the data is noisy a lower value of ν . As is stated in Remark 1.4.1 below higher values of ν correspond to smoother basic functions and therefore smoother interpolants.

Remark 1.4.1. *If Φ_ν is a member of the Matern family given in (1.25) then Φ_ν is $2m$ -times differentiable everywhere in \mathcal{R}^d for all $m < \nu$.*

For example, consider the Matern function with $\nu = 3/2$ on \mathcal{R} . This is clearly infinitely differentiable for $x \neq 0$. For $x = 0$, we use a Taylor expansion of $\exp(-x)$ and obtain

$$\begin{aligned} \phi_{3/2}(|x|) &= (1+x)(1-x+x^2/2-x^3/6+\mathcal{O}(x^4)), \quad x \rightarrow 0^+, \\ &= 1-x^2/2+x^3/3+\mathcal{O}(x^4), \quad x \rightarrow 0^+, \end{aligned}$$

and

$$\begin{aligned} \phi_{3/2}(|x|) &= (1-x)(1+x+x^2/2+x^3/6+\mathcal{O}(x^4)), \quad x \rightarrow 0^-, \\ &= 1-x^2/2-x^3/3+\mathcal{O}(x^4), \quad x \rightarrow 0^-. \end{aligned}$$

Clearly $\phi_{3/2}(|\cdot|)$ is only twice differentiable at zero. To lift this result into d dimensions firstly consider the $2m$ times differentiable function

$$\phi(r) = a_0 + a_2 r^2 + \dots + a_{2m} r^{2m} + \mathcal{O}(r^{2m+1}), \quad r \rightarrow 0.$$

Then for x in \mathcal{R}^d

$$\begin{aligned}\Phi(x) &= \phi(\|x\|) = a_0 + a_2\|x\|^2 + \dots + a_{2m}\|x\|^{2m} + \mathcal{O}(x^{2m+1}), \quad x \rightarrow 0, \\ &= p_{2m}(x) + \mathcal{O}(x^{2m+1}),\end{aligned}$$

which implies Φ is $2m$ times differentiable at zero and p_{2m} is the Maclaurin polynomial. From this it is clear that $\Phi_{3/2} \in C^2(\mathcal{R}^d)$.

For $X \subset \Omega \subset \mathcal{R}^d$, Schaback and others [92, 73, 74] show, for any f in the “native space” of Φ , error bounds of the form

$$|s(x) - f(x)| \leq Ch^\gamma \|f\|_\Phi, \quad x \in \Omega, \quad (1.27)$$

where C is a Φ specific constant and

$$h := \sup_{x \in \Omega} \min_{1 \leq i \leq N} \|x - x_i\|.$$

The highest value of γ for which such a bound holds is called the approximation order. The native space, F_Φ , consists of all functions $f : \mathcal{R}^d \rightarrow \mathcal{R}$ with Fourier transform \hat{f} that satisfies

$$\|f\|_\Phi := \int_{\mathcal{R}^d} \frac{|\hat{f}(\omega)|^2}{\hat{\Phi}(\omega)} d\omega < \infty. \quad (1.28)$$

In [92] it was shown that the multiquadric and Gaussian basic function have unbounded local approximation orders. For Matern functions the approximation order is ν [73]. It can be shown (see for example [88]) that the native space of a Matern basic function is the Sobolev space $H^{\nu+d/2}(\mathcal{R}^d)$. Thus if f is not sufficiently smooth then the approximation order ν is not applicable.

The following examples illustrate that the parameter ν needs to be carefully chosen to obtain the most accurate RBF interpolants. Figure 1.2 below shows RBF interpolation surfaces and error surfaces where the basic function is Φ_ν , $\nu = 1/2, 3/2, 5/2$. The centres are two hundred random points in $[0, 1]^2$ (see Figure 1.2(b)), and the right hand side is of the form $f_i = F_0(x_i)$. $F_0(x)$, $x = (\xi, \eta)$, is the C^0 function with essentially bounded first partials,

$$F_0(x) = (\xi + \eta - 1)_+^1.$$

The relative 2-norm errors between the fitted surface and F_0 evaluated over 1600 points were 0.546, 0.134, 0.189, and 0.213 for $\nu = 1/2, 3/2, 5/2$ and the thin-plate spline respectively. In this example the best RBF interpolant from the Matern class was with $\nu = 3/2$ which corresponds to a C^2 basic function.

The same experiment except with the function

$$F_1(x) = \frac{3}{4} \exp\left(-\frac{(9\xi - 2)^2 + (9\eta - 2)^2}{4}\right) + \frac{3}{4} \exp\left(-\frac{(9\xi + 1)^2}{49} - \frac{9\eta + 1}{10}\right) \\ + \frac{1}{2} \exp\left(-\frac{(9\xi - 7)^2 + (9\eta - 3)^2}{4}\right) - \frac{1}{5} \exp(-(9\xi - 4)^2 - (9\eta - 7)^2), \quad (1.29)$$

of [31] lead to the errors 0.0195, 0.0067, 0.0037 and 0.0133, for $\nu = 1/2, 3/2, 5/2$ and the thin-plate spline respectively. This time the C^4 basic function was more accurate. This is not unexpected because the C^4 basic function has a higher approximation order for the smoother function F_1 . The observation that larger values of ν are more appropriate for smoother data is the heuristic behind using the Matern class for finding numerical solutions to PDEs in Chapter 6.

In the Kriging setting it is possible to use maximum likelihood techniques, as given by [82], as a way of estimating ν so that (1.24) is minimized.

1.5 Preconditioning techniques

Section 1.3 indicates that the usual system (1.10) of interpolation equations is often badly conditioned when the number of data points is large. Also solving the interpolation system without any customized method will require $\mathcal{O}(N^3)$ operations. If a matrix iterative method, such as GMRES or CG, is used then the major work of a single iteration is a matrix vector product “requiring” $\mathcal{O}(N^2)$ operations. However, convergence of these methods depends on the relative clustering of the eigenvalue spectrum of the matrix (see Chapter 2). As we’ll see in later chapters the eigenvalue spectrum for the usual interpolation matrix can be spread through many orders of magnitude and any iterative method will require a large number of iterations to

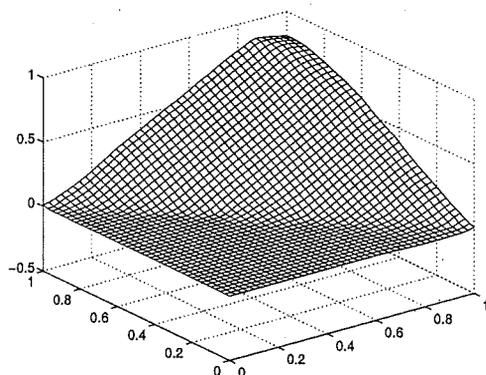
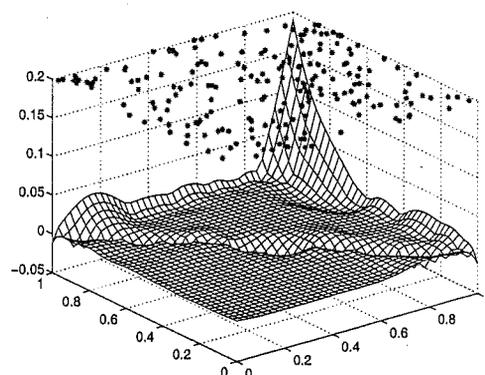
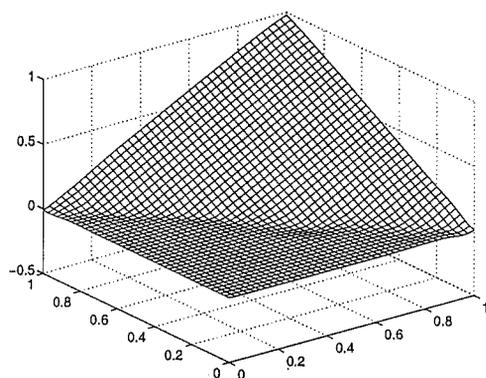
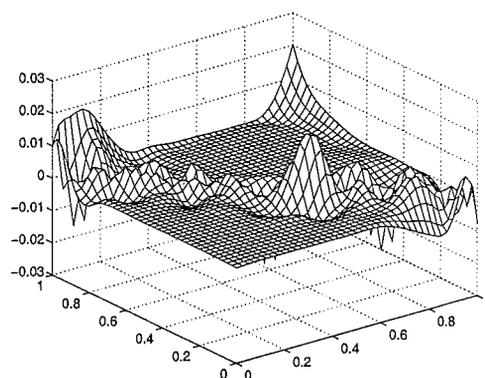
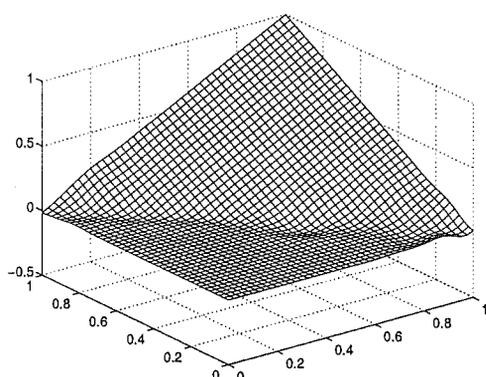
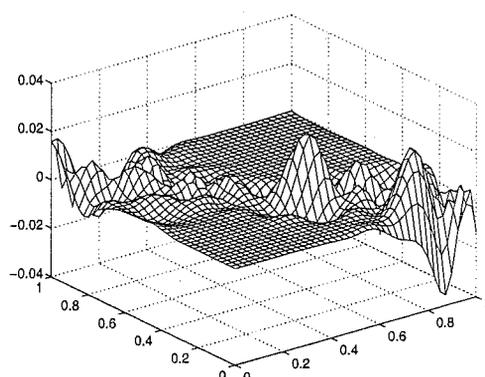
(a) Interpolation with $\nu = 1/2$.(b) Error with $\nu = 1/2$.(c) Interpolation with $\nu = 3/2$.(d) Error with $\nu = 3/2$.(e) Interpolation with $\nu = 5/2$.(f) Error with $\nu = 5/2$.

Figure 1.2: Interpolation to function F_0 using various Φ_ν functions from the Matern family. See the text for more information.

converge. Fast matrix-vector product algorithms can reduce the computation to $\mathcal{O}(N \log N)$ per iteration but for large N this is still prohibitive. Loss of accuracy will also occur through bad conditioning.

The main emphasis of this thesis is to study preconditioning techniques which allow for fast fitting of RBFs and Kriging surfaces when N is large. This section gives an overview of two preconditioning techniques which have been used. The first is designed to form a well conditioned system and the second designed to form a positive definite system to which conjugate gradients can be applied.

1.5.1 Preconditioning by forming decaying functions

In [30, 31] the authors consider forming new basis functions so that the preconditioned matrix is well conditioned with clustered eigenvalues. Consider a function $\psi = \Delta^k \phi$, $k \geq 1$ that has the properties

$$\psi(r) \rightarrow 0 \text{ as } r \rightarrow \infty, \quad (1.30)$$

and

$$\psi(r) \gg 1 \text{ as } r \rightarrow 0. \quad (1.31)$$

The matrix given by $A_{ij} = \psi(\|x_i - x_j\|)$ will be approximately diagonally dominant and is likely to be well conditioned (of course this obviously depends on how fast the function goes to zero as r grows).

The preconditioning strategy is to form new basis elements by using discretised approximations to $\Delta^k \phi$. When the centres are on a regular grid, and $k = 1$, a suitable discretised Laplacian is the 5-point star. Then for an interior centre x_i the ψ element is,

$$\psi_i(\cdot) = \phi(x_{i,j+1} - \cdot) + \phi(x_{i,j-1} - \cdot) + \phi(x_{i+1,j} - \cdot) + \phi(x_{i-1,j} - \cdot) - 4\phi(x_{i,j} - \cdot), \quad (1.32)$$

where $x_{ij} = x_{00} + i(h, 0) + j(0, h)$, h being the separation distance of the regular grid. For the multiquadric and linear basic functions these ψ elements are bell shaped, see Figure 1.3, and have the properties (1.30) and (1.31).

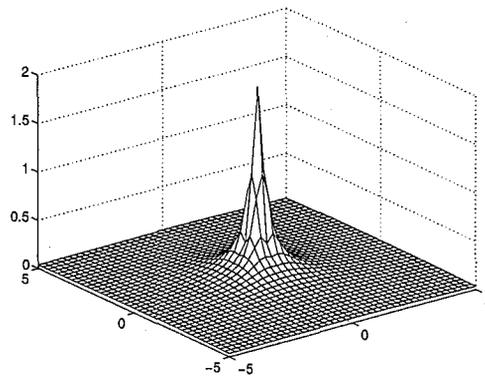


Figure 1.3: A bell shaped new basis function of the form (1.32), with $x_i = (0, 0)$ and $h = 0.5$, and ϕ is the multiquadric.

Unfortunately the authors only trialed small data sets and regular data arrangements experimentally. Using the multiquadric basic function and centres on an 11×11 grid the condition number of the interpolation matrix decreased from 7274 to 126 for the regular system and the preconditioned system respectively. Similarly for the thin-plate spline the condition numbers were 6764 and 5.1 respectively. Although these problems are small the preconditioning strategy has a marked effect on the conditioning of the interpolation matrix. We can see that even on these small regular data sets it is not the problem of RBF interpolation that is badly conditioned, it is only its formulation in terms of the “basis” of ϕ functions. Choosing a “better” basis can dramatically improve conditioning. This idea is discussed more in chapters to follow.

1.5.2 Powell’s QR method

Since fast matrix-vector product algorithms have been known there has been much interest in using matrix iterative methods for solving preconditioned RBF systems. Using a fast multiply even a method requiring N iterations will solve interpolation problems in $\mathcal{O}(N^2)$ or $\mathcal{O}(N^2 \log N)$ operations rather than $\mathcal{O}(N^3)$. One iterative method is conjugate gradients (CG) which requires the preconditioned matrix to be

symmetric positive definite. An early application of such methods to RBF interpolation was given by Powell [69]. Powell converts the usual interpolation system (1.8) into a symmetric positive definite system by finding an $N \times (N - p_{\text{dim}})$ matrix Q which spans P^\perp as in Theorem 1.2.2. Q is formed by constructing Householder matrices, H_j , $j = 1, \dots, p_{\text{dim}}$, so that

$$P_j = H_j^T \dots H_1^T P, \quad j = 1, \dots, p_{\text{dim}},$$

is zero below the diagonal in the first j columns. Each Householder matrix is of the form

$$H_j = I - \frac{2u_j u_j^T}{u_j^T u_j},$$

where u_j is found so that H_j zeros P_{j-1} below the diagonal in the j th column and doesn't disturb the zeros below the diagonal in the first $j - 1$ columns. Clearly the matrix given by $\Omega = H_1 H_2 \dots H_{p_{\text{dim}}}$ will have the last $N - p_{\text{dim}}$ columns orthogonal to P . Let Q be the matrix given by these columns. Applying CG with the positive definite matrix $Q^T A_\phi Q$ will require an $\mathcal{O}(N \log N)$ matrix-vector product per iteration and Q multiplied by a vector. Calculating the inner product $u_j^T y$ before the outer product $u_j u_j^T$ ensures

$$H_j y = y - 2 \frac{u_j^T y}{u_j^T u_j} u_j,$$

is found in $\mathcal{O}(N)$ operations. The product Qy can then be calculated in $\mathcal{O}(N)$ operations.

Powell gives numerical results for the thin-plate spline in \mathcal{R}^2 that indicate that this iterative technique will require a large iteration count when N is moderate or large. However this technique is still of interest for small problems since the symmetric positive definite system can be solved via the Cholesky factorisation in approximately half the operations required to solve a similarly sized problem with a variant of Gauss elimination that does not exploit symmetry.

1.6 Fast matrix-vector product algorithms

An essential ingredient in any fast technique for fitting a RBF is a fast matrix-vector product algorithm. In fact, without such an algorithm, any iterative method for fitting would still be cumbersome for large N . These fast algorithms allow the computation of a matrix-vector product in $\mathcal{O}(N \log N)$ or $\mathcal{O}(N)$ operations after $\mathcal{O}(N \log N)$ setup. The first fast algorithm of this type was presented by Greengard and Rokhlin [43] for use in potential theory with the basic function $\phi(\cdot) = \log(\cdot)$. Since then fast algorithms have been developed by Beatson and others for the multiquadric basic function [15, 7] and the thin-plate spline basic function [11, 8]. Beatson and Greengard in a review paper [10] give the following key features for a fast hierarchical or multipole method.

- A user specified accuracy level ϵ .
- A hierarchical subdivision of space into panels.
- Far field expansions (or approximations) to the basic function Φ to within a given accuracy.
- (*Optional*) Conversion of far field series into near-field series.

This section reviews the key features of a hierarchical method and the moment method. Following [10] we develop a hierarchical scheme in one dimension so that the key ideas can be understood. Moving to higher dimensions is then conceptually easy. Consider the multiquadric in one dimension. The far field expansion is

$$\phi(|x-t|) = \sqrt{|x-t|^2 + c^2} = x-t + \frac{1}{2}c^2x^{-1} + \frac{1}{2}tc^2x^{-2} + \dots + Q_p(c, t)x^{-p+1} + \mathcal{O}(|x|^{-p}), \quad (1.33)$$

which is valid provided $|x| > \sqrt{t^2 + c^2}$. Note that in the series (1.33) the general term is a product

$$x^{-j+1}Q_j(c, t), \quad (1.34)$$

where $Q_j(c, t)$ is given in [15]. In this product the influence of the source, t , and the evaluation point, x , is separated. Let

$$s_T(x) = \sum_{i:t_i \in T} d_i \phi(|x - t_i|),$$

be the influence of the centres $t_i \in T$ (where T is a panel centred at zero). Then if $|x| > \sqrt{t_i^2 + c^2}$, for all $t_i \in T$ and the series in (1.33) is truncated to p terms,

$$\begin{aligned} s_T(x) &\approx \sum_{i:t_i \in T} d_i \sum_{j=1}^p x^{-j+1} Q_j(c, t_i), \\ &= \sum_{j=1}^p x^{-j+1} \hat{Q}_j(c, T) =: r_T(x). \end{aligned}$$

Once the coefficients \hat{Q}_j are formed evaluating r_T takes $\mathcal{O}(p)$ operations. Clearly as the number of centres in T increases then evaluating r_T becomes comparatively more efficient compared to evaluating s_T directly. It is this idea that leads to the fast algorithms. Note that the length of the series, p , depends on the required accuracy, ϵ . For more details see [10].

Now, consider if all the centres, t_i , are contained in a panel T with centre t and radius h . Then the approximation r_T converges rapidly to s_T whenever $|x - t| \geq 3h$. These points, x , are referred to as being well-separated from T . The approximation r_T is valid whenever x is well-separated from T .

The final component required is a subdivision of the space containing the centres. Starting with a parent panel containing all the centres and then recursively dividing each parent panel in half to form two child panels gives a tree structure as in Figure 1.4.

To evaluate $s(x)$, for x in some panel $[3/4, 7/8)$ say, use a combination of approximations, $r_T(x)$, and exact evaluations, $s_T(x)$. The ‘‘source’’ panels to use approximations from are those on the interaction list of the target panel.

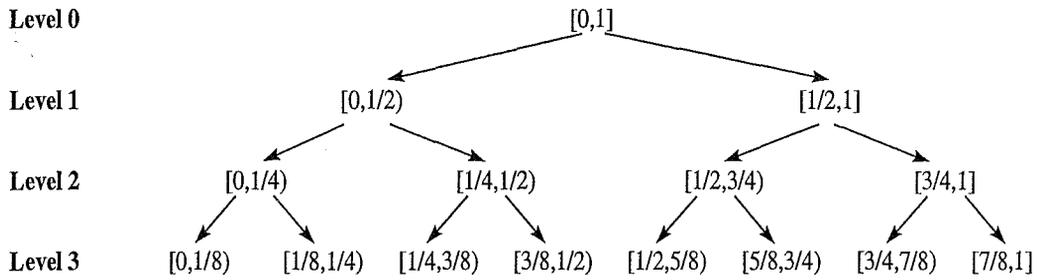


Figure 1.4: A three level subdivision of the unit interval.

Definition 1.6.1. A panel, T , is in the interaction list of a panel U if

- T is well-separated from the ancestor of U that is at the same level as T .
- The parents of T are not in the interaction list.

For $x \in [3/4, 7/8)$ the following approximation for $s(x)$ is obtained.

$$s(x) \approx s_{[3/4,7/8)}(x) + s_{[7/8,1]}(x) + s_{[5/8,3/4)}(x) + r_{[1/2,5/8)}(x) \\ + r_{[1/4,1/2)}(x) + r_{[0,1/4)}(x).$$

The centres in panels $[3/4, 7/8)$, $[5/8, 3/4)$ and $[7/8, 1]$ are clearly not well-separated from $[3/4, 7/8)$ at any level so the influence of these centres is calculated directly. The panels $[1/4, 1/2)$ and $[0, 1/4)$ are well-separated from the panel containing x at level 2. Thus the influence of these centres is approximated at a higher level.

Now consider if x is in the panel $[1/4, 3/8)$, then we have the approximation,

$$s(x) \approx s_{[1/4,3/8)}(x) + s_{[3/8,1/2)}(x) + s_{[1/8,1/4)}(x) \\ + r_{[0,1/8)}(x) + r_{[1/2,5/8)}(x) + r_{[5/8,3/4)}(x) + r_{[3/4,1]}(x).$$

As before the influence of panels that are not well-separated from the target panel are calculated directly and the far field series are used for panels on the interaction list.

If the centres are uniformly distributed then the number of levels is usually $m \approx \log_2 N$. Then there are $\mathcal{O}(1)$ centres in each level m panel. Because there is

at most three panels at each level on the interaction list of a given target panel and each series is length p , then the work for evaluating $s(x)$ at a single point x will be $\mathcal{O}(pm) = \mathcal{O}(p \log_2 N)$. The setup cost can be shown to be $\mathcal{O}(N \log N)$ and the total cost for an approximate matrix vector product is $\mathcal{O}(pN \log N)$ operations.

1.6.1 Moment method

The moment method [15, 5] is a method which allows the evaluation of an RBF s in $\mathcal{O}(1)$ operations after $\mathcal{O}(N \log N)$ setup. The advantage of the moment method is that it can be applied to a wide range of functions by only changing a few lines of code. In particular there is no need for far field series for each new basic function.

The moment method algorithm forms an approximation to ϕ by interpolating to ϕ at the zeros of the shifted Chebychev polynomials. As the approximations are only needed for panels on the interaction list at each level these approximations are formed so that they are sufficiently accurate in these panels. Hence, new polynomial approximations are required at each level. The following lemma shows that once such approximations are found we can find a good approximation to s .

Lemma 1.6.2 (Beatson and Newsam [15]). *Let b, c and $\epsilon > 0$. Let $|t| > b + c$ and ϕ be a function in $C[t - (b + c), t + (b + c)]$. Define $V_j(x) = x^j/j!$ and let*

$$q(x) = \sum_{j=0}^k a_j V_j(x - t),$$

be a polynomial of degree k such that $\|\phi(\cdot) - q\|_{L^\infty[t-(b+c), t+(b+c)]} \leq \epsilon$. Given centres x_1, \dots, x_M with $|x_m| \leq b$ for $1 \leq m \leq M$ and weights d_1, \dots, d_M , let the corresponding RBF

$$s(x) = \sum_{m=1}^M d_m \phi(|x - x_m|)$$

be approximated by

$$s_1(x) = \sum_{m=1}^M d_m q(x - x_m).$$

Then $\|s - s_1\|_{L^\infty[t-c, t+c]} \leq \epsilon \|d\|_1$. Moreover,

$$s_1(x) = \sum_{l=0}^k b_l V_l(x - t),$$

where $b_l = \sum_{j=l}^k a_j \sum_{m=1}^M d_m V_{j-l}(-x_m)$.

By summarising the influence of all the panels on the interaction list into a local polynomial approximately evaluating s reduces to evaluating the influence of nearby centres directly together with evaluation of a polynomial. In Table 1.6.1 we give various timing results for a matrix-vector product using the moment method and direct computation. It can clearly be seen that the moment method provides huge computational savings when N is large and is even competitive for small values of N . The moment method timings include the time for setup which is a substantial

Basic function	Number of centres, N	Direct calculation	Moment method	Ratio
linear	2500	0.92	0.38	2.42
	5000	3.68	0.83	4.43
	10000	14.74	1.77	8.33
	20000	59.16	4.58	12.92
	40000	239.52	11.02	21.74
exponential	2500	2.05	0.60	3.42
	5000	8.14	1.32	6.17
	10000	32.93	2.79	11.80
	20000	131.43	6.71	19.59
	40000	530.19	15.41	34.41

Table 1.1: Timings of matrix-vector products by direct multiplication and by the moment method for centres in \mathcal{R}^2 .

portion of the algorithm. Subsequent matrix-vectors would not include this setup and consequently timings would be somewhat reduced.

Chapter 2

Preconditioned GMRES iteration

2.1 Introduction

Radial basis functions (RBFs) are a recent tool for interpolating data. Applications of RBFs include bathymetry (ocean depth measurement), topography (altitude measurements), hydrology (rainfall interpolation), surveying, mapping, geophysics and geology (see the survey of applications [44]). More recent applications include image warping [4, 38] and medical imaging [21]. Experience in a variety of applications has shown RBFs to be particularly well suited to scattered data interpolation problems.

An RBF, s , is a function of the form

$$s(\cdot) = p_{m-1}(\cdot) + \sum_{i=1}^N \lambda_i \phi(|\cdot - x_i|). \quad (2.1)$$

Here, $p_{m-1}(\cdot)$ is a member of π_{m-1}^d , the space of $(m-1)^{\text{th}}$ degree polynomials in d variables, $\lambda = [\lambda_1, \dots, \lambda_N]^T \in \mathcal{R}^N$ are coefficients and $\{x_1, \dots, x_N\} = X$ are the distinct centres of the RBF. The function ϕ is SCPD m , and $|\cdot|$ denotes the 2-norm. Popular choices of basic function include the multiquadric $\phi(r) = \sqrt{r^2 + c^2}$, $c > 0$ and the thin-plate spline $\phi(r) = r^2 \log(r)$. Radial basis functions are often fitted by interpolation at the centres x_i . Interpolation with a function of form (2.1) leads to

the system of interpolation equations

$$\begin{bmatrix} A_\phi & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ a \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}. \quad (2.2)$$

Here $(A_\phi)_{ij} = \phi(|x_i - x_j|)$, f is the vector of values to be interpolated, a is the vector of coefficients of p_{m-1} with respect to some basis $\{q_1, \dots, q_{\dim(\pi_{m-1}^d)}\}$ for π_{m-1}^d , and $P_{ij} = q_j(x_i)$.

We shall be interested in the conditioning and spectrum of the matrix

$$A = \begin{bmatrix} A_\phi & P \\ P^T & 0 \end{bmatrix}, \quad (2.3)$$

on the left of the usual interpolation system (2.2). The second row of the partitioned system (2.2) is a set of side conditions that can be rewritten as

$$\sum_{j=1}^N \lambda_j q(x_j) = 0, \quad \text{for all } q \in \pi_{m-1}^d. \quad (2.4)$$

Loosely speaking these moment conditions correspond to λ being orthogonal to the space $\pi_{m-1}(X)$.

As we saw in Chapter 1 the RBF interpolation matrix will be invertible under very weak conditions on the geometry of the centres. For example in \mathcal{R}^2 the system corresponding to the thin-plate spline interpolation will be invertible whenever the centres are not collinear. This guarantee of invertibility is a considerable advantage over many other interpolation methods. Indeed, according to the counter-example of Mairhuber, any method for which the space of interpolants does not change with the centres of interpolation is doomed to fail for at least one choice of $N \geq 2$ centres in \mathcal{R}^d , $d \geq 2$. In RBF interpolation the space from which the interpolant is chosen changes with the centres of interpolation. Thus the Mairhuber counter example does not apply. Unfortunately the matrix (2.3) is not usually sparse and when the number of interpolation points, N , is large then the system is usually ill-conditioned. Solving this system by non-customised methods exploiting symmetry requires $\mathcal{O}(N^3/6)$

flops* and $\mathcal{O}(N^2)$ storage. These computational costs are unacceptable for large N .

Experience with RBF interpolants on small problems has been almost universally positive. However, applications to large problems with 10,000 or more centres have been limited due to the perceived prohibitive computational costs. The literature contains many comments on the desirable properties of RBF interpolants and the expense of fitting and evaluating them. For example:

“The global interpolation methods with Duchon’s “thin plate splines” and Hardy’s multiquadrics are considered to be of high quality; however, their application is limited, due to computational difficulties, to ~ 150 data points.” *Dyn, Levin and Rippa* [31, 1986]—talking about the state of the art prior to their paper.

“Practical problems often arise with many more than 10,000 data sites; for example, in aeromagnetic survey work it is common to have 50,000 to 100,000 observations in a single data set. We believe that such problems will indefinitely remain beyond the scope of thin-plate splines.” *Sibson and Stone* [77, 1991].

“The most accurate results of registration of images with local distortions were obtained by using the surface spline mapping functions. As shown below, their direct use has extreme computing complexity and is not suitable for practical applications.” *Flusser* [38, 1992].

The purpose of this chapter is to present one method which overcomes the computational complexity and conditioning difficulties presented by large RBF interpolation problems. One feature of the method is that the problem is preconditioned by changing the basis in which the RBFs are represented. Several choices of approximate cardinal function preconditioner have been tried, all with at least some degree

*We define a flop as one addition or subtraction, plus one multiplication or division, and some indexing. Some authors would count such a combination of operations as two flops.

of success. Extensive numerical experiments have been conducted with thin-plate spline and multiquadric RBFs. These show that the combination of a suitable approximate cardinal function preconditioner, a fast multiply, and GMRES iteration, can make the solution of large RBF interpolation problems orders of magnitude less expensive in storage and operations. Typically, the preconditioning results in dramatic clustering of eigenvalues and improves the condition numbers of the interpolation problem by several orders of magnitude. Known convergence estimates for the GMRES iterative method show that eigenvalue clustering should speed convergence. Such behaviour is very much in evidence in our computations; the number of GMRES iterations required to solve the preconditioned problem being typically of the order of 10 to 20 and growing slowly with N .

In summary the combination of a suitable approximate cardinal function preconditioner, the GMRES iterative method, and existing fast matrix-vector algorithms for RBFs [11, 14] lowers the computational cost of solving an RBF interpolation problem to $\mathcal{O}(N)$ storage and $\mathcal{O}(N \log N)$ operations. This makes computations with 10,000 or more centres an easy task, even with very modest hardware.

2.2 The GMRES iterative method

The GMRES iterative method for non-symmetric systems, $Ax = b$, was presented by Saad and Schultz [72] in 1986. The algorithm is a Krylov subspace method and its convergence properties parallel those of the conjugate gradient (CG) method for symmetric positive definite systems. Unlike CG iteration, GMRES iteration requires storage of all the previous search directions, or equivalently storage of an orthonormal basis for κ_k , where

$$\kappa_k = \text{span} \{r_0, Ar_0, \dots, A^{k-1}r_0\},$$

and $r_k = b - Ax_k$ is the residual at the k^{th} step of GMRES.

In the k^{th} iteration of GMRES x_k is taken as the unique solution of the least

squares problem

$$\min_{x \in x_0 + \kappa_k} \|b - Ax\|_2. \quad (2.5)$$

It is well known, see for example [53, pp.33-34], that if A is invertible

$$\|r_k\|_2 = \min_{p \in \pi_k^1, p(0)=1} \|p(A)r_0\|_2.$$

Furthermore, if A is diagonalisable with $A = VDV^{-1}$ and $p_k \in \pi_k^1$ is any polynomial with $p_k(0) = 1$ then

$$\|r_k\|_2 \leq \|r_0\|_2 \operatorname{cond}_2(V) \max_{z \in \sigma(A)} |p_k(z)|,$$

where $\sigma(A)$ is the spectrum of A . If the eigenvalues of A fall into a single cluster about c with relative radius $\rho < 1$ then choosing

$$p_k(z) = \frac{(c - z)^k}{c^k}, \quad (2.6)$$

in the above yields the estimate

$$\|r_k\|_2 \leq \|r_0\|_2 \operatorname{cond}_2(V) \rho^k,$$

for the error after k iterations.

Campbell et al. [18], with a more sophisticated analysis, show the bound

$$\|r_{u+k}\|_2 \leq \|r_0\|_2 C \rho^k, \quad (2.7)$$

when A , not necessarily diagonalisable, has a single cluster of eigenvalues together with some outliers. In equation (2.7), u is the number of outlying eigenvalues, and C reflects the relative distance of outliers from cluster and the non-normality of A . ρ is the relative radius of the clustered eigenvalues.

The k^{th} iteration of the GMRES algorithm on an $N \times N$ system requires one matrix-vector product and $\mathcal{O}(kN)$ additional floating point operations [53, 72, 84]. The method also requires the storage of an orthonormal basis for the Krylov subspace so that conjugate vectors can be formed at each iteration. Hence, if the total number of iterations is K , total storage requirements, excluding any storage needed for the

matrix A or computing its action, is $\mathcal{O}(KN)$. The corresponding flop count is K matrix-vector products and $\mathcal{O}(K^2N)$ other floating point operations.

In our application the preconditioner will, by design, cluster most of the eigenvalues near one. Thus the total number, K , of iterations will be small, usually less than 20, and the computational costs of approximate solution via GMRES, excluding those related to computing matrix-vector products, will be very moderate. The cost of each matrix-vector product will be reduced to $\mathcal{O}(N)$ storage and $\mathcal{O}(N \log N)$ flops using a fast algorithm.

2.3 Preconditioning: Good basis versus bad basis

Radial basis function interpolation problems have acquired a reputation for being badly conditioned. In our view this reputation is somewhat undeserved. In this section we give an empirical justification of our viewpoint. The numerical results of Section 5 provide quantitative support for it.

Our empirical argument is based on several other examples in numerical analysis. First, consider the problem of fitting a polynomial by least squares to uniformly spaced data on $[0, 1]$. If the power basis is used then the problem is horribly conditioned, the matrix involved being close to the Hilbert matrix. If however an appropriate basis of orthogonal polynomials is used the problem is beautifully conditioned, the matrix involved being close to the identity matrix. Second, consider the problem of fitting a univariate polynomial spline of degree k on mesh, $\mathbf{t} : t_0 < t_1 < \dots < t_N$, to data. For the purpose of finding the dimension of the spline space, considered as a subspace of $C[t_0, t_N]$, the expression

$$s(\cdot) = p(\cdot) + \sum_{i=1}^{N-1} \lambda_i (\cdot - t_i)_+^k ,$$

is extremely good. However, for the purpose of fitting a spline to numerical data it is awful, the matrix involved typically being badly conditioned. The remedy is

again to change the basis, the basis of normalised B-splines being one much better choice.

We have seen above that the conditioning of data fitting problems is heavily dependent on the choice of basis. If a *bad* basis is used then such problems are often badly conditioned. However, if a *good* basis is used the problems are well conditioned. The properties of a single basis are not the properties of the space! In our opinion the reputation of RBF interpolation problems for ill-conditioning is due to most computations being done with respect to the natural basis, which happens to be *bad* for numerical fitting of interpolants. After all if one used the basis of cardinal RBFs[†], the matrix of the interpolation problem would be the identity, which has perfectly clustered eigenvalues and is perfectly conditioned. Unfortunately, finding all the cardinal functions is not a practical solution, being more expensive than interpolation to one right hand side. Therefore our aim in the next section is to investigate practical strategies for finding *good* bases for RBF interpolation problems.

2.4 Preconditioning: Approximate cardinal functions

In this section we shall introduce preconditioning strategies based on a change of basis to a basis of *approximate cardinal functions*. We have seen in Section 2.2, concerning the GMRES algorithm, that a preconditioner that clusters eigenvalues should result in fast convergence. We have also seen in Section 2.3, on good bases versus bad bases, that using the basis of cardinal functions would result in the interpolation matrix being the identity matrix, with all eigenvalues equal to 1. For this matrix GMRES would converge in one iteration! However, converting to the basis of cardinal functions is totally impractical, so we choose instead to follow [9, 16] and convert to a basis of approximate cardinal functions.

[†]Each basis element is one at one node and zero at the others.

All the approximate cardinal functions which occur below fall into the general pattern discussed in this paragraph. Given centres x_0, x_1, \dots, x_N one associates with x_j an approximate cardinal function

$$\psi_j(\cdot) = p_j(\cdot) + \sum_{i=1}^N \nu_{ji} \phi(|\cdot - x_i|), \quad (2.8)$$

where ψ_j is an element of our spline space. Thus p_j is an element of π_{m-1}^d and the coefficients $\{\nu_{ji}\}_{i=1}^N$ are orthogonal to polynomials in the sense of equation (2.4). Various choices of the manner in which ψ_j 's approximate the corresponding cardinal function will be discussed later. Suppose there is a relatively small number, $\beta \ll N$, such that for each j the number of possibly non-zero coefficients ν_{ji} is less than or equal to β . Thus to find ψ_j we solve a system of size approximately $\beta \times \beta$. Then the setup cost, typically $\mathcal{O}(N\beta^3/6)$ flops, of solving N small systems to find the coefficients of all the elements ψ_j will also be small compared with the cost of solving the original large unpreconditioned system, which is typically $\mathcal{O}(N^3/6)$ flops. See Table 1 below for more precise estimates in the case of the thin-plate spline in \mathcal{R}^2 . In this case we estimate the cost of forming a single ψ_j as approximately $(\beta - 3)^3/6 + \beta^2 + 6\beta^2 + 5\beta^2 + \mathcal{O}(\beta)$ flops. Here the $5\beta^2$ is an estimate of the cost of forming the lower triangle of the matrix $(\phi(|x_i - x_k|))$ for i, k in the subset of at most β indices \mathcal{S}_j (see below); the $6\beta^2$ is the cost of obtaining from that matrix Powell's $(\beta - 3) \times (\beta - 3)$ Householder preconditioned matrix [69] (performing the transformation exploiting symmetry); the $(\beta - 3)^3/6$ is the cost in flops of Cholesky decomposition of the Householder preconditioned matrix; and the β^2 the cost of solution by forward and back substitution.

In terms of the ψ functions the interpolation problem is

$$A_\psi \mu = f, \quad (2.9)$$

where A_ψ has (i, j) entry $\psi_j(x_i)$. If our choice of the functions ψ_j is successful then A_ψ will have clustered eigenvalues and (2.9) will be solvable in relatively few GMRES iterations. These iterations themselves are not too expensive. The main

Number of centres, N .	Number of ϕ 's in each ψ , β .	Ratio of flop counts. Solution of single large unpreconditioned system versus forming the N ψ functions.
1000	30	13
4225	50	64
10000	50	355
20000	50	1414

Table 2.1: Ratio of the flop count for solving the single large unpreconditioned system by Cholesky compared to the flop count for forming the ψ functions.

cost of an iteration is to compute the action of A_ψ on a vector. This can be done by converting a spline from representation with respect to the good, or ψ , basis into terms of the bad, or ϕ , basis in $\mathcal{O}(\beta N)$ flops and multiplying by the matrix of the interpolation problem with respect to the bad basis. The latter multiplication taking $\mathcal{O}(N^2)$ flops if performed directly, or $\mathcal{O}(N \log N)$ flops if a fast algorithm is used. Indeed iterative methods for solving the interpolation equations were part of the motivation for the development of fast algorithms in [11, 14].

2.4.1 Approximate cardinal functions based on purely local centres

As mentioned earlier we want to restrict the number of nonzero coefficients, ν_{ji} , to be small, for each j . Therefore, for any j , we select a group of β points with possibly non-zero ν_{ji} 's and force the remaining $N - \beta$ coefficients ν_{ji} to be zero. In the *pure local* method we take this set of β points to be a set of β closest points to x_j . For some data sets and some x_j 's this set may not be unique. The corresponding set of indices will be denoted hereafter by \mathcal{S}_j . Finding all the subsets \mathcal{S}_j can be performed efficiently in $\mathcal{O}(N \log N)$ time with a balanced range tree. The *pure local* strategy approximates the true cardinal function well for nodes near x_j . However, as we will

see later, the approximation can be poor at points far away from x_j . The *pure local* approximate cardinal function ψ_j will have the form

$$\psi_j(\cdot) = p_j(\cdot) + \sum_{i \in \mathcal{S}_j} \nu_{ji} \phi(|\cdot - x_i|), \quad (2.10)$$

and satisfy the interpolation conditions

$$\psi_j(x_i) = \delta_{ij}, \quad i \in \mathcal{S}_j. \quad (2.11)$$

The coefficients of ψ_j can be found by solving an appropriate $(\beta + \ell) \times (\beta + \ell)$ system similar to (2.2), where $\ell = \dim(\pi_{m-1}^d)$.

This subproblem can be solved by Gaussian elimination in $\mathcal{O}(\beta^3/3)$ flops after $\mathcal{O}(\beta^2)$ setup. Bad conditioning in the subproblem may occur due to the conditioning of the interpolation matrix being scale dependent. For the thin-plate spline the subproblem can be solved in a scale independent manner by a transformation to a symmetric positive definite system. See Powell [69] for a full description of one such transformation. Somewhat surprisingly for the data sets we've considered using the scale independent approach made no significant difference to the convergence of our GMRES iterations. However moving to a symmetric positive definite system makes it easy to find ψ_j in $\mathcal{O}(\beta^3/6)$ flops. This approximate halving of setup costs is very worthwhile. Figure 2.1 below shows *pure local* approximate cardinal functions based on using 50 centres from the uniform grid $\{0, 1/16, \dots, 1\} \times \{0, 1/16, \dots, 1\}$ of 289 centres. Note that in the typical case of closest points all around the *pure local* approximate cardinal function works very well being close to zero or displaying small linear growth far away from x_j . When the closest point set is unbalanced then the approximate cardinal function shows strong linear growth as $|x - x_j|$ gets large. In the figures a dark asterisk on the top of the 3D box indicates a centre used in the element. A light dot indicates an unused centre.

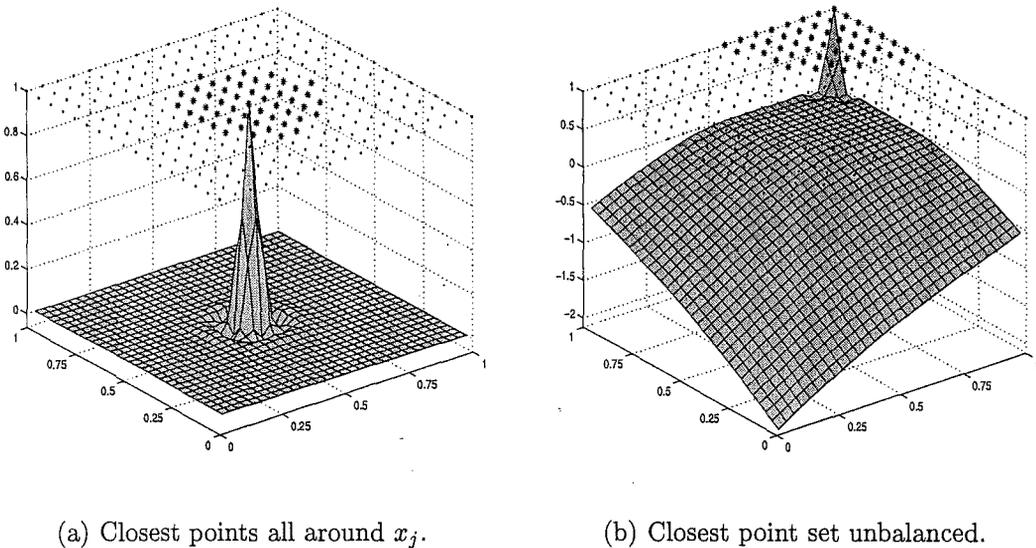


Figure 2.1: *Pure local* thin-plate spline approximate cardinal functions based on 50 local centres from a 17×17 grid.

2.4.2 Approximate cardinal functions based on local centres and special points

To counter the growth of the *pure local* ψ_j 's described above we add in special points to the local node sets specified by \mathcal{S}_j . Our heuristic is to clamp ψ_j to be zero at these widely scattered special points, and expect ψ_j 's smoothness to constrain it to be small near these points. In the thin-plate spline case the variational characterisation argues against any large deviation of $\psi_j(x)$ from zero in regions containing only special points. These special points will be scattered widely within the domain. For example if the centres of interpolation are distributed within the square $[0, 1] \times [0, 1]$ and we wanted 4 special points, then these could be chosen as the centres closest to $(0, 0)$, $(0, 1)$, $(1, 0)$ and $(1, 1)$ respectively. Interpolating as before to δ_{ij} data gives an approximate cardinal function which is a good approximation near x_j , and small far away from x_j . Figure 2.2 shows that this typically leads to a good approximation in between these regions as well.

In this chapter we consider the 4 special points just mentioned and also 9 special points which are found by forming a 3×3 grid over the data rectangle, and then finding the centres closest to the 9 vertices.

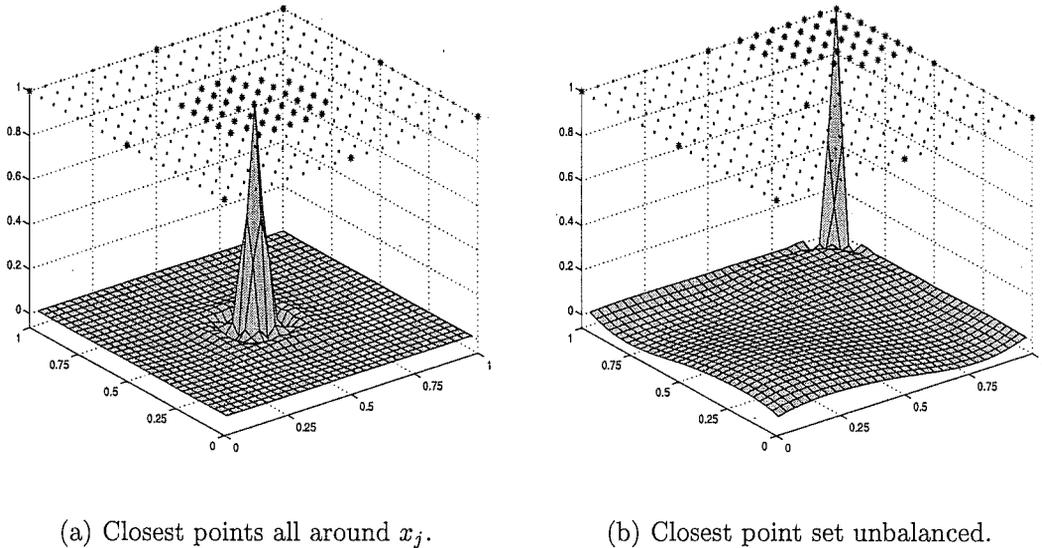


Figure 2.2: *Local centres and special points* thin-plate spline approximate cardinal functions based on 41 local centres and 9 special points from a 17×17 grid.

2.4.3 Decay element approximate cardinal functions with r^{-3} growth at infinity

The *decay element* approximate cardinal functions, ψ_j , are constructed to enforce decay as x moves away from x_j . The idea is that then the entries in the matrix A_ψ corresponding to $|x_i - x_j|$ large, will be very small. Hence, corresponding columns of A_ψ will probably be column diagonally dominant, with 1 as the diagonal element. If this held for every j then A_ψ would have eigenvalues clustered about 1 by Gershgorin. We find ψ_j by solving the constrained least squares problem

$$\psi_j(x_i) \approx \delta_{ij}, \quad i \in \mathcal{S}_j, \quad (2.12)$$

subject to

$$\psi_j(x) = \mathcal{O}(|x|^{-3}) \text{ as } |x| \rightarrow \infty. \quad (2.13)$$

The growth condition (2.13) is equivalent to a homogeneous system of linear constraints on the coefficients $\{\nu_{ji} : i \in \mathcal{S}_j\}$ and as we shall see below requires the polynomial p_j in (2.10) to be zero. Thus the system to find the coefficients of a single *decay element* approximate cardinal function takes the form

$$\min_{\nu} \|G\nu - e\|_2,$$

subject to

$$H\nu = 0,$$

where G is the β by β matrix corresponding to interpolation by a “pure” sum of shifts of $\phi(|\cdot|)$, H is the matrix of conditions given in equation (2.16) for the multiquadric and (2.18) for the thin-plate spline, and e is the vector with 1 in the row corresponding to j and zeros elsewhere.

In what follows we will adopt the notation $x = (\xi, \eta)$ and $x_i = (\xi_i, \eta_i)$. The truncated far field expansion of a single multiquadric basic function centred at (s, t) (see [7]) is

$$\begin{aligned} & \sqrt{(\xi - s)^2 + (\eta - t)^2 + c^2} \\ &= \sqrt{\xi^2 + \eta^2} - \frac{t\eta + s\xi}{\sqrt{\xi^2 + \eta^2}} + \frac{1}{2} \frac{(t^2 + c^2)\xi^2 + (s^2 + c^2)\eta^2 - 2st\xi\eta}{(\xi^2 + \eta^2)^{\frac{3}{2}}} \\ &+ \frac{1}{2} \frac{(s\xi + t\eta) \{(t^2 + c^2)\xi^2 + (s^2 + c^2)\eta^2 - 2st\xi\eta\}}{(\xi^2 + \eta^2)^{\frac{5}{2}}} + \mathcal{O}(|x|^{-3}). \end{aligned} \quad (2.14)$$

Substituting the far field expansion from equation (2.14) into the expression for

$\psi_j(x)$ we find

$$\begin{aligned}
\psi_j(x) &= p_j(x) + \sum_{i \in \mathcal{S}_j} \nu_{ji} \sqrt{\xi^2 + \eta^2} - \sum_{i \in \mathcal{S}_j} \nu_{ji} \frac{\xi_i \xi + \eta_i \eta}{\sqrt{\xi^2 + \eta^2}} \\
&+ \frac{1}{2} \sum_{i \in \mathcal{S}_j} \nu_{ji} \frac{(\eta_i^2 + c^2)\xi^2 + (\xi_i^2 + c^2)\eta^2 - 2\xi_i \eta_i \xi \eta}{(\xi^2 + \eta^2)^{\frac{3}{2}}} \\
&+ \frac{1}{2} \sum_{i \in \mathcal{S}_j} \nu_{ji} \frac{(\xi_i \xi + \eta_i \eta) \{(\eta_i^2 + c^2)\xi^2 + (\xi_i^2 + c^2)\eta^2 - 2\xi_i \eta_i \xi \eta\}}{(\xi^2 + \eta^2)^{\frac{5}{2}}} + \mathcal{O}(|x|^{-3}).
\end{aligned} \tag{2.15}$$

Hence, when c is constant, $\psi_j(x)$ will be $\mathcal{O}(|x|^{-3})$ as $|x| \rightarrow \infty$ if and only if $p_j = 0$ and

$$\sum_{i \in \mathcal{S}_j} \nu_{ji} x_i^\alpha = 0 \quad \text{for all } \alpha = (\alpha_1, \alpha_2) \in \mathcal{N}_0^2 \text{ such that } (\alpha_1 + \alpha_2) \leq 3, \tag{2.16}$$

where \mathcal{N}_0^2 is the space of ordered pairs of nonnegative integers.

The truncated far field expansion of a single thin-plate spline basic function centred at (s, t) (see [11, Theorem 1], [14, Lemma 2]) is

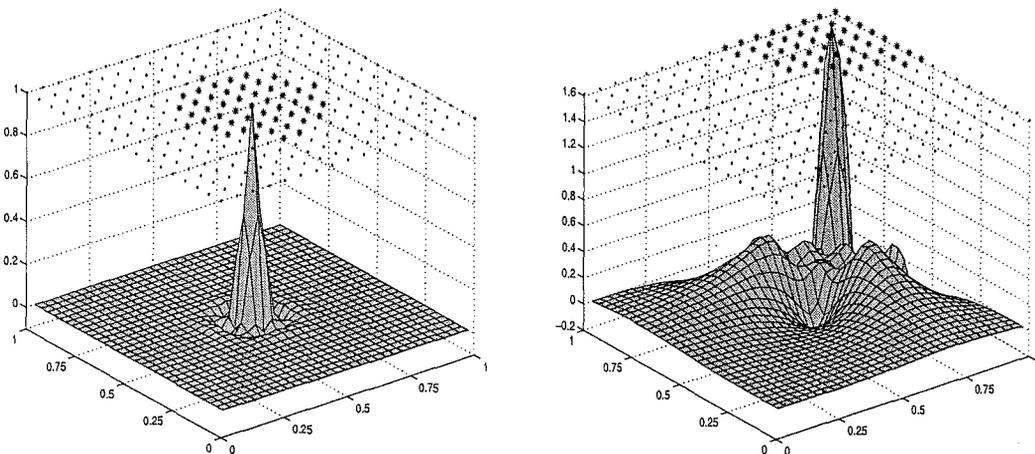
$$\begin{aligned}
& [(\xi - s)^2 + (\eta - t)^2] \log \left([(\xi - s)^2 + (\eta - t)^2]^{\frac{1}{2}} \right) \\
&= \frac{1}{2} (\xi^2 + \eta^2) \log(\xi^2 + \eta^2) - (s\xi + t\eta) \log(\xi^2 + \eta^2) - s\xi - t\eta \\
&+ \frac{1}{2} (s^2 + t^2) \log(\xi^2 + \eta^2) + \frac{1}{2} \frac{(3s^2 + t^2)\xi^2 + (s^2 + 3t^2)\eta^2 + 4st\xi\eta}{\xi^2 + \eta^2} \\
&- \frac{1}{3} \frac{(s\xi + t\eta) \left((s^2 + 3t^2)\xi^2 + (3s^2 + t^2)\eta^2 - 4st\xi\eta \right)}{(\xi^2 + \eta^2)^2} \\
&- \frac{1}{12} \frac{6(s^4 + t^4 - 6s^2t^2)\xi^2\eta^2 + (6s^2t^2 + s^4 - 3t^4)\xi^4 + (6t^2s^2 - 3s^4 + t^4)\eta^4}{(\xi^2 + \eta^2)^3} \\
&- \frac{1}{12} \frac{8(3st^3 - s^3t)\xi^3\eta + 8(3s^3t - t^3s)\xi\eta^3}{(\xi^2 + \eta^2)^3} + \mathcal{O}(|x|^{-3}).
\end{aligned} \tag{2.17}$$

By an argument analogous to that leading from (2.14) to (2.16) we find from (2.17) that a ψ_j based on thin plate splines is $\mathcal{O}(|x|^{-3})$ as $|x| \rightarrow \infty$ if and only if certain

moment conditions hold. These are $p_j = 0$ and

$$\begin{aligned}
 \sum_{i \in \mathcal{S}_j} \nu_{ji} x_i^\alpha &= 0 \quad \text{for all } \alpha \in \mathcal{N}_0^2 \text{ such that } (\alpha_1 + \alpha_2) \leq 3, \\
 \sum_{i \in \mathcal{S}_j} \nu_{ji} (\xi_i^4 + \eta_i^4 - 6\xi_i^2\eta_i^2) &= 0, \\
 \sum_{i \in \mathcal{S}_j} \nu_{ji} (\xi_i^4 - \eta_i^4) &= 0, \\
 \sum_{i \in \mathcal{S}_j} \nu_{ji} \eta_i \xi_i^3 &= 0, \\
 \sum_{i \in \mathcal{S}_j} \nu_{ji} \eta_i^3 \xi_i &= 0.
 \end{aligned} \tag{2.18}$$

Figure 2.3 shows the resulting thin-plate spline *decay element* approximate cardinal functions for two different configurations of centres.



(a) Closest points all around x_j .

(b) Closest point set unbalanced.

Figure 2.3: *Decay element* thin-plate spline approximate cardinal functions based on 50 local centres from a 17×17 grid.

We cannot expect to be able to use decay elements exclusively as a basis. In view of the decay condition they do not span the whole space! Thus we need to use some non-decay elements as well. As a criteria we choose to use decay element, ψ_j ,

if and only if

$$\sum_{i \in \mathcal{S}_j} |\psi_j(x_i) - \delta_{ji}| < \mu, \quad (2.19)$$

for some suitable tolerance μ (typically 0.5). Otherwise we use an approximate cardinal function based on local centres and special points. We call centres for which the *decay element* satisfies condition (2.19) good points and centres for which the *decay element* fails condition (2.19) bad points. Figure 2.4 shows the good and bad points for a random set of 4225 centres from $[0, 1]^2$. In the figure dots mark centres for which the corresponding decay element satisfied condition (2.19). An asterisk marks a centre whose corresponding decay element failed the condition (2.19). In this example the tolerance was taken to be 0.5 and the size of \mathcal{S}_j is 50. Note that most points are good points, and bad points are generally on the periphery of the region or points with an unbalanced node set.

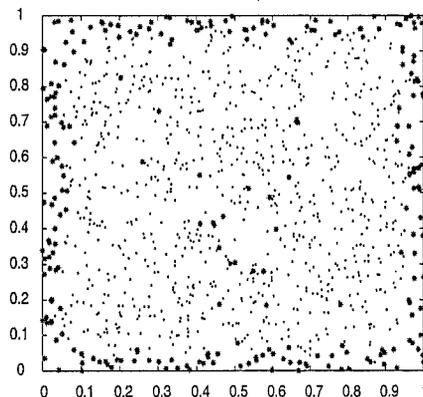


Figure 2.4: Good (\cdot) versus bad centres ($*$) for the thin-plate spline *decay element* approximate cardinal functions.

2.5 Numerical Results

In this section we present numerical results using the approximate cardinal functions described in Section 2.4. Results are presented in the form of eigenvalue plots, to

show how well the eigenvalues have been clustered by the preconditioner; condition numbers, to show that the conditioning of the A_ψ matrix is superior to that of the A matrix; and iteration counts, to show the speed of convergence using GMRES. All experiments are for problems in 2D with the size of \mathcal{S}_j as 50 and function values from the Franke function $F^{(1)}$

$$\begin{aligned}
 F^{(1)}(\xi, \eta) = & \frac{3}{4} \exp\left(-\frac{(9\xi - 2)^2 + (9\eta - 2)^2}{4}\right) + \frac{3}{4} \exp\left(-\frac{(9\xi + 1)^2}{49} - \frac{9\eta + 1}{10}\right) \\
 & + \frac{1}{2} \exp\left(-\frac{(9\xi - 7)^2 + (9\eta - 3)^2}{4}\right) - \frac{1}{5} \exp(-(9\xi - 4)^2 - (9\eta - 7)^2),
 \end{aligned}
 \tag{2.20}$$

of [32]. All sets of interpolation centres are fixed, initially uniformly at random points from $[0, 1]^2$. As well as the experiments with up to 10,000 nodes detailed below we have used the strategies of this chapter to solve larger geophysical data fitting problems in 2D, and larger image processing data fitting problems in 3D. In 3D with $\phi(r) = r$ we found it useful to take the size of \mathcal{S}_j somewhat larger, say 100.

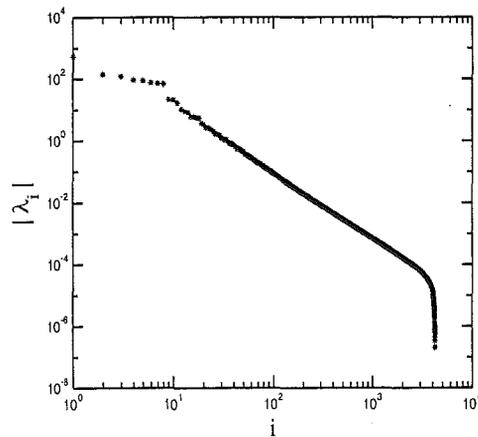


Figure 2.5: Eigenvalues of the thin-plate spline interpolation matrix of equation (2.3). Sorted absolute eigenvalues are plotted in order of decreasing magnitude.

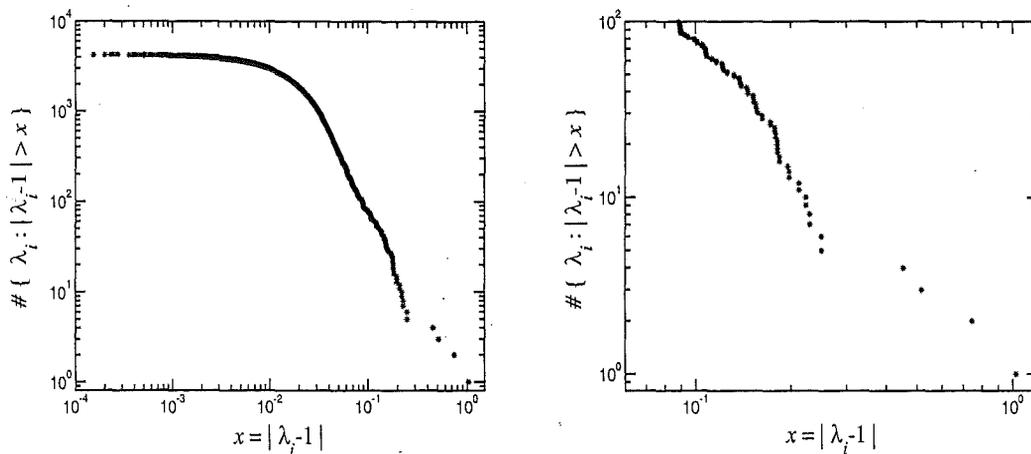
Figure 2.5 is a plot of eigenvalues from the A interpolation matrix (2.3) for thin-plate spline interpolation with 4225 nodes. It is apparent that the eigenvalues are

spread over a range of magnitudes with no clustering. Applying GMRES to the unpreconditioned interpolation problem (2.2) would therefore not be expected to be very successful. The iteration counts in Table 2.2 following the eigenvalue plots confirm that this is the case.

In Figures 2.6, 2.7 and 2.8 eigenvalues of the A_ψ interpolation matrix for the various approximate cardinal functions are presented. The plots on the left show all the eigenvalues whereas those on the right show only the 100 eigenvalues furthest from 1. The eigenvalue plots for the *decay element* approximate cardinal function in Figure 2.6 show that there are only 2 eigenvalues at a distance greater than 0.5 from 1. Consequently, after the first few iterations we would expect GMRES to start converging rapidly, since the bound (2.7) at least halves for all iterations after the second.

The *local centres and special points* approximate cardinal function and the *local centres* approximate cardinal functions are not as effective at clustering the eigenvalues with 10 and 25 eigenvalues at a distance greater than 0.5 from 1 respectively. As expected the iteration counts are larger for these 2 preconditioners.

Condition numbers of the A_ψ interpolation matrix and GMRES iteration counts for the thin-plate spline are given in Table 2.2. In this table an entry of the form $d_0.d_1d_2d_3(e)$ with d_0, d_1, d_2, d_3 decimal digits represents the number $d_0.d_1d_2d_3 \times 10^e$. Condition numbers for 10,000 nodes of interpolation are not given due to storage limitations. The first thing of note in the table is the high number of iteration counts to get to a mean square residual (MSR) of 10^{-12} for the unpreconditioned system. The cause of this, as mentioned earlier, is the poor clustering of the eigenvalues of the A matrix. This matrix is also badly conditioned with condition numbers of magnitude 10^{10} for 4225 points. The most exciting result from the table is the performance of the *decay element* approximate cardinal function. Only 14 iterations were needed to reduce the MSR to 10^{-12} with 10,000 centres. The iteration count was even less for the random data than it was for a 100×100 uniform grid of centres! The table also shows that the condition numbers of the linear systems



(a) Plot of $\#\{\lambda_i : |\lambda_i - 1| > x\}$ against x , for all eigenvalues.

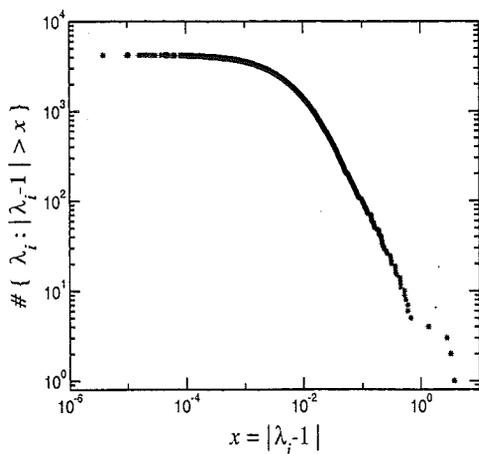
(b) Plot of $\#\{\lambda_i : |\lambda_i - 1| > x\}$ against x , for the 100 eigenvalues furthest from 1.

Figure 2.6: Eigenvalues of the preconditioned interpolation matrix for the *decay element* approximate cardinal function. The ϕ function used is the thin-plate spline and the nodes of interpolation are 4225 random points in $[0, 1]^2$.

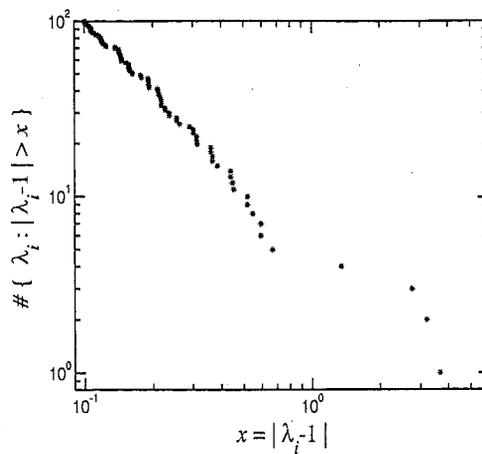
corresponding to the *decay element* strategy are orders of magnitude better than those of the unpreconditioned systems.

For the multiquadric the results, although still good, are somewhat surprisingly not as impressive as those for the thin-plate spline. The *decay element* approximate cardinal function took 55 iterations at 10,000 centres to reduce the MSR to less than 10^{-12} . In this example only 5,335 out of 10,000 centres satisfied condition (2.19) whereas the corresponding example with the thin-plate spline resulted in 9,113 centres qualifying. This finding was unexpected as the number of conditions for the multiquadric is less than for the thin-plate spline and therefore the dimension of the set of coefficients that satisfies condition (2.18) is larger for each subproblem. The smoothness property of the thin-plate spline could well explain why the *decay element* technique was so successful for it.

In Section 2.4 we showed how small the setup cost would be for the *pure local*

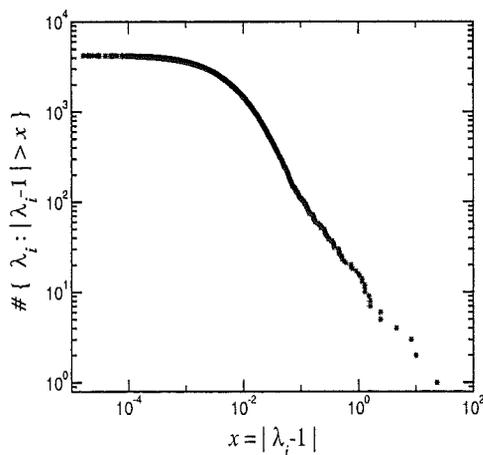


(a) Plot of $\#\{\lambda_i : |\lambda_i - 1| > x\}$ against x , for all eigenvalues.

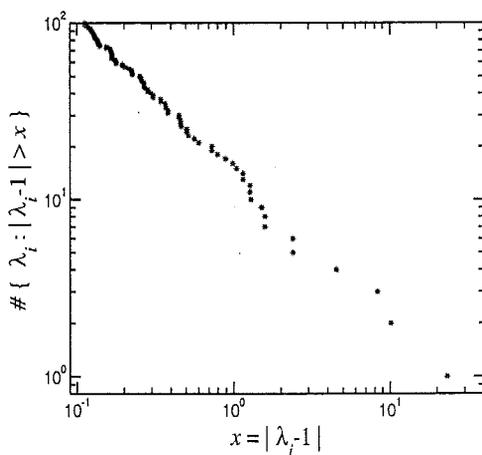


(b) Plot of $\#\{\lambda_i : |\lambda_i - 1| > x\}$ against x , for the 100 eigenvalues furthest from 1.

Figure 2.7: Eigenvalues of the preconditioned interpolation matrix for the *local centres and special points* approximate cardinal function. The ϕ function used is the thin-plate spline and the nodes of interpolation are 4225 random points in $[0, 1]^2$.



(a) Plot of $\#\{\lambda_i : |\lambda_i - 1| > x\}$ against x , for all eigenvalues.



(b) Plot of $\#\{\lambda_i : |\lambda_i - 1| > x\}$ against x , for the 100 eigenvalues furthest from 1.

Figure 2.8: Eigenvalues of the preconditioned interpolation matrix for the *pure local* approximate cardinal function. The ϕ function used is the thin-plate spline and the nodes of interpolation are 4225 random points in $[0, 1]^2$.

approximate cardinal functions. Unfortunately, forming the *decay element* approximate cardinal functions requires solving constrained least squares problems, and is more expensive. However the cost is not prohibitive for N greater than about 1500 when compared to the $\mathcal{O}(N^3)$ cost of direct solution. A somewhat cheaper way of finding a (different) decay element is to solve the interpolation subproblem and then force the solution to be orthogonal to the subspace spanned by the conditions in (2.16) and (2.18). This method requires only 16 iterations for convergence to 10^{-12} MSR at 10,000 centres and has a setup cost per element of $\beta^3/6 + \mathcal{O}(\beta^2 + p^2\beta)$ where p is the number of decay conditions. In situations where the setup cost is high relative to the cost of solving the system by direct methods this decay element may be more effective.

Number of centres	Approximate cardinal function strategy	Iteration count to specified MSR error		Condition number in specified norm		
		$< 10^{-6}$	$< 10^{-12}$	$\ \cdot\ _1$	$\ \cdot\ _2$	$\ \cdot\ _\infty$
289	Unpreconditioned	24	103	2.756(7)	4.005(6)	2.756(7)
	50 pure local	5	8	4.690(4)	1.464(3)	1.256(3)
	41 local + 9 special	2	5	1.164(2)	5.721(0)	1.563(1)
	50 local + 9 special	2	5	4.595(1)	3.651(0)	1.075(1)
	50 decay, $\mu = 0.5$	2	5	4.555(1)	3.330(0)	8.111(0)
1089	Unpreconditioned	23	145	2.059(9)	2,753(8)	2.059(9)
	50 pure local	14	18	9.551(7)	6.359(5)	2.242(5)
	41 local + 9 special	6	11	2.297(4)	1.818(2)	3.311(2)
	50 local + 9 special	3	7	2.303(4)	1.523(2)	1.522(2)
	50 decay, $\mu = 0.5$	3	6	2.422(4)	1.411(2)	6.904(1)
4225	Unpreconditioned	23	>150	2.082(10)	2.605(9)	2.082(10)
	50 pure local	32	41	1.103(9)	2.381(6)	1.369(6)
	41 local + 9 special	46	55	1.629(8)	1.040(6)	2.209(6)
	50 local + 9 special	9	16	4.228(5)	1.640(3)	2.496(3)
	50 decay, $\mu = 0.5$	5	9	8.148(5)	2.025(3)	6.494(2)
10000	50 pure local	61	73	-	-	-
	41 local + 9 special	107	122	-	-	-
	50 local + 9 special	33	45	-	-	-
	50 decay, $\mu = 0.5$	7	14	-	-	-

Table 2.2: Results of numerical experiments using the approximate cardinal functions discussed in this chapter for the thin-plate spline in 2 dimensions. Four different random sets in $[0, 1]^2$ are used as centres of interpolation.

Number of centres	Approximate cardinal function strategy	Iteration count to specified MSR error		Condition number in specified norm		
		$< 10^{-6}$	$< 10^{-12}$	$\ \cdot\ _1$	$\ \cdot\ _2$	$\ \cdot\ _\infty$
289	Unpreconditioned	22	145	3.160(8)	1.506(8)	3.160(8)
	50 pure local	7	11	2.261(5)	3.185(3)	4.134(3)
	41 local + 9 special	4	8	1.434(4)	2.639(2)	2.021(2)
	50 local + 9 special	3	6	3.773(3)	5.537(1)	6.480(1)
	50 decay, $\mu = 0.5$	4	8	4.105(3)	5.742(1)	6.108(1)
1089	Unpreconditioned	20	>150	4.284(9)	2.154(9)	4.284(9)
	50 pure local	10	17	1.856(8)	8.125(5)	2.328(5)
	41 local + 9 special	13	20	8.914(6)	5.234(4)	5.243(4)
	50 local + 9 special	6	11	2.604(5)	1.826(3)	1.335(3)
	50 decay, $\mu = 0.5$	9	15	4.099(5)	2.995(3)	1.886(3)
4225	Unpreconditioned	21	>150	8.544(10)	3.734(10)	8.544(10)
	50 pure local	18	29	6.827(9)	1.390(7)	5.542(6)
	41 local + 9 special	26	39	1.556(7)	4.071(4)	5.460(4)
	50 local + 9 special	16	24	1.784(7)	6.781(4)	8.054(4)
	50 decay, $\mu = 0.5$	19	28	1.436(7)	4.369(4)	3.630(4)
10000	50 pure local	22	42	-	-	-
	41 local + 9 special	56	78	-	-	-
	50 local + 9 special	32	43	-	-	-
	50 decay, $\mu = 0.5$	43	55	-	-	-

Table 2.3: Results of numerical experiments using the approximate cardinal functions discussed in this chapter for the multiquadric in 2 dimensions. Four different random sets in $[0, 1]^2$ are used as centres of interpolation. The parameter c was chosen as $1/\sqrt{N}$ where N is the number of centres.

Chapter 3

Fast Kriging

3.1 Introduction

Kriging is a surface fitting method which models spatial processes, $Z(x)$, in d dimensions. Areas where it has been applied include geophysics, hydrology, meteorology, mining engineering and bathymetry, see for example Cressie [25]. Kriging approximates the spatial process, $Z(x)$, under the assumption

$$Z(x) = \mu(x) + \kappa(x).$$

Here $\mu(x)$, the large scale variation, or the trend, of the process is an unknown element of some linear space \mathcal{P} of functions. For the purposes of the discussion in this chapter $\kappa(x)$ is a zero mean intrinsically stationary random process of order zero. However, the Kriging setting is well known to extend to random processes of order k in which case one uses a generalised covariance. These assumptions on the variability of $Z(x)$ are described in detail in Cressie [26, §2.2.1 and §3.4]. The Kriging approximation is the linear unbiased estimator which minimizes the mean-squared prediction error

$$E\left((Z(x) - s(x))^2\right), \tag{3.1}$$

for each $x \in \mathcal{D} \subset \mathcal{R}^d$, where $Z(x)$ is the actual value of the surface and $s(x)$ is the Kriged value obtained using the linear estimator

$$s(x) = \sum_{i=1}^N \lambda_i z_i. \quad (3.2)$$

Here, $\{z_1, \dots, z_N\}$ are the observed values at the corresponding spatial locations $\{x_1, \dots, x_N\}$ and $\lambda = \lambda(x) = \{\lambda_1, \dots, \lambda_N\} \in R^N$ is the vector of coefficients to be determined. To ensure the estimator is unbiased these coefficients are subject to the constraints

$$\sum_{j=1}^N \lambda_j q(x_j) = q(x), \quad \text{for all } q \in \mathcal{P}. \quad (3.3)$$

Let $m = \dim(\mathcal{P})$ and $\{q_1, \dots, q_m\}$ be some basis for \mathcal{P} then the constraints (3.3) can be written

$$\sum_{j=1}^N \lambda_j q_i(x_j) = q_i(x), \quad \text{for all } i = 1, \dots, m. \quad (3.4)$$

Often \mathcal{P} will be π_k^d , the space of polynomials of degree k in d variables. Then the constraints are the same as those found in the usual formulation of the radial basis function (RBF) interpolation equations. In the RBF setting they are interpreted as conditions which take away the extra degrees of freedom added by introducing the polynomial part. In the Kriging context they are conditions that make the Kriging estimator unbiased. When appropriate conditions (see equation (3.5)) on $Z(x)$ are met, the Kriging estimator is as accurate as possible at the point x . The estimator in this case is called a best linear unbiased estimator. It results in an estimator (3.2) that can be written in the RBF-like form

$$s(\cdot) = \sum_j \alpha_j \Phi(\cdot - x_j) + \sum_{j=1}^m \gamma_j q_j(\cdot).$$

Furthermore, the system (3.14) and (3.16) that would be solved to find the coefficients in the above “dual” formulation of the Kriging surface, are identical with those of the usual formulation of the RBF interpolation problem.

As can be seen above, and is well known, Kriging and RBF fitting are highly related. Indeed for fixed Φ and polynomial degree the fits produced by both methods

are identical. However, the motivations of the two methods, and the assumptions underlying them, are different. In RBF fitting, the choice of a quadratic smoothness penalty, also called an energy seminorm, will determine Φ and thus the RBF interpolant to be fitted. In the Kriging model, assumptions are made about the spatial correlation of the random process, and then Φ is determined experimentally from the data. Which method is more appropriate is application dependent. See [27, 28, 49, 50, 87] for discussions and numerical results comparing Kriging and splines.

Our discussion here applies to a type of Kriging known as universal Kriging with polynomial trend. This type of Kriging can be split into three parts: finding the semi-variogram (or basis function); fitting the Kriged surface; and forming the prediction error surface. In this chapter we concentrate on the second part, fitting the Kriged surface. Here fitting the Kriged surface means calculating its coefficients in RBF form.

The purpose of this chapter is to present a fast method for fitting the Kriged surface when the number of data points, N , is large, say greater than 4000, and assuming the semi-variogram is already known. Experiments indicate that this method requires $\mathcal{O}(N \log N)$ operations and $\mathcal{O}(N)$ storage, whereas direct fitting requires $\mathcal{O}(N^3)$ operations and $\mathcal{O}(N^2)$ storage. The method used is a development of a method which has been successful in the RBF context in the special cases of thin-plate spline and multiquadric Φ 's. Here it is used instead with several semi-variograms commonly used in Kriging, and in combination with a moment-based fast multiplication technique. Furthermore, the moment method, used as a fast evaluator, allows evaluation of the fitted surface at an incremental cost of $\mathcal{O}(1)$ operations per extra point.

The layout of the chapter is as follows. Section 2 outlines the universal Kriging procedure. Section 3 describes a fast method based on preconditioned GMRES iteration, and moment method multiplication, for forming the Kriged surface. Section 4 presents numerical results using random data. Section 5 discusses forming

the prediction error surface and Section 6 gives numerical results for a geophysical data set.

We alert the reader that there are other iterative methods for fast solution of RBF interpolation systems that may also be useful in the Kriging context. We mention in particular an improved version of the method described in [12], which is currently under development.

3.2 Surface fitting by universal Kriging

In this section we outline some of the theory of Kriging, and the “dual” form of the Kriging equations.

Universal Kriging with polynomial trend is a form of Kriging that is appropriate if the trend in the data can be approximated by an unknown polynomial of degree k , see Cressie [26, §3.4]. This type of Kriging can be used when $Z(x)$ satisfies,

$$\begin{aligned} \mu &\in \pi_k^d, \\ E(Z(x)) &= \mu(x), \text{ for all } x \in D, \\ \text{var}(Z(x+h) - Z(x)) &= 2\Phi(h) \text{ for all } x, x+h \in D. \end{aligned} \tag{3.5}$$

The function $\Phi(h)$ is referred to as the semi-variogram and $2\Phi(h)$ as the variogram. If the semi-variogram is radial, i.e. $\Phi(h) = \phi(\|h\|)$, then $Z(x)$ is called isotropic, otherwise it is anisotropic. Often a simple linear transformation of the spatial locations can give a system which is isotropic in the new data.

The numerical results in this chapter are only for the isotropic case although the method can easily be applied in the anisotropic setting. Due to microscale variation and measurement error $\Phi(h)$ is often discontinuous at the origin. If this is the case we can write

$$\Phi(h) = \begin{cases} 0, & \|h\| = 0, \\ c_0 + f(h), & \|h\| \neq 0, \end{cases} \tag{3.6}$$

where $f(h)$ is not necessarily radial but is continuous at 0 with $f(0) = 0$. The equations specifying an universal Kriging surface are solvable whenever the semi-

variogram, Φ , is strictly conditionally negative definite of order $k+1$ (SCND- $(k+1)$). See for example [12] for the definition of SCND- $(k+1)$ and a proof of solvability. In this chapter we consider semi-variograms in \mathcal{R}^2 of type (3.6).

In the remainder of this section we develop the well known equations for finding the coefficients, $\lambda = \lambda(x)$, of the Kriged surface. A good source for these and other Kriging equations is [26]. Expanding the square term in equation (3.1), using the estimator (3.2), and using the constraint from (3.4) that $\sum_{j=1}^N \lambda_j = 1$, we obtain

$$\begin{aligned}
(Z(x) - s(x))^2 &= \left(Z(x) - \sum_i \lambda_i Z_i \right)^2, \\
&= \sum_i \lambda_i Z(x)^2 - 2 \sum_i \lambda_i Z_i Z(x) + \left(\sum_i \lambda_i Z_i \right)^2, \\
&= \sum_i \lambda_i (Z_i - Z(x))^2 - \sum_i \lambda_i Z_i^2 + \left(\sum_i \lambda_i Z_i \right)^2, \\
&= \sum_i \lambda_i (Z_i - Z(x))^2 - \sum_j \sum_i \lambda_j \lambda_i Z_i^2 + \sum_j \sum_i \lambda_j \lambda_i Z_j Z_i, \\
&= \sum_i \lambda_i (Z_i - Z(x))^2 - \frac{1}{2} \sum_j \sum_i \lambda_j \lambda_i (Z_i - Z_j)^2. \tag{3.7}
\end{aligned}$$

Now using equations (3.3), (3.5) and (3.7) in (3.1) the following expression can be obtained for the prediction error

$$\begin{aligned}
E\left((Z(x) - s(x))^2 \right) &= E\left(\sum_i \lambda_i (Z_i - Z(x))^2 - \frac{1}{2} \sum_j \sum_i \lambda_j \lambda_i (Z_i - Z_j)^2 \right), \\
&= 2 \sum_i \lambda_i \Phi(x_i - x) - \sum_j \sum_i \lambda_i \lambda_j \Phi(x_i - x_j) \\
&\quad + \sum_i \lambda_i (\mu(x_i) - \mu(x))^2 - \frac{1}{2} \sum_i \sum_j \lambda_i \lambda_j (\mu(x_i) - \mu(x_j))^2, \\
&= 2 \sum_i \lambda_i \Phi(x_i - x) - \sum_j \sum_i \lambda_i \lambda_j \Phi(x_i - x_j). \tag{3.8}
\end{aligned}$$

The coefficients $\lambda = \lambda(x) \in \mathcal{R}^N$ are found by solving a constrained minimization

problem of the form

$$\begin{aligned} \min_{\lambda} E\left((Z(x) - s(x))^2\right), \\ \text{subject to } \sum_i \lambda_i q_j(x_i) = q_j(x), \quad j = 1, \dots, m. \end{aligned} \quad (3.9)$$

If we write $c_j(\lambda) = \sum_i \lambda_i q_j(x_i) - q_j(x)$ then the Lagrangian of this equality constrained problem can be written

$$l(\lambda, \nu) = E\left((Z(x) - s(x))^2\right) - 2 \sum_{j=1}^m \nu_j c_j(\lambda), \quad (3.10)$$

where $\nu = (\nu_1, \dots, \nu_m)^T$ are Lagrange multipliers. The first order necessary conditions for a solution of (3.9) are then

$$\nabla_{\lambda} l(\lambda, \nu) = 0, \text{ and } \nabla_{\nu} l(\lambda, \nu) = 0. \quad (3.11)$$

Substituting (3.8) into the expression (3.10) we find that at a minimum

$$\begin{aligned} \sum_j \lambda_j \Phi(x_j - x_i) - \Phi(x_i - x) + \sum_{k=1}^m \nu_k q_k(x_i) = 0, \quad i = 1 \dots N, \\ \sum_j \lambda_j q_i(x_j) = q_i(x), \quad i = 1, \dots, m, \end{aligned} \quad (3.12)$$

which can be written in matrix form as

$$B_{\Phi} \begin{bmatrix} \lambda \\ \nu \end{bmatrix} = \begin{bmatrix} g \\ c \end{bmatrix}, \quad (3.13)$$

where

$$B_{\Phi} = \begin{bmatrix} A_{\Phi} & Q \\ Q^T & 0 \end{bmatrix}, \quad (3.14)$$

$$(A_{\Phi})_{ij} = \Phi(x_i - x_j), \quad i, j = 1, \dots, N,$$

$$Q_{ij} = q_j(x_i), \quad i = 1, \dots, N, \quad j = 1, \dots, m,$$

$$g = (\Phi(x - x_1), \dots, \Phi(x - x_N))^T,$$

and

$$c = (q_1(x), \dots, q_m(x))^T.$$

Now, g is a function of the evaluation point x , so finding the surface will involve solving (3.13) for each evaluation point. This requires $\mathcal{O}(N^2)$ operations assuming N is small enough that we can form and store some suitable factorization of B_Φ , for example a $P^T LU$ or a QR factorization. Of course it will often be the case that N is too large for factorization of B_Φ to be practical. It is common practice [49, 50, 80] to find $s(x)$ using a subset consisting of the closest β points to x^\dagger . This approximate method involves something between $\mathcal{O}(\beta^2)$ and $\mathcal{O}(\beta^3)$ operations per point. Unfortunately, using this approach, minor discontinuities can occur at evaluation points where the subsets of local indices change. Also, prediction errors and confidence intervals are larger. The following “dual” Kriging equations [80, 26, 61, 62] suggest an alternative method to form the prediction surface. From (3.2) and (3.13) we obtain

$$s(x) = [z^T \ 0] \begin{bmatrix} \lambda \\ \nu \end{bmatrix} = [z^T \ 0] B_\Phi^{-1} \begin{bmatrix} g \\ c \end{bmatrix}. \quad (3.15)$$

Now B_Φ^{-1} is symmetric so we can solve

$$B_\Phi \begin{bmatrix} \alpha \\ \gamma \end{bmatrix} = \begin{bmatrix} z \\ 0 \end{bmatrix}, \quad (3.16)$$

for $[\alpha^T \ \gamma^T]^T$ to get RBF-like coefficients that do not depend on x . Equation (3.15) then becomes

$$s(x) = [\alpha^T \ \gamma^T] \begin{bmatrix} g \\ c \end{bmatrix} = \sum_i \alpha_i \Phi(x - x_i) + \sum_{j=1}^m \gamma_j q_j(x). \quad (3.17)$$

Note that (3.17) and (3.16) are the usual expressions for an RBF, and the usual system used to determine the coefficients of an interpolatory RBF, respectively.

The possibility of using fast evaluation techniques similar to those developed for radial basis functions [11, 14] to reduce the operation count for finding the

[†]For some data sets and some evaluation points the set of β points may not be unique. If this is the case then the set should be found in a consistent manner.

“dual” Kriging coefficients has been previously mentioned in passing by some authors including [49, 50, 17].

3.3 A fast fitting method for large N

In this section we present a method for forming the Kriged surface using the RBF-like coefficients in (3.17). The method involves a combination of a preconditioner, a fast matrix-vector multiplication code appropriate for the function Φ , and the GMRES iterative algorithm for solving linear systems. The numerical experiments of later sections will show that for several typical semi-variograms this method can determine the RBF-like coefficients of the Kriged surface in $\mathcal{O}(N \log N)$ operations.

A similar approach has been successfully applied to solving RBF interpolation problems in the special cases of thin-plate spline and multiquadric basic functions in [6]. In the current chapter we use instead several semi-variograms Φ common in the Kriging context. Another difference here is that the matrix-vector products arising in the GMRES iteration are performed with a fast moment method [15, 5]. This is a method which computes the action of the matrix $A_\Phi = (\Phi(x_i - x_j))$ on a vector of coefficients in approximately $\mathcal{O}(N)$ operations and using only $\mathcal{O}(N)$ storage. Direct calculation of the same matrix-vector product requires $\mathcal{O}(N^2)$ operations and also $\mathcal{O}(N^2)$ storage, making large problems intractable. In contrast to the more established fast multipole like methods, the fast moment method is highly adaptive to changes of basic function Φ . Changing to a different Φ requires only the coding of a one or two line function for the slow evaluation of Φ . This adaptivity makes the fast moment method well suited for the Kriging application where many different Φ 's occur. The fast moment method can also be used to reduce the incremental cost of evaluation of the fitted surface, s , at a single additional point, x , to $\mathcal{O}(1)$ operations instead of $\mathcal{O}(N)$ operations.

The heuristic underlying the approximate Lagrange function preconditioning used is to change from the basis of functions, $\Phi_j = \Phi(\cdot - x_j)$, to a basis of functions

ψ_j , where $\psi_j(x_i) \approx \delta_{ij}$. We write the new basis element, ψ_j , in the form

$$\psi_j(\cdot) = \sum_{k=1}^m c_{jk} q_k(\cdot) + \sum_{i=1}^N \theta_{ji} \Phi(\cdot - x_i). \quad (3.18)$$

Each ψ_j is constructed so that $\{\theta_{ji}\}_{i=1}^N$ is orthogonal to polynomials in the sense of equation (3.4).

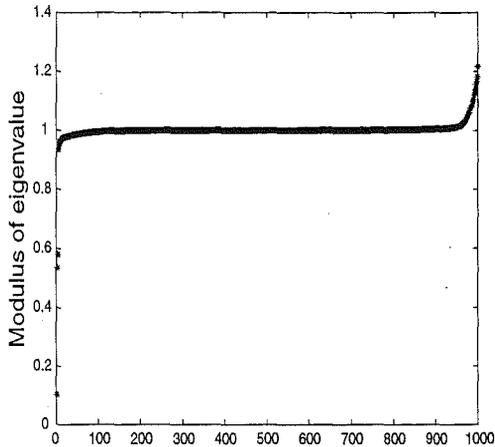
This change of basis leads to the new system of fitting equations

$$A_\psi y = z, \quad (3.19)$$

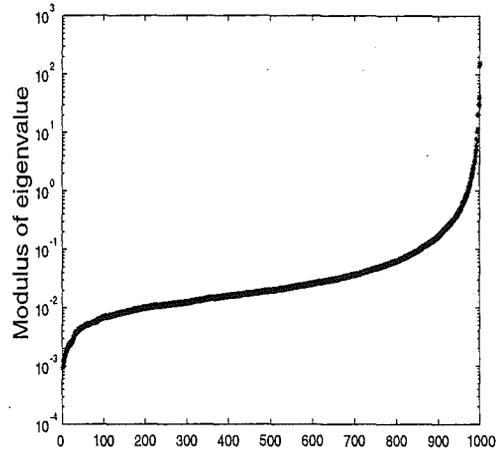
where $(A_\psi)_{ij} = \psi_j(x_i)$ and y are coefficients of the fitted function in terms of this new basis. This is the linear system to which GMRES is applied. In practice A_ψ is never formed, as it is too expensive to store and use. Rather its action on a vector is calculated using the fast moment method in $\mathcal{O}(N)$ operations. To ensure fast convergence of the GMRES iteration we aim to choose the ψ_j elements so that A_ψ has eigenvalues that are clustered within a small relative radius. Such clustering is well known [53, 72, 18] to guarantee fast convergence of the GMRES iteration. Our ψ elements are constructed so that the ψ interpolation matrix has ones on the main diagonal and is close to zero everywhere else. Figure (3.1) shows the resulting clustered eigenvalues of A_ψ and the non-clustered eigenvalues of A_ϕ for the basis function $\phi(h) = h$.

Different strategies for finding the θ_{ji} 's can greatly affect the performance of A_ψ in GMRES. If the θ_{ji} 's were chosen so that $\psi_j(x_j) = 1$ and $\psi_j(x_i) = 0$, $i \neq j$ then $A_\psi = I$ and GMRES would converge in one iteration. However, forming the ψ_j elements in this way would require the solution of N full size linear systems. Clearly this is not practical. To reduce the computation we restrict the number of non-zero $\{\theta_{ji}\}_{i=1}^N$ to $\beta \ll N$ for each j . We define the set \mathcal{S}_j to be the set of β indices i such that θ_{ji} is possibly non-zero. Thus the new basis element is a sum over the indices in \mathcal{S}_j , or specifically

$$\psi_j(\cdot) = \sum_{k=1}^m c_{jk} q_k(\cdot) + \sum_{i \in \mathcal{S}_j} \theta_{ji} \Phi(\cdot - x_i). \quad (3.20)$$



(a) Preconditioned matrix, A_ψ . Each \mathcal{S}_j is fifty local points and nine special points.



(b) Unpreconditioned matrix, B_ϕ .

Figure 3.1: Eigenvalue plots for the basis function $\phi(h) = h$. The spatial data is one thousand random points within the domain $[0, 1]^2$.

Various strategies for finding the θ_{ji} 's are given in [6]. Some of these strategies are Φ specific and are unsuitable for this chapter. The strategy we use for most of this chapter is to pick the index set \mathcal{S}_j as a set of indices of β closest points to x_j , together with the indices of a small number, τ , of special points. Then we form ψ_j by requiring that $\psi_j(x_j) = 1$ and $\psi_j(x_i) = 0$, $i \in \mathcal{S}_j$, $i \neq j$. This strategy yields an approximate cardinal function, called a *local centres and special points approximate cardinal function* in [6]. The idea of including special points in \mathcal{S}_j is to force ψ_j to zero at various points widely scattered throughout the domain. It is then expected that ψ_j will be close to zero near these points. If we were fitting within the square $[0, 1]^2$ then a suitable choice of four special points would be the centres closest to $(0, 0)$, $(0, 1)$, $(1, 0)$, and $(1, 1)$ respectively. Figure (3.2) shows a single new basis element formed with this strategy for the semi-variogram $\phi(h) = 1 - \exp(-h)$. It is clear from the graph that in this case the strategy has been extremely successful

and the corresponding column of the matrix A_ψ will be very close to the j^{th} column of the identity.

Forming ψ_j involves solving a $(\beta + \tau + m) \times (\beta + \tau + m)$ system of interpolation equations. These systems can be converted to be symmetric positive definite using the method of [12]. Solving via Cholesky then takes $\mathcal{O}((\beta + \tau + m)^3/6)$ operations [†]

To reduce the operation counts in this setup we often use the same Cholesky decomposition for more than one ψ element. In the examples given in table 3.1 about $0.1N$ Cholesky decompositions were formed and in table 3.2 about $0.3N$ Cholesky decompositions were formed. The preconditioning is slightly less effective using this technique due to the unbalanced nature of the subsets. In our experience the increase in GMRES iterations is only small and so forming ψ elements in this way is worthwhile.

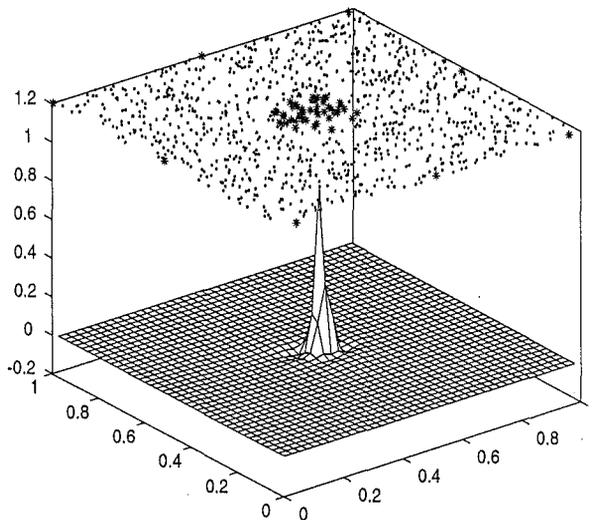


Figure 3.2: A new basis element based on fifty local points and nine special points out of a data set of one thousand points. The dots above the graph indicate the spatial data points and the asterix's indicate the location of points in \mathcal{S}_j .

Now to find the RBF-like coefficients $[\alpha^T \ \gamma^T]$ we simply convert from the good

[†]We count operations in *old flops*, each old flop being one multiplication or division, together with one addition or subtraction, plus a little indexing.

ψ basis to the bad Φ basis. Letting T be an $N \times N$ matrix with $T_{ij} = \theta_{ji}$ and C an $m \times N$ matrix with $C_{ij} = c_{ji}$ then equation (3.19) can be written

$$[A_{\Phi} \ Q] \begin{bmatrix} T \\ C \end{bmatrix} y = z. \quad (3.21)$$

The coefficients $[\alpha^T \ \gamma^T]$ can easily be found from

$$\begin{bmatrix} \alpha \\ \gamma \end{bmatrix} = \begin{bmatrix} T \\ C \end{bmatrix} y.$$

Exploiting the sparsity of T allows the above conversion from coefficients with respect to the good basis, to coefficients with respect to the bad basis, to be performed in $(\beta + \tau + m)N$ operations.

3.4 Numerical Results

This section presents numerical results generated with an initial implementation of the method of this chapter. The method was applied to a selection of random data sets with various typical variograms valid in \mathcal{R}^2 and the computation times recorded. All the numerical examples are for the important special case of ordinary Kriging when the degree of the polynomial trend, k , is 0. The experiments were conducted on a Sun Ultrasparc machine. Section 3.6 below describes an application of the method to a non simulated data set, an electromagnetic survey. Note that if r is the residual vector then the mean square residual (MSR) is $r^T r / N$.

All the semi-variograms considered are isotropic and of the form (3.6). The number of iterations for convergence of GMRES varies greatly depending on the preconditioning strategy used and also on the initial basis function. (3.22)-(3.25) specify $\phi(h)$ for $h > 0$, and we assume throughout that $\phi(0) = 0$.

$$\text{exponential} \quad \phi(h) = c_0 + c_1(1 - \exp(-c_2h)) \quad (3.22)$$

$$\text{linear} \quad \phi(h) = c_0 + c_1h \quad (3.23)$$

$$\text{power} \quad \phi(h) = c_0 + c_1h^{c_2}, \quad c_2 \in [0, 2) \quad (3.24)$$

$$\text{rational quadratic} \quad \phi(h) = c_0 + c_1 \frac{h^2}{1 + h^2/c_2} \quad (3.25)$$

All these functions are valid semi-variograms (SCND1 functions) provided $c_0, c_1, c_2 \geq 0$ and also $c_2 < 2$ in the case of (3.24) [26].

The results in tables 3.1 and 3.2 show that our approach can be successfully used for moderate and large problems. Many of the numerical experiments in these tables would not be solvable using standard direct methods. We have demonstrated that for twenty thousand centres the RBF-like coefficients can be found in less than two minutes in most cases. In the case of a linear variogram with each \mathcal{S}_j consisting of one hundred closest points and nine special points the solution is found in 100.4 seconds. The choice of the size of \mathcal{S}_j is a tradeoff between minimizing the setup time and minimizing the time for convergence in GMRES. Increasing β would decrease the time for GMRES to converge but increase the setup time. For smaller values of N , say between one thousand and ten thousand, then having $\beta = 100$ rather than $\beta = 50$ has no clear advantage with respect to computation time. Once N is about twenty thousand a clear advantage can be seen in having larger subsets. However, larger subsets means an increase in storage requirements. The current computation times will be improved with algorithmic changes within both the moment method and the setup codes.

3.5 Prediction errors

One advantage of Kriging over RBF fitting is that Kriging is designed to minimize the prediction error at a point. To find the prediction error at a point x we solve the linear system in equation (3.13) for each evaluation point and then evaluate using

equation (3.8). Clearly this is undesirable when the number of data points is large. In the RBF literature functions of the form of the right hand side of (3.8) are referred to as power functions. The fundamental properties of the power functions are now well known (see e.g. Wu and Schaback [92], Light and Wayne [55] and Powell [68]). The following result may be proved by an argument analogous to that of Light and Wayne [55, Lemma 2.7].

Theorem: Let $X_N = \{x_1, \dots, x_N\}$ and $X_B = \{x_1, \dots, x_B\}$ be finite subsets of \mathcal{R}^d with $X_B \subset X_N$. For a given $x \in \mathcal{R}^d$, let $s_N(x)$ and $s_B(x)$ be the Kriged values formed using observations at the points of X_N and X_B respectively. Then

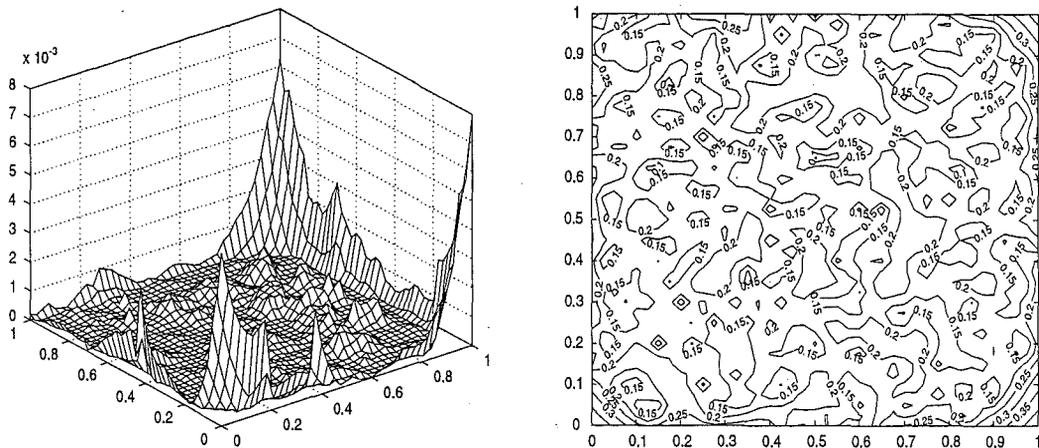
$$E\left((Z(x) - s_N(x))^2\right) \leq E\left((Z(x) - s_B(x))^2\right). \quad (3.26)$$

This result may be summarized as saying that using additional data points will not increase, and indeed is likely to decrease, the prediction error. Thus giving further motivation for fitting surfaces using all N points. If we fit using all N points and then form prediction errors based on local subsets of data we will obtain estimated prediction errors that are slightly greater than the actual prediction errors.

Figure 3.3 gives an example of prediction errors for a data set of 300 points uniformly distributed in $[0, 1]^2$. This example shows the prediction error surface corresponding to using all points is only slightly below the prediction error surface corresponding to using local subsets of 30 points.

3.6 Geophysical application

This section describes the application of the method of this chapter to a geophysical data set. The data considered is an electromagnetic survey consisting of measurements of radiation due to decay of uranium at 18824 spatial locations. The fitted surfaces are given in Figure 3.5. We are grateful to the Australian Geological Survey Organisation for the use of this data. We have scaled the data so it is contained in the region $[0, 0.5] \times [0, 1]$. The measurements of uranium radioactivity were taken



(a) Relative difference between the standard errors based on the whole data set and standard errors based on subsets of thirty points.

(b) Standard error surface where each point is found using the entire data set.

Figure 3.3: Standard error surfaces for the basis function $\phi(h) = 1 - \exp(-h)$. The spatial data is three hundred random points within the domain $[0, 1]^2$.

by an aeroplane flying in transects across the domain. Assuming stationarity for this example we fitted a variogram by standard parametric techniques (see [93, 59] for more detail on fitting variograms). The experimental semi-variogram and fitted semi-variogram can be seen in Figure 3.4.

The fitted equation is

$$\phi(h) = 0.014 + 0.025(1 - \exp(-19h)), \quad \|h\| > 0, \quad (3.27)$$

which is a valid semi-variogram, that is an SCND1 function, in \mathcal{R}^2 .

Our preconditioning approach here was to use only nearby centres and no special points in forming the ψ functions. ψ elements formed in this way are called *pure local approximate cardinal functions* in [6]. Using this preconditioner we were able to make our subset size 30 for each new basis element. This decreased our setup time considerably. Convergence to MSR 10^{-6} and MSR 10^{-12} took 11 and 18 iterations

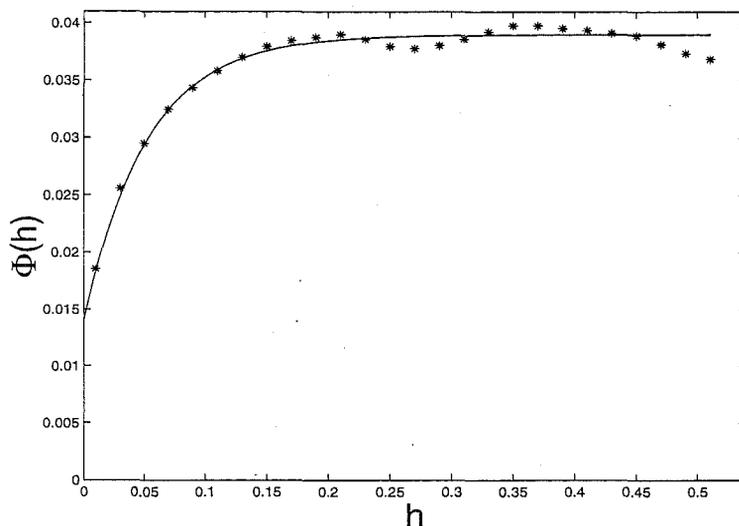
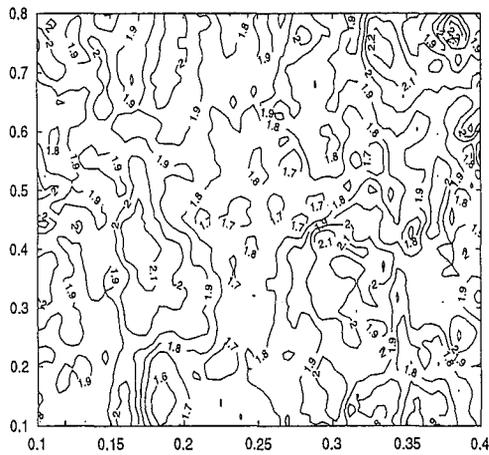


Figure 3.4: Experimental (*) and fitted (-) semi-variograms for the geophysical data set. The fitted semi-variogram is given by equation (3.27).

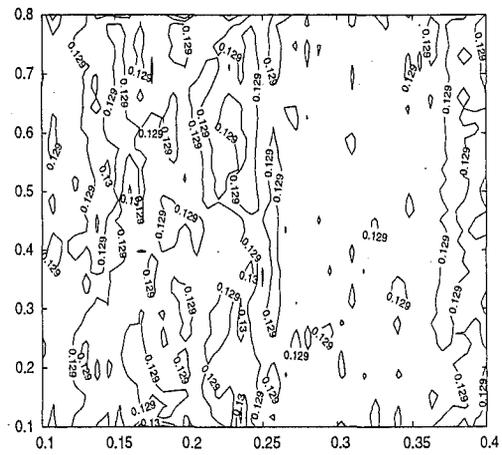
respectively. This compares favourably with the 17 and 23 iterations seen for the random data set of twenty thousand points in table 3.1. The total set up time was 11.8 seconds and GMRES iteration required a further 84.5 seconds for convergence of the 2-norm residual to 10^{-6} . Solving for the RBF coefficients by standard methods would take hours of computer time.

3.7 Discussion

We have presented a fast method for forming the fitted Kriging surface using RBF-like coefficients. In numerical experiments with the method fitting takes $\mathcal{O}(N \log N)$ operations and $\mathcal{O}(N)$ storage. Previously finding these coefficients would have required $\mathcal{O}(N^3)$ operations and $\mathcal{O}(N^2)$ storage, therefore making the use of “global” Kriging surfaces impossible for large data sets. The numerical experiments presented show the effectiveness of this new method for a number of isotropic variogram functions using simulated data, and also for a geophysical data set. It is now possible to find these fitting coefficients for a data set containing 20,000 points in less than 2



(a) Kriged surface of radiation due to decay of uranium over the scaled area.



(b) Standard error surface where each point is found using subsets of size 40

Figure 3.5: The Kriged surface and the prediction error surface for the geophysical uranium data. The semi-variogram is given by equation (3.27).

minutes. Planned improvements in several aspects of the numerical code should extend the size of data set that can be handled to hundreds of thousands, or millions, of points in the near future.

Number of centres	Basic function, ϕ	Iteration count to specified MSR error		Time in seconds for specified task	
		$< 10^{-6}$	$< 10^{-12}$	Setup	GMRES
4000	exponential	8	11	6.0	7.0
	linear	8	11	4.9	6.5
	power, $c_2=1/2$	9	11	5.8	6.9
	power, $c_2=3/2$	9	12	7.8	8.2
	rational quadratic	41	51	5.2	27.0
10000	exponential	13	17	17.2	26.0
	linear	13	16	14.0	24.4
	power, $c_2=1/2$	10	14	16.9	23.5
	power, $c_2=3/2$	10	15	22.5	26.1
	rational quadratic	56	67	14.6	93.1
20000	exponential	17	23	33.2	77.8
	linear	17	23	26.4	74.0
	power, $c_2=1/2$	16	21	32.1	71.7
	power, $c_2=3/2$	19	23	43.1	82.5
	rational quadratic	42	61	27.4	203.5

Table 3.1: Results of numerical experiments for Φ functions given by equations (3.22)-(3.25). The preconditioning elements consist of one hundred closest points and nine special points. GMRES timings are for 2-norm convergence to residual $< 10^{-6}$.

Number of centres	Basic function, ϕ	Iteration count to specified MSR error		Time in seconds for specified task	
		$< 10^{-6}$	$< 10^{-12}$	Setup	GMRES
4000	exponential	13	17	3.6	9.3
	linear	13	17	2.7	9.4
	power, $c_2=1/2$	9	13	3.5	7.4
	power, $c_2=3/2$	10	15	5.2	9.5
	rational quadratic	31	46	2.8	24.3
10000	exponential	23	28	9.1	38.3
	linear	23	28	6.8	36.7
	power, $c_2=1/2$	15	22	8.7	31.3
	power, $c_2=3/2$	18	25	12.7	37.6
	rational quadratic	46	64	7.2	85.5
20000	exponential	29	35	18.2	107.2
	linear	29	34	13.5	101.3
	power, $c_2=1/2$	25	32	17.6	99.3
	power, $c_2=3/2$	29	35	25.5	112.6
	rational quadratic	34	62	14.3	203.1

Table 3.2: Results of numerical experiments for Φ functions given by equations (3.22)-(3.25). The preconditioning elements consist of fifty closest points and nine special points. GMRES timings are for 2-norm convergence to residual $< 10^{-6}$.

Chapter 4

On the boundary over distance preconditioner

4.1 Introduction

Let $\Phi : \mathcal{R}^d \rightarrow \mathcal{R}$, $X = \{x_1, \dots, x_N\}$ be a set of N distinct points in \mathcal{R}^d and f be a real valued function which we can evaluate at least at the x_i 's. Define

$$S_{\Phi, X} = \left\{ \begin{array}{l} g : g = \sum_{i=1}^N \lambda_i \Phi(\cdot - x_i) \\ \text{where } \sum_{j=1}^N \lambda_j q(x_j) = 0, \text{ for all } q \in \pi_1^d \end{array} \right\}. \quad (4.1)$$

We consider the problem of finding an element s of $S_{\Phi, X} + \pi_1^d$ satisfying the interpolation conditions

$$s(x_i) = f(x_i), \quad \text{for all } x_i \in X. \quad (4.2)$$

Assume Φ is strictly conditionally positive definite of order 2 and X is unisolvent for π_1^d . Then there is a unique element of $S_{\Phi, X} + \pi_1^d$ satisfying the interpolation conditions (4.2). This setting includes popular choices of the basic function such as the thin-plate spline, $\Phi(\cdot) = |\cdot|^2 \log |\cdot|$, and the negative of the ordinary multiquadric, $\Phi(\cdot) = -\sqrt{|\cdot|^2 + c^2}$. In this chapter we consider various ways of formulating the interpolation problem, showing in particular that a certain inexpensive change of basis can dramatically improve its conditioning.

The usual way to formulate this problem is in terms of the functions $\{\Phi(\cdot - x_i)\}$ and some basis $\{p_0, p_1, \dots, p_d\}$ for π_1^d . Then the interpolation conditions together with the side conditions taking away the extra degrees of freedom introduced by the polynomial part can be written as

$$A\lambda + Pc = f \quad \text{and} \quad P^T\lambda = 0, \quad (4.3)$$

where

$$A_{ij} = \Phi(x_i - x_j), \quad P_{ij} = p_j(x_i),$$

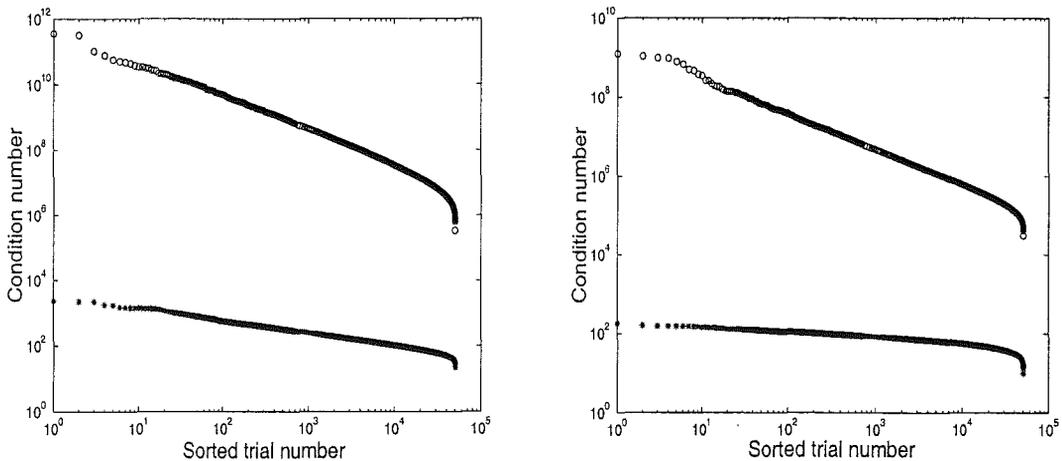
and $f = [f(x_1), \dots, f(x_N)]^T$. It is well known [31, 64, 77] that the matrix

$$A_\Phi = \begin{bmatrix} A & P \\ P^T & O \end{bmatrix}, \quad (4.4)$$

of this usual formulation is frequently badly conditioned, even when the number of nodes is small. Indeed many authors have commented on the numerical difficulties that solving this system presents [77, 38, 31, 64]. However, frequently in numerical analysis a change of basis, or other reformulation, can make a highly intractable problem tractable. Indeed, in the case of the RBF interpolation equations changing to the basis of cardinal functions would result in the interpolation matrix becoming the identity and the system being perfectly conditioned and trivial to solve. Unfortunately, finding the cardinal RBFs would be more computationally expensive than solving the system itself. Hence, our goal is to find less expensive but still highly effective preconditioners for the interpolation system.

In this chapter we establish properties of a preconditioning method for the RBF interpolation equations which was first presented in Sibson and Stone [77]. In the following section we give a detailed account of the preconditioning method. In Section 4.3 we prove that the construction produces an $N \times (N - 3)$ matrix Q whose columns are orthogonal to P , and which is of full rank whenever the nodes X are unisolvent for π_1^2 . In Section 4.4 we show that for certain functions Φ , B is a homogeneous function of scale. Hence, its condition number, and the relative clustering

of its eigenvalues, are independent of scale. Section 4.5 contains a proof that the elements B_{ij} decay like $|x_i - x_j|^{-\kappa}$ when $|x_i - x_j|$ is large. For the multiquadric κ is three and for the thin-plate spline κ is two. Sections 4.6 and 4.7 contain numerical results for different SCPD2 basic functions over a range of data sets and scales. These numerical results show that using this inexpensive $\mathcal{O}(N \log N)$ flop preconditioner and variants of it, dramatically improves the conditioning of RBF interpolation problems. See Figure 4.1 below. Finally, Section 4.8 discusses the effects of roundoff error when using a fast technique to compute the product of the preconditioned matrix and a vector.



(a) Multiquadric basic function.

(b) Thin-plate spline basic function.

Figure 4.1: Sorted 2-norm condition numbers of the unpreconditioned matrices, A_Φ , (top) and of the preconditioned matrices, S , (bottom) for fifty thousand random data sets of size one hundred.

4.2 A preconditioning method

A general approach to preconditioning interpolation problems with SCPD2 basic functions in \mathcal{R}^2 [12, 77] is to choose Q as any $N \times (N - 3)$ matrix whose columns

are orthogonal to P and has rank $N - 3$. Letting $\lambda = Q\mu$ and premultiplying (4.3) by Q^T gives the new system to be solved for μ , or equivalently λ ,

$$B\mu = Q^T f \quad \text{where} \quad B = Q^T A Q. \quad (4.5)$$

The three polynomial coefficients can then be found by a small subsidiary calculation.

In this section we present the boundary over distance method of Sibson and Stone [77] for constructing the matrix Q . We will prove in the subsequent section that Q has full rank and its columns are orthogonal to P for any set of distinct nodes $X = \{x_1, \dots, x_N\} \subset \mathcal{R}^2$, which are unisolvent for π_1^2 . These properties of Q are well known (see e.g. [12, 77]) to imply that the matrix of the preconditioned system $B = Q^T A Q$ is positive definite. The construction is appealing in that for “interior” points x_j of X it is local. That is, for such points the entries in the j -th column of Q depend only on the geometry of the nodes near x_j and not on any properties of nodes far away.

Choose W as a closed bounded convex polygonal region of \mathcal{R}^2 such that $X \subset W$. Suppose without loss of generality that $\{x_{N-2}, x_{N-1}, x_N\}$ is unisolvent for π_1^2 . We will refer to these points as special points. They are generally chosen so that they are well spread throughout W .

The region W is first divided into panels by intersecting a Voronoi diagram of the points of X with the region W . We denote this panelling of W by

$$V_W(X) = \bigcup_{i=1}^N V_i$$

where V_i is the Voronoi panel about the i th centre and is defined by

$$V_i = \{x \in W : |x - x_i| < |x - x_j|, \quad \text{for all } 1 \leq j \leq N \text{ with } j \neq i\}.$$

Recall that the locus of points equidistant from two fixed points is the perpendicular bisector of the segment connecting the points. It follows that each Voronoi region is polygonal. Associated with a panel V_i are its edges. These are a finite number of

distinct closed line segments of non-zero length. They are the boundaries between different Voronoi panels, or between a Voronoi panel and W^C . The collection of all edges of all the Voronoi panels will be denoted by \mathcal{E} .

Definition 4.2.1. *Two polygonal regions of \mathcal{R}^2 will be said to be strongly contiguous if they have a common boundary of non-zero length.*

Definition 4.2.2. *Two Voronoi regions V_i and V_j will be said to be C-related if there is a sequence*

$$\{V_i, V_{\ell_1}, V_{\ell_2}, \dots, V_{\ell_m}, V_j\},$$

in which all adjacent pairs are strongly contiguous.

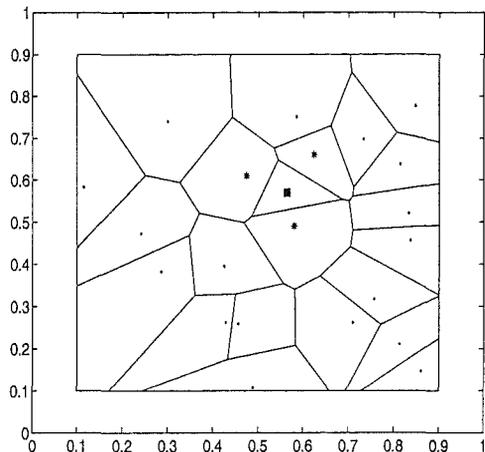
Loosely speaking V_i and V_j are C-related if they are connected by a chain of strongly contiguous pairs. C-related is an equivalence relation on the set $\{V_i\}_{i=1}^{N-3}$ of Voronoi regions of non-special points. Therefore it breaks this set into a finite number of nonempty equivalence classes $\{\mathcal{G}_l : 1 \leq l \leq k\}$. Figure 4.2 illustrates the different equivalence classes of strongly contiguous sets of Voronoi panels arising from different choices of the three special points.

Lemma 4.2.3. *Let \mathcal{G}_ℓ be any of the equivalence classes above. Then there is at least one Voronoi region V_i in \mathcal{G}_ℓ which is strongly contiguous to either W^C or one of $\{V_{N-2}, V_{N-1}, V_N\}$.*

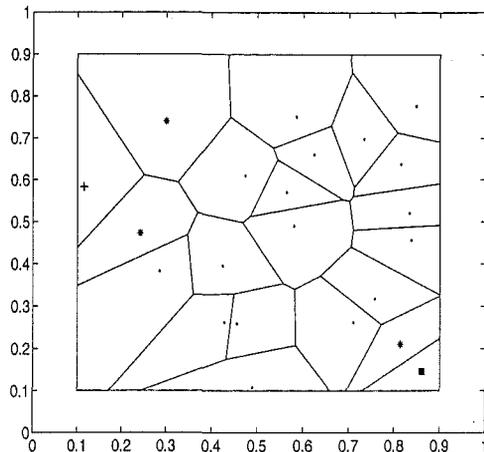
Proof. Consider

$$T = \bigcup_{i:V_i \in \mathcal{G}_\ell} \bar{V}_i$$

This union is a closed bounded connected polygonal set whose boundary can be written as the union of some of the line segments from \mathcal{E} . Recall in particular that all these line segments have non-zero length. Pick one line segment $\langle a, b \rangle$ from the boundary of T . Since it forms part of the boundary of T on one side of it lies a Voronoi region V_i from \mathcal{G}_ℓ . On the other side lies either W^C or another Voronoi region V_j . In the first case the Lemma is proven. Consider the second case. If



(a) A configuration of special points (*) leading to two strongly contiguous sets of Voronoi panels.



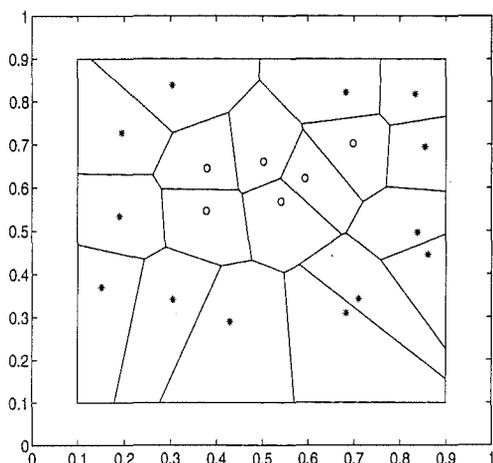
(b) A configuration of special points (*) leading to three strongly contiguous sets of Voronoi panels.

Figure 4.2: Configurations of points where E (see equation (4.10)) is reducible. In each figure centres in the same strongly contiguous regions share the same symbol, and *'s denote special points.

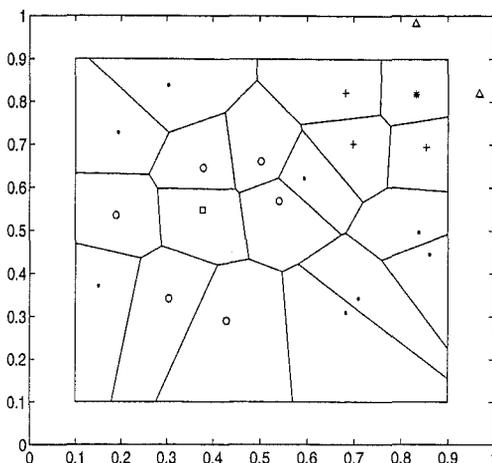
$1 \leq j \leq N - 3$ then V_i is strongly contiguous to V_j . Consequently, $V_j \in \mathcal{G}_\ell$. This contradicts $\langle a, b \rangle$ being on the boundary of T . Hence, $N - 2 \leq j \leq N$ and the Lemma follows. \square

We now detail the construction of the $N \times (N - 3)$ matrix Q using boundary over distance weights. Note that because most elements of Q are zero sparse storage of Q requires only $\mathcal{O}(N)$ memory. A non-special point from $\mathcal{S} = \{x_i : 1 \leq i \leq N - 3\}$ which is strongly contiguous to W^C will be called a *Voronoi external point*. Define $V_E(X)$ as the set of indices of all Voronoi external points. All other points are referred to as *Voronoi internal points*. The corresponding indices are $V_I(X) = \{1, \dots, N - 3\} - V_E(X)$. See Figure 4.3 for examples of Voronoi internal and external points.

We first consider forming a column of Q for an index, j , such that $j \in V_I(X)$. In



(a) Internal centres (o) and external centres (*) of X .



(b) The neighbours (o) of an internal centre (□) and neighbours (+) of an external centre (*). Artificial points corresponding to the external centre * are denoted by Δ .

Figure 4.3: Voronoi panelling of a set of twenty data points in the region $W = [0.1, 0.9]^2$.

this case the panel V_j shares non-trivial edges only with other Voronoi panels and not with W^C . The column is formed using boundary over distance weights, found from the Voronoi diagram. For $j \in V_I(X)$ the boundary over distance weight r_{ij} is

$$r_{ij} = \frac{b(x_i, x_j)}{|x_i - x_j|}, \quad \text{for all } V_i \text{ strongly contiguous to } V_j, \quad (4.6)$$

where $b(x_i, x_j)$ is the length of the boundary between V_i and V_j . For other values of $i \neq j$, r_{ij} is set to zero. In order that column j of Q is orthogonal to constants the diagonal element r_{jj} is specified as

$$r_{jj} = - \sum_{i \neq j} r_{ij}. \quad (4.7)$$

Finally, the j th column of R is scaled by dividing by the area of V_j to obtain the j th column of Q . Note that the column is by construction diagonally dominant, but

not strictly so.

If $j \in V_E(X)$ then V_j is strongly contiguous to the complement of W , W^C . The boundary segment corresponds to a Voronoi edge between x_j and an artificial point, the reflection of x_j in the boundary (see Figure 4.3(b)). The reflected point, \hat{x}_j , can be written as a linear combination of the special points, i.e.,

$$\hat{x}_j = \lambda_n x_n + \lambda_{n-1} x_{n-1} + \lambda_{n-2} x_{n-2}, \quad (4.8)$$

where $\lambda_n + \lambda_{n-1} + \lambda_{n-2} = 1$. If V_j has k edges with W^C then k reflected points $\{\hat{x}_j^1, \dots, \hat{x}_j^k\}$ are required. Associated with each reflected point, \hat{x}_j^a , are the coefficients $\{\lambda_n^a, \lambda_{n-1}^a, \lambda_{n-2}^a\}$. The boundary over distance weights for \hat{x}_j^a are partitioned amongst the special points to obtain for all $j \in V_E(X)$ and $i \neq j$

$$r_{ij} = \begin{cases} \frac{b(x_i, x_j)}{|x_i - x_j|}, & V_i \text{ strongly contiguous to } V_j, \\ \sum_{l=1}^k \lambda_i^l \frac{b(\hat{x}_j^l, x_j)}{|\hat{x}_j^l - x_j|}, & i \in \{n, n-1, n-2\}. \end{cases} \quad (4.9)$$

Of course, V_j could be strongly contiguous with a Voronoi panel associated with a special point. If this is the case $r_{ij} = \frac{b(x_i, x_j)}{|x_i - x_j|} + \sum_{l=1}^k \lambda_i^l \frac{b(\hat{x}_j^l, x_j)}{|\hat{x}_j^l - x_j|}$. Again, for other values of $i \neq j$, r_{ij} is set to zero. Finally r_{jj} is specified as in (4.7) and column j of Q is defined as column j of R scaled by dividing by the area of V_j .

Partition Q as

$$Q = \begin{bmatrix} E \\ F \end{bmatrix}, \quad (4.10)$$

where E is $(N-3) \times (N-3)$. Thus E results from interactions between non-special points, and F those between special and non-special points. Note in the construction above that for $1 \leq i, j \leq N-3$, e_{ij} is non-zero if and only if V_i is strongly contiguous to V_j . Furthermore, note that E is necessarily column diagonally dominant, with strict dominance in column j whenever V_j is strongly contiguous to the Voronoi region of a special point, or to W^C .

Relabelling if necessary we can assume the indices of the Voronoi regions in each of the equivalence classes \mathcal{G}_i form a contiguous subset of $\{1, \dots, N-3\}$. Similarly,

we can also assume that the indices corresponding to any \mathcal{G}_i precede those corresponding to \mathcal{G}_{i+1} . Furthermore, by construction if $i \neq j$ none of the regions in \mathcal{G}_i is strongly contiguous with a region in \mathcal{G}_j . Thus, corresponding entries in the matrix E constructed using boundary over distance weights and artificial points are zero. That is, E is block diagonal with the square matrix E_{ii} on the main diagonal corresponding to the equivalence class of Voronoi regions \mathcal{G}_i . More precisely, if k is the number of equivalence classes then Q will have the form

$$Q = \begin{bmatrix} E_{11} & O & \cdots & O \\ O & E_{22} & \cdots & O \\ \vdots & \vdots & \ddots & \vdots \\ O & O & \cdots & E_{kk} \\ F_1 & F_2 & \cdots & F_k \end{bmatrix}. \quad (4.11)$$

4.3 Properties of the matrix Q

In this section we establish the fundamental properties of the matrix Q of (4.11). Namely that it is of full rank and that its columns are orthogonal to those of P .

Definition 4.3.1. For $m \geq 2$, an $m \times m$ matrix K is irreducible if there does not exist an $m \times m$ permutation matrix P such that

$$PKP^T = \begin{bmatrix} M_{11} & M_{12} \\ 0 & M_{22} \end{bmatrix},$$

where M_{11} is $r \times r$, M_{22} is $(m-r) \times (m-r)$, and $1 \leq r < m$.

The following result is well known, see for example Varga [81].

Theorem 4.3.2. Suppose the square matrix K is irreducible and row (column) diagonally dominant with strict row (column) diagonal dominance in at least one row (column). Then K is invertible.

The proof of the following result relies on the concept of directed graphs from graph theory. The directed graph, $G(K)$, of a matrix K , is a graph such that there

is a directed arc between vertices y_i and y_j of the graph if and only if the entry k_{ij} of the matrix is non-zero.

Definition 4.3.3. *A directed graph is strongly connected if for any pair of points y_i and y_j there exists a directed path $\overrightarrow{y_i y_{i_1}}, \overrightarrow{y_{i_1} y_{i_2}}, \dots, \overrightarrow{y_{i_{k-1}} y_j}$, connecting y_i to y_j .*

Lemma 4.3.4. *(Theorem 1.6 of Varga [81]) A square complex matrix K is irreducible if and only if its directed graph $G(K)$ is strongly connected.*

Lemma 4.3.5. *Let X be a finite set of distinct points unisolvent for π_1^2 . Let E_{ii} be one of the square blocks from the diagonal of Q constructed in the previous section. Then E_{ii} is invertible.*

Proof. From the construction E_{ii} is column diagonally dominant. Furthermore, by Lemma 4.2.3 the diagonal dominance is strict for at least one column of E_{ii} . From the definition of the equivalence relation C-related there is a chain of strongly contiguous pairs of Voronoi regions, connecting any two Voronoi regions in \mathcal{G}_i . This implies the corresponding entries in E_{ii} are non-zero and hence from Lemma 4.3.4 E_{ii} is irreducible. It follows from Theorem 4.3.2 that E_{ii} is invertible. \square

Theorem 4.3.6. *The columns of the matrix Q described in Section 4.2 are orthogonal to P .*

Proof. It suffices to show that the matrix R (which is Q before column scaling) is orthogonal to P . The proof of this theorem for interior nodes is taken from Christ et.al. [23, Theorem 1]. Let $j \in V_I(X)$. If we let v be a constant vector then from the divergence theorem

$$0 = \int_{V_j} \nabla \cdot v \, dt = \int v \cdot n \, dS,$$

where n is a normal vector and S is the boundary of V_j .

Each boundary segment of V_j is associated with a contiguous Voronoi panel. Let \mathcal{S}_j be the set of indices of such contiguous panels. For $i \in \mathcal{S}_j$ the length of the

boundary between V_j and V_i is given by $b(x_j, x_i)$. From the properties of a Voronoi diagram an outward normal to this boundary is $\frac{x_i - x_j}{|x_i - x_j|}$. Integrating over each of these boundaries separately gives

$$0 = \int v \cdot n \, dS = v \cdot \sum_{i \in \mathcal{S}_j} \frac{b(x_i, x_j)}{|x_i - x_j|} (x_i - x_j). \quad (4.12)$$

Because v is any constant vector we obtain

$$\sum_{i \in \mathcal{S}_j} \frac{b(x_i, x_j)}{|x_i - x_j|} (x_i - x_j) = 0, \quad (4.13)$$

and the result follows from $r_{jj} = -\sum_{i \in \mathcal{S}_j} r_{ij}$. Note the interesting alternative interpretation of (4.13) as an expression for x_j as a convex combination of its neighbours.

For $j \in V_E(X)$ we have at least one boundary segment of V_j which corresponds to a boundary between V_j and W^C . In the case of only one boundary segment between V_j and W^C we introduce the corresponding artificial point $\hat{x}_j = \lambda_n x_n + \lambda_{n-1} x_{n-1} + \lambda_{n-2} x_{n-2}$. Then

$$\begin{aligned} & v \cdot \sum_{i \neq j} r_{ij} (x_j - x_i) \\ &= v \cdot \left(\sum_{i \in \mathcal{S}_j} r_{ij} (x_j - x_i) + r_{n,j} (x_j - x_n) + r_{n-1,j} (x_j - x_{n-1}) + r_{n-2,j} (x_j - x_{n-2}) \right), \\ &= v \cdot \left(\sum_{i \in \mathcal{S}_j} r_{ij} (x_j - x_i) + \frac{b(\hat{x}_j, x_j)}{|\hat{x}_j - x_j|} (x_j - \hat{x}_j) \right), \\ &= \int v \cdot n \, dS = 0, \end{aligned} \quad (4.14)$$

where the last line follows from (4.12) and because $\frac{x_j - \hat{x}_j}{|\hat{x}_j - x_j|}$ is a normal vector to the boundary between V_j and W^C . The result again follows from $r_{jj} = -\sum_{i \neq j} r_{ij}$. If V_j has more than one boundary with W^C then the proof can be easily extended by using more artificial points in (4.14). \square

Theorem 4.3.7. *Let X be a set of distinct points unisolvent for π_1^2 . Let Q be formed by the construction in Section 4.2 and $A_{ij} = \Phi(x_i - x_j)$ where Φ is strictly conditionally positive definite of order 2. Then $B = Q^T A Q$ is positive definite.*

Proof. From Lemma 4.3.5 each of the matrices E_{ii} occurring in the block partitioning of Q given in Equation (4.11) is invertible. Hence Q has full rank. Also from Theorem 4.3.6 the columns of Q are orthogonal to the columns of P . Let μ be any non-zero vector in \mathcal{R}^{N-3} , and define $\lambda = Q\mu$. Then $\lambda \neq 0$, $P^T\lambda = P^TQ\mu = 0$, and $\mu^TB\mu = \mu^TQ^TAQ\mu = \lambda^TA\lambda$. Hence, by the definition of strictly conditionally positive definite, $\mu^TB\mu > 0$ whenever $\mu \neq 0$ and B is symmetric positive definite. \square

4.4 Scaleability

In this section we show that for certain functions Φ the new interpolation matrix $B = Q^TAQ$ is a homogeneous function of scale. Thus its condition number, and the relative spread of its eigenvalues, are scale independent. If the interpolation matrix is not a homogeneous function of scale then the condition number can change dramatically over different scales. This is important for fitting methods such as those described in [6] and [12], where solutions to systems on many different scales are required.

Lemma 4.4.1. *Given $X = \{x_1, \dots, x_N\}$ unisolvent with respect to π_{k-1}^d . Let $\{r_1, \dots, r_N\}$ and $\{s_1, \dots, s_N\}$ satisfy $\sum_{j=1}^N r_j q(x_j) = 0$, and $\sum_{j=1}^N s_j q(x_j) = 0$, for all $q \in \pi_{k-1}^d$. Define $T : C(\mathcal{R}^d \times \mathcal{R}^d) \rightarrow \mathcal{R}$ by*

$$Tg = \sum_{i,j=1}^N r_i s_j g(x_i, x_j), \quad (4.15)$$

for $g \in C(\mathcal{R}^d \times \mathcal{R}^d)$. Then T annihilates all functions g of the form $g(x, y) = p(x-y)$ with $p \in \pi_{2k-1}^d$.

Proof. Following [12, Lemma 2.1] we let $p(x) = p_\alpha(x) = x^\alpha$, where $x \in \mathcal{R}^d$ and $\alpha \in Z_+^d$ with $|\alpha| < 2k$. From the binomial theorem we have

$$g(x, y) = p(x-y) = \sum_{0 \leq \beta \leq \alpha} a_\beta x^{\alpha-\beta} y^\beta = \sum_{0 \leq \beta \leq \alpha} a_\beta p_{\alpha-\beta}(x) p_\beta(y), \quad x, y \in \mathcal{R}^d,$$

Define $g_{\alpha\beta} = p_{\alpha-\beta}(x)p_{\beta}(y)$, then

$$Tp_{\alpha} = \sum_{0 \leq \beta \leq \alpha} a_{\beta} Tg_{\alpha\beta}.$$

Now, from (4.15)

$$\begin{aligned} Tg_{\alpha\beta} &= \sum_{i,j=1}^N r_i s_j g_{\alpha\beta}(x_i, x_j), \\ &= \sum_{i,j} r_i s_j p_{\alpha-\beta}(x_i) p_{\beta}(x_j), \\ &= \left(\sum_i r_i p_{\alpha-\beta}(x_i) \right) \left(\sum_j s_j p_{\beta}(x_j) \right). \end{aligned} \tag{4.16}$$

From the hypothesis, and because either $|\beta| \leq k-1$ or $|\alpha-\beta| \leq k-1$, one of the bracketed expressions is zero. Hence Tg is zero. \square

Theorem 4.4.2. *Let the symmetric function $\Phi \in C(\mathcal{R}^d \times \mathcal{R}^d)$ be such that $\Phi(hx, hy) = h^{\gamma} \Phi(x, y) + p_h(x-y)$ for all $h > 0$ and $x, y \in \mathcal{R}^d$, where $\gamma \in \mathcal{R}$ and $p_h \in \pi_{2k-1}^d$. Let $X = \{x_1, \dots, x_N\}$ be a unisolvent set of points with respect to π_{k-1}^d and let $\{r_1, \dots, r_N\}$ and $\{s_1, \dots, s_N\}$ be as in Lemma 4.4.1. Define the functional $T_h \Phi$ by*

$$T_h \Phi = \sum_{i,j=1}^N r_i s_j \Phi(hx_i, hx_j),$$

and write T for T_1 . Then for $h > 0$, $T_h \Phi = h^{\gamma} T \Phi$.

Proof. From the definition we have

$$\begin{aligned} T_h \Phi &= \sum_{i,j} r_i s_j \Phi(hx_i, hx_j), \\ &= \sum_{i,j} r_i s_j \{h^{\gamma} \Phi(x_i, x_j) + p_h(x_i - x_j)\}, \\ &= h^{\gamma} T \Phi + Tv, \end{aligned} \tag{4.17}$$

where $v(x, y) = p_h(x-y)$ for some $p_h \in \pi_{2k-1}^d$ and T is as in Lemma 4.4.1. From that lemma $Tv = 0$ and the Theorem follows. \square

In the following Theorem matrices with a subscript h are defined in the same way as the matrix without the subscript except with the point set hX instead of X .

Theorem 4.4.3. *Let $X = \{x_1, \dots, x_N\}$ be unisolvent with respect to π_{k-1}^d , and P be defined by $P_{ij} = p_j(x_i)$, where p_1, \dots, p_l is a basis for π_{k-1}^d . Let Q_h be any $N \times (N - \dim(\pi_{k-1}^d))$ matrix which depends homogeneously on the scale parameter h such that $Q_h = h^\nu Q$, and $P^T Q = 0$.*

Then if $\Phi(hx, hy) = h^\gamma \Phi(x, y) + p_h(x - y)$, $h > 0$ for some $p_h \in \pi_{2k-1}^d$, B_h is a homogeneous function of h . Specifically

$$B_h = h^{2\nu+\gamma} B.$$

Proof. Let r_j be the j th column of Q . Then from Theorem 4.4.2 and the condition on Φ we have,

$$r_j^T A_h r_i = h^\gamma r_j^T A r_i,$$

and so

$$Q^T A_h Q = h^\gamma Q^T A Q = h^\gamma B. \quad (4.18)$$

From the conditions on Q and (4.18)

$$\begin{aligned} B_h &= Q_h^T A_h Q_h, \\ &= h^{2\nu} Q^T A_h Q, \\ &= h^{2\nu+\gamma} B. \end{aligned}$$

□

Remark 4.4.4. *If Φ is the basic function $\Phi(\cdot) = (-1)^k |\cdot|^{2(k-1)} \log |\cdot|$ then from the proof of Corollary 2.3 in [12] we have,*

$$\Phi(hx, hy) = h^{2(k-1)} \Phi(x, y) + p_h(x - y), \quad (4.19)$$

where $p_h \in \pi_{2(k-1)}^d$. So the thin-plate spline basic function satisfies the condition on Φ in Theorem 4.4.3.

Corollary 4.4.5. *Let Φ be strictly conditionally positive definite of order 2 and such that $\Phi(hx, hy) = h^\gamma \Phi(x, y) + p_h(x - y)$, $h > 0$ with $p_h \in \pi_3^2$. Then the interpolation matrix, B_h , produced by the algorithm in Section 4.2 is a homogeneous function of scale.*

Proof. From Theorem 4.4.3 it is sufficient to show that $Q_h = h^\nu Q$, for some ν . The Voronoi diagram scales homogeneously hence $b(hx_i, hx_j) = hb(x_i, x_j)$. Also, the area of the panel associated with hx_i is h^2 times that of the panel associated with x_i . Therefore, we have for $Q_{ij} \neq 0$, $j \in V_I(X)$, $i \neq j$,

$$(Q_h)_{ij} = \frac{b(hx_i, hx_j)}{|h(x_i - x_j)|h^2 A(V_i)} = h^{-2} Q_{ij}.$$

Noticing that $\{\lambda_n, \lambda_{n-1}, \lambda_{n-2}\}$ in (4.8) are unchanged by scale gives $(Q_h)_{ij} = h^{-2} Q_{ij}$, $j \in V_E(X)$. \square

Theorem 4.4.6. *Let Φ be strictly conditionally positive definite of order 2 and such that $\Phi(hx, hy) = h^\gamma \Phi(x, y) + p_h(x - y)$, $h > 0$ with $p_h \in \pi_3^2$. Then the interpolation matrix, S , produced by scaling the matrix B so that $S = DBD$, where D is diagonal and $D_{ii} = 1/\sqrt{B_{ii}}$, is constant over all scales.*

Proof. From Corollary 4.4.5, $B_h = h^\theta B$, for some θ . So $d_{ii}^h = (b_{ii}^h)^{-\frac{1}{2}} = h^{-\frac{\theta}{2}} d_{ii}$ and $S_h = D_h B_h D_h = h^{-\theta} D B_h D = h^{-\theta} h^\theta D B D = S$. \square

4.5 Decay

The dramatic improvement in conditioning between the unpreconditioned matrix and the preconditioned matrix is due to the localisation of the preconditioner. Specifically, in this section we show that these local preconditioners have the property that $|B_{ij}| \approx \|x_i - x_j\|^{-\kappa}$ as $\|x_i - x_j\|$ grows, where κ is three for the multiquadric and two for the thin-plate spline. This decay means the interpolation matrix is “almost” diagonally dominant and thus better conditioned. In this section $|\cdot|$ applied to a multiindex means its 1-norm.

Definition 4.5.1. Let $X = \{x_1, \dots, x_{N+1}\} \subset \mathcal{R}^2$. The set $\mathcal{U}_X \subset \mathcal{R}^{N+1}$ consists of all $\beta \in \mathcal{R}^{N+1} \setminus \{0\}$ that satisfy

$$\sum_{i=1}^{N+1} \beta_i q(x_i) = 0, \text{ for all } q \in \pi_1^2. \quad (4.20)$$

In this section we denote the set \mathcal{S}_j to be all the indices i such that V_i is strongly contiguous with V_j . Note that $j \notin \mathcal{S}_j$.

Lemma 4.5.2. Let x_j be an internal node of a Voronoi diagram. The area of a Voronoi polygon, V_j , about x_j , is given by

$$\frac{1}{4} \sum_{i \in \mathcal{S}_j} \|x_j - x_i\| b(x_j, x_i),$$

where $b(x_j, x_i)$ is the length of the Voronoi boundary orthogonal to $x_j - x_i$.

Proof. A Voronoi polygon of x_j can be divided into triangles by line segments between x_j and the vertices of the polygon. The area of each of these triangles can easily be shown to be $\|x_j - x_i\| b(x_j, x_i)/4$. Summing these areas gives the result. \square

Lemma 4.5.3. Let x_j be an internal node and V_j the corresponding Voronoi panel. Then if $\beta_i = \frac{b(x_j, x_i)}{\|x_j - x_i\|}$, $i \in \mathcal{S}_j$,

$$\left| \sum_{i \in \mathcal{S}_j} \beta_i (x_j - x_i)^\alpha \right| \leq 4 \text{Area}(V_j), \text{ for } |\alpha| = 2. \quad (4.21)$$

Proof. Noticing that $|(x_j - x_i)^\alpha| \leq \|x_j - x_i\|^2$ for $|\alpha| = 2$ we have

$$\begin{aligned} \left| \sum_{i \in \mathcal{S}_j} \beta_i (x_j - x_i)^\alpha \right| &\leq \sum_{i \in \mathcal{S}_j} \beta_i |(x_j - x_i)^\alpha|, \\ &\leq \sum_{i \in \mathcal{S}_j} \beta_i \|x_j - x_i\|^2, \\ &= \sum_{i \in \mathcal{S}_j} \frac{b(x_j, x_i)}{\|x_j - x_i\|} \|x_j - x_i\|^2, \\ &= 4 \text{Area}(V_j), \end{aligned}$$

by Lemma 4.5.2. \square

Lemma 4.5.4. *Let $f(\cdot) := |\cdot|^{-k}$, $k > 0$ so that $f : \mathcal{R}^2 \setminus \{0\} \rightarrow \mathcal{R}$. Then if \mathcal{D}^α , $\alpha \in \mathcal{N}^2$, is the differential operator*

$$\frac{\partial^{\alpha_1 + \alpha_2}}{\partial \xi^{\alpha_1} \partial \eta^{\alpha_2}},$$

where $x = (\xi, \eta)$ we have

$$\mathcal{D}^\alpha f(\cdot) = \frac{P_\alpha(\cdot)}{|\cdot|^{k+2|\alpha|}}, \tag{4.22}$$

where P_α is a homogeneous polynomial of degree $|\alpha|$.

Proof. Let $|\alpha| = 1$, then

$$\begin{aligned} \mathcal{D}^\alpha f(x) &= \mathcal{D}^\alpha(|x|^{-k}), \\ &= \frac{-kx^\alpha}{|x|^{k+2}}, \end{aligned}$$

as required. Now assume that (4.22) holds for $|\alpha| = n$. If $|\gamma| = 1$ we obtain

$$\begin{aligned} \mathcal{D}^{\alpha+\gamma} f(x) &= \mathcal{D}^\gamma \left(\frac{P_\alpha(x)}{|x|^{k+2n}} \right), \\ &= \frac{\mathcal{D}^\gamma(P_\alpha(x))|x|^{k+2n} - \mathcal{D}^\gamma(|x|^{k+2n})P_\alpha(x)}{|x|^{2(k+2n)}}, \\ &= \frac{\hat{P}_{\alpha-1}(x)|x|^2 - (k+2n)P_\alpha(x)x^\gamma}{|x|^{k+2(n+1)}}. \end{aligned} \tag{4.23}$$

In (4.23) $\hat{P}_{\alpha-1} = \mathcal{D}^\gamma(P_\alpha(x))$ is a homogeneous polynomial of degree $n - 1$. The result follows from setting $P_{\alpha+\gamma}(x) = \hat{P}_{\alpha-1}(x)|x|^2 - (k+2n)P_\alpha(x)x^\gamma$. \square

Lemma 4.5.5. *Let $X = \{x_1, \dots, x_{N+1}\}$, $x_i = (\xi_i, \eta_i) \in \mathcal{R}^2 \setminus \{0\}$, be $N + 1$ distinct points contained in the circle $|\cdot - x| \leq H$, with $H < |x|$ and where $x = x_{N+1}$. If $\beta \in \mathcal{U}_X$, and $n, \alpha > 0$ then,*

$$\left| \sum_{i=1}^{N+1} \beta_i \frac{x_i^\alpha}{|x_i|^n} \right| \leq \frac{H^2 DE}{|x|^{n-|\alpha|+2}} + \mathcal{O}(|x|^{-(n-|\alpha|+3)}),$$

where $E = \sum_{i=1}^N |\beta_i|$ and D depends on α and n . Furthermore, if $\beta_i = \frac{b(x_i, x)}{\|x_i - x\|}$, $i = 1, \dots, N$ then

$$E = \frac{4\text{Area}(V)}{H^2},$$

where V is the Voronoi region about the point x .

Proof. Let $h_i = (\Delta\xi_i, \Delta\eta_i) = x - x_i$. Now we obtain via the binomial expansion

$$\begin{aligned}
(x - h_i)^\alpha &= \sum_{k=0}^{\alpha_1} \sum_{j=0}^{\alpha_2} \binom{\alpha_1}{k} \binom{\alpha_2}{j} \xi^k \eta^j (-\Delta\xi_i)^{\alpha_1-k} (-\Delta\eta_i)^{\alpha_2-j}, \\
&= \sum_{l=0}^{|\alpha|} \sum_{k=\max\{0, l-\alpha_2\}}^{\min\{l, \alpha_1\}} \binom{\alpha_1}{k} \binom{\alpha_2}{l-k} (-1)^{|\alpha|-l} \xi^k \eta^{l-k} \Delta\xi_i^{\alpha_1-k} \Delta\eta_i^{\alpha_2+k-l}, \\
&= \sum_{l=0}^{|\alpha|} P_l(x, h_i, \alpha),
\end{aligned} \tag{4.24}$$

where $P_l(x, h_i, \alpha)$ is a polynomial of degree l in x and $|\alpha| - l$ in h_i .

By a Taylor's expansion in h about x and using Lemma 4.5.4, or alternatively using that $(1 - 2xy + y^2)^{-n}$ is the generating function for a family of Gegenbauer polynomials [79, 4.7.23], we have

$$|x_i|^{-n} = |x - h_i|^{-n} = |x|^{-n} \sum_{m=0}^{\infty} \frac{Q_m(x, h_i, n)}{|x|^{2m}}, \quad |h_i| < |x|, \tag{4.25}$$

where in (4.25) $Q_m(x, h, n)$ is a polynomial of degree m in x and h .

Now by using $\sum_{i=1}^{N+1} \beta_i = 0$ from (4.20) and combining (4.24) and (4.25)

$$\begin{aligned}
\sum_{i=1}^{N+1} \beta_i \frac{x_i^\alpha}{|x_i|^n} &= \sum_{i=1}^N \beta_i \left(\frac{(x - h_i)^\alpha}{|x_i|^n} - \frac{x^\alpha}{|x|^n} \right), \\
&= \frac{1}{|x|^n} \sum_{i=1}^N \beta_i \left(\sum_{l=0}^{|\alpha|} P_l(x, h_i, \alpha) \sum_{m=0}^{\infty} \frac{Q_m(x, h_i, n)}{|x|^{2m}} \right) - \frac{1}{|x|^n} \sum_{i=1}^N \beta_i x^\alpha.
\end{aligned} \tag{4.26}$$

In the series on the left all terms that arise are of order $-n + |\alpha| - \gamma$ in x , $\gamma \geq 0$. Those of order $-n + |\alpha| - \gamma$ correspond to values of l and m such that $m - l = \gamma - |\alpha|$. The restrictions on m and l are as in (4.26). The lemma will be proved by showing that the terms corresponding to $\gamma = 0, 1$ vanish.

Firstly consider $\gamma = 0$; then $m = 0$ and $l = |\alpha|$ and the corresponding terms in (4.26) of order $-n + |\alpha|$ are

$$\frac{1}{|x|^n} \sum_{i=1}^N \beta_i \left(P_{|\alpha|}(x, h_i, \alpha) Q_0(x, h_i, n) - x^\alpha \right) = 0,$$

because $P_{|\alpha|}(x, h, \alpha) = x^\alpha$ and $Q_0(x, h, n) = 1$. For $\gamma = 1$; $(l, m) = (|\alpha|, 1)$ and $(|\alpha| - 1, 0)$. Expanding the functions P and Q for these values of (l, m) gives the terms

$$-\frac{1}{|x|^n} \sum_{i=1}^N \beta_i \left(\binom{\alpha_1}{\alpha_1 - 1} \xi^{\alpha_1 - 1} \eta^{\alpha_2} \Delta \xi_i + \binom{\alpha_2}{\alpha_2 - 1} \xi^{\alpha_1} \eta^{\alpha_2 - 1} \Delta \eta_i \right) = 0, \quad (4.27)$$

from (4.20). This leaves us with the most positive power of $|x|$ being $-(n - |\alpha| + 2)$ as required. These terms correspond to values of $(l, m) = (|\alpha| - 2, 0)$; $(|\alpha| - 1, 1)$; and $(|\alpha|, 2)$. The terms in (4.26) of exact order $|x|^{-(n - |\alpha| + 2)}$ are

$$\begin{aligned} & \frac{1}{|x|^4} \left(|x|^4 Q_0 P_{|\alpha| - 2} + |x|^2 Q_1 P_{|\alpha| - 1} + Q_2 P_{|\alpha|} \right) \\ &= \frac{x^\alpha}{|x|^4} \left(\xi^2 (\Delta \xi_i^2 c_1 + \Delta \eta_i^2 c_4) + \eta^2 (\Delta \xi_i^2 c_5 + \Delta \eta_i^2 c_2) \right. \\ & \quad + \xi \eta \Delta \xi_i \Delta \eta_i c_3 + \xi^{-1} \eta^3 \Delta \xi_i \Delta \eta_i c_6 + \xi^3 \eta^{-1} \Delta \xi_i \Delta \eta_i c_7 \\ & \quad \left. + \xi^{-2} \eta^4 \Delta \xi_i^2 c_8 + \xi^4 \eta^{-2} \Delta \eta_i^2 c_9 \right), \end{aligned} \quad (4.28)$$

where

$$\begin{aligned} c_1 &= -n\alpha_1 + \frac{1}{2}\alpha_1(\alpha_1 - 1), \\ c_2 &= -n\alpha_2 + \frac{1}{2}\alpha_2(\alpha_2 - 1), \\ c_3 &= n(1 - |\alpha|) + 2\alpha_1\alpha_2, \\ c_4 &= -n\left(\frac{1}{2} + \alpha_2\right) + \alpha_2(\alpha_2 - 1), \\ c_5 &= -n\left(\frac{1}{2} + \alpha_1\right) + \alpha_1(\alpha_1 - 1), \\ c_6 &= \alpha_1(\alpha_2 - n), \\ c_7 &= \alpha_2(\alpha_1 - n), \\ c_8 &= \frac{1}{2}\alpha_1(\alpha_1 - 1), \\ c_9 &= \frac{1}{2}\alpha_2(\alpha_2 - 1). \end{aligned}$$

By taking absolute values and noticing that $|x^\alpha| \leq |x|^{|\alpha|}$ we obtain from (4.28)

$$\begin{aligned} & \frac{1}{|x|^4} \left(|x|^4 Q_0 P_{|\alpha|-2} + |x|^2 Q_1 P_{|\alpha|-1} + Q_2 P_{|\alpha|-2} \right) \\ & \leq \frac{|x|^{|\alpha|+2}}{|x|^4} \left(\Delta \xi_i^2 d_1 + |\Delta \xi_i \Delta \eta_i| d_2 + \Delta \eta_i^2 d_3 \right), \\ & = \frac{1}{|x|^{-|\alpha|+2}} \left(\Delta \xi_i^2 d_1 + |\Delta \xi_i \Delta \eta_i| d_2 + \Delta \eta_i^2 d_3 \right), \end{aligned} \quad (4.29)$$

where $d_1 = |c_1| + |c_5| + |c_8|$, $d_2 = |c_3| + |c_6| + |c_7|$, and $d_3 = |c_2| + |c_4| + |c_9|$. Now substituting (4.29) into equation (4.26) we obtain

$$\begin{aligned} \left| \sum_{i=1}^{N+1} \beta_i \frac{x_i^\alpha}{|x_i|^n} \right| & \leq \frac{1}{|x|^{n-|\alpha|+2}} \sum_{i=1}^N |\beta_i| \left(\Delta \xi_i^2 d_1 + |\Delta \xi_i \Delta \eta_i| d_2 + \Delta \eta_i^2 d_3 \right) + \mathcal{O}(|x|^{-(n-|\alpha|+3)}), \\ & \leq \frac{H^2 D E}{|x|^{n-|\alpha|+2}} + \mathcal{O}(|x|^{-(n-|\alpha|+3)}), \end{aligned} \quad (4.30)$$

where $E = \sum_{i=1}^N |\beta_i|$ and $D = d_1 + d_2 + d_3$. If we take β_i to be boundary over distance weights then using Lemma 4.5.3 gives the bound

$$\left| \sum_{i=1}^{N+1} \beta_i \frac{x_i^\alpha}{|x_i|^n} \right| \leq \frac{4 \text{Area}(V) D}{|x|^{n-|\alpha|+2}} + \mathcal{O}(|x|^{-(n-|\alpha|+3)}).$$

□

The following two theorems show that for local preconditioners the entries of the preconditioned matrix, $(B)_{kl} = \sum_{i,j=1}^N r_{jk} r_{il} \Phi(x_i - x_j)$ will decay as $|x_k - x_l|$ gets large.

Theorem 4.5.6. *Let Φ be the multiquadric and let $X = \{x_1, \dots, x_{N+1}\}$, $x_i = (\xi_i, \eta_i) \in \mathcal{R}^2$, be $N + 1$ distinct points contained in the circle $|\cdot - x| \leq H_1 < |x|$ where $x = x_{N+1}$. Also let $Y = \{y_1, \dots, y_{N+1}\}$, $y_i = (s_i, t_i) \in \mathcal{R}^2$, be $N + 1$ distinct points contained in the circle $|\cdot| < H_2$ with $y_{N+1} = 0$. Then if $\beta \in \mathcal{U}_X$, $\gamma \in \mathcal{U}_Y$ and $|x| > H_1 + \sqrt{H_2^2 + c^2}$,*

$$\left| \sum_{i=1}^{N+1} \sum_{j=1}^{N+1} \gamma_i \beta_j \Phi(x_j - y_i) \right| \leq \frac{20 H_1^2 H_2^2 E_1 E_2}{|x|^3} + \mathcal{O}(|x|^{-4}),$$

where $E_1 = \sum_{i=1}^N |\beta_i|$ and $E_2 = \sum_{i=1}^N |\gamma_i|$. Furthermore, if β and γ are boundary over distance weights

$$\left| \sum_{i=1}^{N+1} \sum_{j=1}^{N+1} \gamma_i \beta_j \Phi(x_j - y_i) \right| \leq \frac{640 \text{Area}(V_1) \text{Area}(V_2)}{|x|^3} + \mathcal{O}(|x|^{-4}),$$

where V_1 is the Voronoi region around x_{N+1} and V_2 is the Voronoi region around y_{N+1} .

Proof. If $|x|$ is big enough we can approximate Φ by a far field expansion. For the multiquadric this was given in [7]. The far field expansion about zero for the multiquadrics $\Phi(\cdot - y_i)$ are valid for $|\cdot| > \sqrt{H_2^2 + c^2}$. Since by hypothesis $|x| > H_1 + \sqrt{H_2^2 + c^2}$ then $\min_{x \in X, y \in Y} |x - y| > \sqrt{H_2^2 + c^2}$. Now due to the sets X and Y

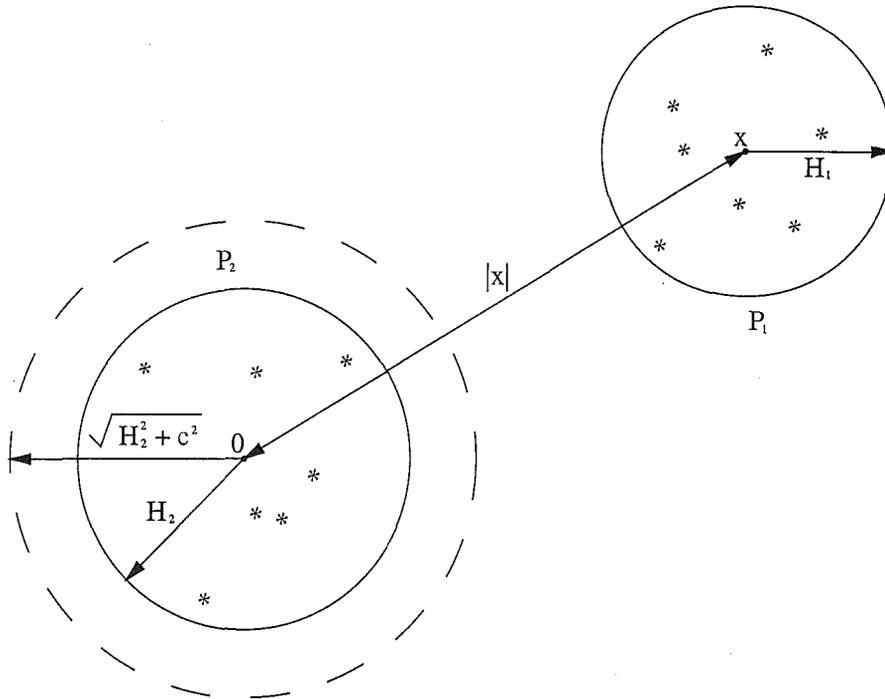


Figure 4.4: Two clusters of points (*) where the far field expansion is valid for any source point in P_1 and evaluation point in P_2 .

being far enough apart, and because γ and β annihilate linears we obtain far field

expansions of $\Phi(\cdot - y_i)$ about zero

$$\begin{aligned} \sum_{i,j=1}^{N+1} \gamma_i \beta_j \Phi(x_j - y_i) &= \sum_{i,j=1}^{N+1} \gamma_i \beta_j \left(|x_j| - \frac{s_i \xi_j + t_i \eta_j}{|x_j|} \right. \\ &\quad \left. + \frac{1}{2} \frac{(t_i^2 + \tau^2) \xi_j^2 + (s_i^2 + \tau^2) \eta_j^2 - 2s_i t_i \xi_j \eta_j}{|x_j|^3} + \mathcal{O}(|x|^{-2}) \right), \\ &= \sum_{i,j=1}^{N+1} \gamma_i \beta_j \left(\frac{1}{2} \frac{t_i^2 \xi_j^2 + s_i^2 \eta_j^2 - 2s_i t_i \xi_j \eta_j}{|x_j|^3} + \mathcal{O}(|x|^{-2}) \right). \end{aligned} \quad (4.31)$$

Taking absolute values and using Lemma 4.5.5 gives

$$\begin{aligned} \left| \sum_{i,j=1}^{N+1} \gamma_i \beta_j \Phi(x_j - y_i) \right| &\leq \frac{20}{2} \frac{H_1^2 E_1}{|x|^3} \sum_{i=1}^{N+1} |\gamma_i| (t_i^2 + s_i^2 + 2|s_i t_i|) + \mathcal{O}(|x|^{-4}), \\ &\leq \frac{20 H_1^2 H_2^2 E_1 E_2}{|x|^3} + \mathcal{O}(|x|^{-4}). \end{aligned} \quad (4.32)$$

If β and γ are boundary over distance coefficients then

$$\begin{aligned} \left| \sum_{i,j=1}^{N+1} \gamma_i \beta_j \Phi(x_j - y_i) \right| &\leq 40 \frac{\text{Area}(V_1)}{|x|^3} \sum_{i=1}^{N+1} |\gamma_i| (t_i^2 + s_i^2 + 2|s_i t_i|) + \mathcal{O}(|x|^{-4}), \\ &\leq 640 \frac{\text{Area}(V_1) \text{Area}(V_2)}{|x|^3} + \mathcal{O}(|x|^{-4}). \end{aligned} \quad (4.33)$$

□

Theorem 4.5.7. *Let Φ be the thin-plate spline and let $X = \{x_1, \dots, x_{N+1}\}$, $x_i = (\xi_i, \eta_i) \in \mathcal{R}^2$, be $N + 1$ distinct points contained in the circle $|\cdot - x| \leq H_1 < |x|$ where $x = x_{N+1}$. Also let $Y = \{y_1, \dots, y_{N+1}\}$, $y_i = (s_i, t_i) \in \mathcal{R}^2$, be $N + 1$ distinct points contained in the circle $|\cdot| < H_2$ with $y_{N+1} = 0$. Then if $\beta \in \mathcal{U}_X$, $\gamma \in \mathcal{U}_Y$ and $|x| > H_1 + H_2$,*

$$\left| \sum_{i=1}^{N+1} \sum_{j=1}^{N+1} \gamma_i \beta_j \Phi(x_j - y_i) \right| \leq \frac{44 H_1^2 H_2^2 E_1 E_2}{|x|^2} + \mathcal{O}(|x|^{-3}),$$

where $E_1 = \sum_{i=1}^N |\beta_i|$ and $E_2 = \sum_{i=1}^N |\gamma_i|$. Furthermore, if β and γ are boundary over distance weights

$$\left| \sum_{i=1}^{N+1} \sum_{j=1}^{N+1} \gamma_i \beta_j \Phi(x_j - y_i) \right| \leq \frac{464 \text{Area}(V_1) \text{Area}(V_2)}{|x|^2} + \mathcal{O}(|x|^{-3}),$$

where V_1 is the Voronoi region around x_{N+1} and V_2 is the Voronoi region around y_{N+1} .

Proof. With a similar approach to the proof of Theorem 4.5.6 we use the far field expansion of the thin-plate spline which is given in [11]. For $x = (\xi, \eta) > H_1 + H_2$ all the far field approximations will be valid. A single expansion of a thin-plate spline basic function centred at (s, t) is

$$\begin{aligned} & [(\xi - s)^2 + (\eta - t)^2] \log \left([(\xi - s)^2 + (\eta - t)^2]^{\frac{1}{2}} \right) \\ &= \frac{1}{2}(\xi^2 + \eta^2) \log(\xi^2 + \eta^2) - (s\xi + t\eta) \log(\xi^2 + \eta^2) - s\xi - t\eta \\ & \quad + \frac{1}{2}(s^2 + t^2) \log(\xi^2 + \eta^2) + \frac{1}{2} \frac{(3s^2 + t^2)\xi^2 + (s^2 + 3t^2)\eta^2 + 4st\xi\eta}{\xi^2 + \eta^2} + \mathcal{O}(|x|^{-1}). \end{aligned} \tag{4.34}$$

Using this expansion and summing over the centres X and Y as in (4.31), the linear terms are annihilated giving

$$\begin{aligned} \sum_{i,j=1}^{N+1} \gamma_i \beta_j \Phi(x_j - y_i) &= \frac{1}{2} \sum_{i,j=1}^{N+1} \gamma_i \beta_j \left((s_i^2 + t_i^2) \log(\xi_j^2 + \eta_j^2) \right. \\ & \quad \left. + \frac{(3s_i^2 + t_i^2)\xi_j^2 + (s_i^2 + 3t_i^2)\eta_j^2 + 4s_i t_i \xi_j \eta_j}{|x_j|^2} + \mathcal{O}(|x_j|^{-1}) \right). \end{aligned} \tag{4.35}$$

A Taylor expansion of $\log(\xi_j^2 + \eta_j^2)$ about $x = (\xi, \eta)$ is

$$\begin{aligned} \log(\xi_j^2 + \eta_j^2) &= \log(\xi^2 + \eta^2) - 2 \frac{\xi \Delta \xi_j + \eta \Delta \eta_j}{|x|^2} \\ & \quad + \frac{\xi^2 (\Delta \eta_j^2 - \Delta \xi_j^2) + \eta^2 (\Delta \xi_j^2 - \Delta \eta_j^2) - 4\xi \eta \Delta \xi_j \Delta \eta_j}{|x|^4} + \mathcal{O}(|x|^{-3}). \end{aligned} \tag{4.36}$$

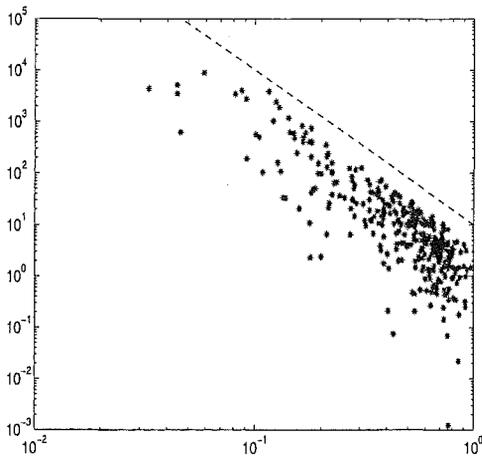
Substituting this Taylor expansion into (4.35) and noticing that the first two terms

in the Taylor expansion will be annihilated leads to the final summation

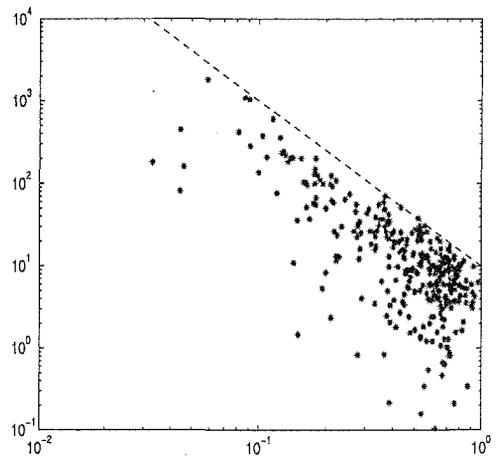
$$\begin{aligned} & \sum_{i,j=1}^{N+1} \gamma_i \beta_j \Phi(x_j - y_i) \\ &= \frac{1}{2} \sum_{i,j=1}^{N+1} \gamma_i \beta_j \left((s_i^2 + t_i^2) \frac{\xi^2(\Delta\eta_j^2 - \Delta\xi_j^2) + \eta^2(\Delta\xi_j^2 - \Delta\eta_j^2) - 4\xi\eta\Delta\xi_j\Delta\eta_j}{|x|^4} \right. \\ & \quad \left. + \frac{(3s_i^2 + t_i^2)\xi_j^2 + (s_i^2 + 3t_i^2)\eta_j^2 + 4s_i t_i \xi_j \eta_j}{|x_j|^2} + \mathcal{O}(|x_j|^{-1}) \right). \end{aligned}$$

Taking absolute values and using Lemma 4.5.5 gives the result. \square

The plots in Figure 4.5 show the distance $\|x_i - x_j\|$ vs $|B_{ij}|$ for a column j of B . The plotted values are only for indices i such that x_i is an internal centre. The total size of the data set is 400 centres of which 350 are internal. The decay rates obtained in this section are consistent with the plots. The dashed line in the log versus log plot 4.5(a) having slope -3, and that in plot 4.5(b) having slope -2.



(a) Multiquadric basis function.



(b) Thin-plate spline basis function.

Figure 4.5: Plots illustrating the results given in Theorems 4.5.6 and 4.5.7. The x-axis is $\|x_i - x_j\|$ and the y-axis is $|B_{ij}|$.

4.6 Numerical results

In this section we present numerical results for the following basic functions.

$$\text{thin-plate spline} \quad \phi(r) = r^2 \log(r), \quad (4.37)$$

$$\text{linear} \quad \phi(r) = -r, \quad (4.38)$$

$$\text{multiquadric} \quad \phi(r) = -\sqrt{r^2 + c^2}, \quad (4.39)$$

$$\text{inverse multiquadric} \quad \phi(r) = \frac{1}{\sqrt{r^2 + c^2}}. \quad (4.40)$$

Of these functions the thin-plate spline and linear functions satisfy the condition on Φ in Theorem 4.4.2 and will result in scale independent preconditioned matrices. In the following tables the matrix A_Φ is defined in (4.4), B in (4.5), S in Theorem 4.4.6 and the homogeneous matrix, C , is presented in [12]. In Table 4.1 we show condition numbers of matrices for the various preconditioning techniques over seven different scales. It is clear that the algorithm in Section 4.2 gives a matrix which dramatically improves the conditioning of the interpolation problem. In one case by a factor of 10^{14} ! Tables 4.2–4.5 contain condition numbers of the matrices resulting from applying the preconditioning techniques of this chapter for the basic functions (4.37)–(4.40). For $N < 3200$, the entries in the tables are the maximum over one hundred random point sets of size N . For $N = 3200$, the tables contain the maximum over twenty random point sets of size 3200. In all cases the preconditioning results in a smaller condition number. However the most impressive results are for the thin-plate spline, the linear, and the multiquadric basic functions. For these basic functions the maximum observed condition number of the scaled preconditioner, S , grows very slowly with N . Certainly there is no numerical evidence of power growth with N .

In an attempt to rule out the possibility that our numerical results were flukes due to the small number of 100 experiments we also conducted 50,000 trials with random data sets of size 100. The results of these trials are shown in Figure 4.1. The maximum condition number, over all trials with the thin-plate spline, for the

Scale parameter α	Conventional matrix A_ϕ	Homogeneous matrix C	Preconditioned matrix B	Scaled matrix S
0.001	1.531(11)	1.534(5)	4.905(1)	2.405(1)
0.01	1.544(9)	1.534(5)	4.905(1)	2.405(1)
0.1	1.597(7)	1.534(5)	4.905(1)	2.405(1)
1	3.107(5)	1.534(5)	4.905(1)	2.405(1)
10	1.915(6)	1.534(5)	4.905(1)	2.405(1)
100	1.271(11)	1.534(5)	4.905(1)	2.405(1)
1000	4.006(15)	1.534(5)	4.905(1)	2.405(1)

Table 4.1: Condition numbers for one hundred points in $[0, \alpha]^2$ and the thin-plate spline. The point set for scale α is $X_\alpha = \alpha X_1$.

Number of data points	Conventional matrix A_ϕ	Homogeneous matrix C	Preconditioned matrix B	Scaled matrix S
100	1.852(7)	1.285(7)	3.865(2)	4.877(1)
200	6.555(7)	3.068(7)	1.617(3)	6.028(1)
400	5.675(8)	3.397(8)	1.945(3)	8.946(1)
800	1.960(10)	1.348(10)	2.034(3)	9.775(1)
1600	1.092(10)	8.413(9)	8.099(3)	1.258(2)
3200	4.997(10)	3.783(10)	1.261(4)	1.569(2)

Table 4.2: Maximum condition numbers encountered over a sample of 100 random point sets of size N in $[0, 1]^2$ with the thin-plate spline.

matrix A_ϕ was 1.2465(9), for matrix C , 1.5750(9) and for matrix S , 1.8066(2). These maximum condition numbers and the results displayed in Figure 4.1 show that in our experiments the matrix S is always well conditioned. This held even for geometries of centres for which the matrix A_ϕ is very badly conditioned. These experiments lead one to suspect that the condition number of the matrix S may well be bounded

Number of data points	Conventional matrix A_ϕ	Preconditioned matrix B	Scaled matrix S
100	2.139(8)	1.129(2)	4.017(1)
200	2.014(8)	1.532(2)	4.224(1)
400	2.045(10)	5.932(2)	7.669(1)
800	6.641(10)	4.559(2)	5.826(1)
1600	1.554(10)	7.025(2)	5.601(1)
3200	2.477(11)	9.362(2)	6.280(1)

Table 4.3: Maximum condition numbers encountered over a sample of 100 random point sets of size N in $[0, 1]^2$ with the the multiquadric function with $c = 1/\sqrt{N}$.

Number of data points	Conventional matrix A_ϕ	Preconditioned matrix B	Scaled matrix S
100	9.732(4)	1.916(4)	6.468(1)
200	3.131(5)	6.099(4)	1.101(2)
400	1.178(6)	2.326(5)	2.364(2)
800	1.254(7)	2.826(5)	7.493(2)
1600	1.280(7)	4.227(5)	5.171(2)
3200	3.886(7)	2.815(6)	4.972(2)

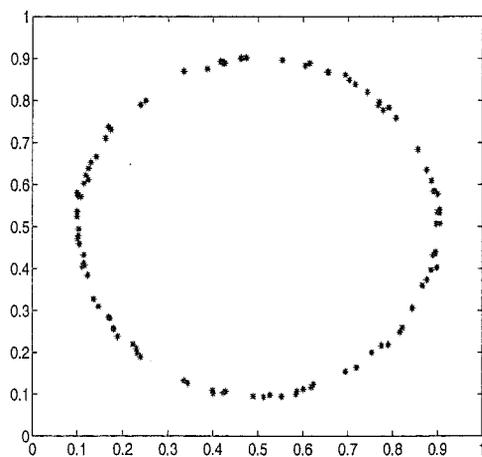
Table 4.4: Maximum condition numbers encountered over a sample of 100 random point sets of size N in $[0, 1]^2$ with the linear function.

independently of the geometry of the mesh. That is it may be bounded by a slowly growing function of N . To test further the behaviour of S for “bad” configurations of points a similar experiment was run with one thousand trials of one hundred points almost on a circle (for an example see Figure 4.6). The maximum condition numbers of the A matrix, C matrix and S matrix were 1.2885(9), 7.2692(8) and 6.6005(2) respectively over 1000 trials. Even though the Voronoi regions are long

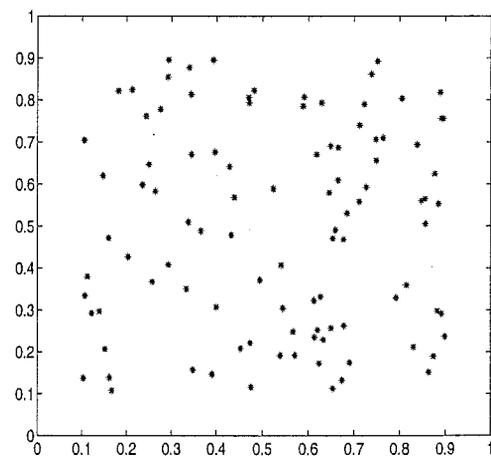
Number of data points	Conventional matrix A_ϕ	Preconditioned matrix B	Scaled matrix S
100	1.166(6)	1.180(3)	7.671(1)
200	9.370(5)	4.225(3)	1.725(2)
400	2.510(7)	9.017(3)	5.159(2)
800	5.853(7)	2.295(4)	1.338(3)
1600	8.174(6)	7.071(4)	3.633(3)
3200	5.311(7)	1.559(5)	9.997(3)

Table 4.5: Maximum condition numbers encountered over a sample of 100 random point sets of size N in $[0, 1]^2$ with the inverse multiquadric.

and thin the matrix is still well conditioned!



(a) One hundred centres almost on a circle.



(b) One hundred random data points in the square.

Figure 4.6: Examples of two configurations of points in the domain $[0, 1]^2$.

4.7 Preconditioning in \mathcal{R}^3

Common basic functions for RBF interpolation in \mathcal{R}^3 include the triharmonic $\Phi(\cdot) = |\cdot|^3$, the biharmonic $\Phi(\cdot) = |\cdot|$ and the multiquadric. The results of Narcowich and Ward show that for large data sets in \mathcal{R}^3 we would expect badly conditioned RBF interpolation matrices. Consequently there is a requirement for a preconditioner in three dimensions which reduces the condition number of the interpolation system. In this section we modify the algorithm of Section 4.2 in order to make it apply for this higher dimensional setting. It is not difficult to show that the theory in two dimensions can easily be transferred into three or more dimensions. For example in three dimensions, the coefficient matrix R , in Section 4.2 can be shown to be of rank $N - 4$ and the invertibility of the preconditioned matrix follows.

One common form of surface fitting in \mathcal{R}^3 is the reconstruction of a closed surface, for example, a scanned object [20]. This can involve finding a zero surface and the centres are usually not uniformly distributed. More traditional interpolation is also required in three dimensions, for example, in meteorology or mining. The preconditioner of Section 4.2 can be modified and applied to both these cases. In the surface reconstruction case the Voronoi regions are often long and thin which may make it difficult for numerical techniques to accurately find the Voronoi vertices. This may lead to the columns of R not being completely orthogonal to the matrix Q .

One difference in three dimensions is that the boundary between two Voronoi regions is a face instead of a line. The corresponding boundary over distance weight is then given by the area of this face over the distance between the two centres. To write the virtual points uniquely as a homogeneous linear combination of special points we use four centres spread throughout the domain. The virtual points are then of the form

$$\hat{x}_j = \lambda_{N-3}x_{N-3} + \lambda_{N-2}x_{N-2} + \lambda_{N-1}x_{N-1} + \lambda_Nx_N,$$

with $\lambda_{N-3} + \lambda_{N-2} + \lambda_{N-1} + \lambda_N = 1$. Implementing this algorithm in three dimensions

is more complicated as centres with Voronoi regions adjacent to edges or corners can have up to three faces on the boundary (if the domain W is a cube) and therefore three virtual points. As before coefficients from each virtual point are added together. The complexity for finding the Voronoi regions in \mathcal{R}^3 is $\mathcal{O}(N^{4/3})$ operations so is slightly more expensive than the $\mathcal{O}(N \log N)$ operations in \mathcal{R}^2 . Voronoi regions also have more neighbours in three dimensions and so slightly more work and storage is required to find the entries of R .

For uniformly distributed data in two dimensions the number of boundary points is $\mathcal{O}(N^{1/2})$ whereas in three dimensions this increases to $\mathcal{O}(N^{2/3})$. Consequently, more entries of B are sums of local centres and special points. The decay rates given in Theorems 4.5.6 and 4.5.7 for the two dimensional case are therefore valid for fewer entries in B . More simply put the preconditioner becomes less local which leads to the preconditioned matrix becoming less diagonally dominant and the eigenvalues less clustered. This is reflected in the condition numbers which are slightly higher than for the two dimensional case. However, as can be seen in Tables 4.6 - 4.8, they are still a great improvement over the usual formulation, A_Φ . For the multiquadric we see an improvement of almost six orders of magnitude between the condition number of A_Φ and the condition number of S for 3200 centres. When the triharmonic basic function or the biharmonic basic function are used the improvement is still significant.

4.8 Roundoff error and fast computation of the action of the preconditioned matrix

In previous sections it has been shown that the preconditioned system B is much better conditioned than A . However, when N is large we cannot store B and therefore the matrix-vector products that occur during an iterative fit are computed with a fast method and not by multiplying by B . In this section we address the question

Number of data points	Conventional matrix A_ϕ	Preconditioned matrix B	Scaled matrix S
100	9.9305(5)	9.8098(3)	1.6065(2)
200	9.1370(5)	1.3060(4)	2.5132(2)
400	1.1619(7)	3.5857(4)	4.0519(2)
800	4.3729(7)	5.3033(4)	7.4332(2)
1600	8.8497(7)	3.1376(5)	4.8450(2)
3200	5.5625(8)	1.6367(5)	8.7156(2)

Table 4.6: Condition numbers for various sized point sets in $[0, 1]^3$ for the multi-quadric function with $c = 1/N^{1/3}$.

Number of data points	Conventional matrix A_ϕ	Preconditioned matrix B	Scaled matrix S
100	6.5877(5)	1.4062(4)	7.0990(2)
200	1.1382(6)	3.6882(4)	2.1499(3)
400	1.3393(7)	1.0986(5)	6.9424(3)
800	6.5861(7)	2.2407(5)	1.3153(4)
1600	2.4541(8)	1.3645(6)	3.0752(4)
3200	1.4898(9)	1.8674(6)	1.3097(5)

Table 4.7: Condition numbers for various sized point sets in $[0, 1]^3$ for the triharmonic function, $\Phi(\cdot) = |\cdot|^3$.

of whether this indirect approach somehow negates all the advantages of the preconditioning. The answer is that it need not, especially if one builds a fast evaluator for the preconditioned function $\{\Psi_j\}$.

Since each ψ function is based on ϕ 's corresponding to a local cluster of centres together with ϕ 's associated with special points it is possible to develop hierarchical and fast multipole methods for fast approximate evaluation of $(AR)y$ rather than

Number of data points	Conventional matrix A_ϕ	Preconditioned matrix B	Scaled matrix S
100	2.9886(3)	1.4888(2)	1.4788(1)
200	6.4223(3)	1.7203(2)	2.7998(1)
400	2.0965(4)	2.0112(2)	3.4539(1)
800	4.9875(4)	3.1780(2)	4.2342(1)
1600	1.6073(5)	4.9316(2)	6.1167(1)
3200	4.4444(5)	8.4780(2)	5.8858(1)

Table 4.8: Condition numbers for various sized point sets in $[0, 1]^3$ for the biharmonic function, $\Phi(\cdot) = |\cdot|$.

Ay . That is it is possible to construct fast evaluators which work with the ψ 's rather than the ϕ 's. The approximate computation of By during an iterative fit would then be performed as a two stage process, $t \approx (AR)y$ and $x = R^T t$. The question which naturally arises is will computing $x = By$ in this two stage manner give reliable estimates of x . A partial answer is given by the error analysis and tables below. These show that computing By in this two stage manner can be expected to be much less susceptible to roundoff than computing via the three stage process corresponding to a fast evaluator for ϕ , $v = Ry$, $w \approx Av$ and $x = R^T w$.

Let $B := R^T AR$ and x the exact product $x = By$. The finite precision counterpart is $\hat{x} := fl(By)$ which from Higham [45, p76] has error

$$\frac{\|\hat{x} - x\|}{\|x\|} \leq \alpha_N \|B\| \|B^{-1}\|, \quad (4.41)$$

where α_N is

$$\alpha_N = \frac{N\epsilon}{1 - N\epsilon},$$

with ϵ being the unit roundoff ($\approx 1 \times 10^{-16}$ for an IEEE double) and $\|\cdot\|$ is either the 1, ∞ or Frobenius norm.

Each column of R has a relatively small number, β say, of non-zero entries so

the product $C = AR$ can be found accurately. From Higham [45, p76], this matrix-matrix product will have error

$$\|C - \hat{C}\| \leq \alpha_\beta \|A\| \|R\|,$$

where $\hat{C} = fl(AR)$ is the finite precision product of AR . For large N , $\alpha_N \gg \alpha_\beta$ so for now we ignore the small error in finding C .

To find the product By by a two stage process without storing B requires the matrix-vector products,

$$t = Cy, \quad x = R^T t.$$

or in finite precision form,

$$\hat{x} = fl(R^T fl(Cy)).$$

We are interested in the error of computing \hat{x} . Let

$$\begin{aligned} \hat{t} &= fl(Cy), \\ &= (C + \Delta C)y, \quad |\Delta C| \leq \alpha_N |C|, \\ &= t + \Delta Cy. \end{aligned}$$

Then \hat{x} is

$$\begin{aligned} \hat{x} &= fl(R^T \hat{t}), \\ &= (R + \Delta R)^T \hat{t}, \quad |\Delta R| \leq \alpha_\beta |R|, \\ &= x + \Delta R^T t + R^T \Delta C y + \mathcal{O}(\epsilon^2). \end{aligned}$$

Paige [66] gives the relationship

$$\|\Delta R\| \leq \|\Delta R\| \leq \alpha_\beta \|R\| = \alpha_\beta \gamma_R \|R\|,$$

where $\gamma_R = 1$ if $\|\cdot\|$ is one of the 1, ∞ or Frobenius norms and $\gamma_R \leq \sqrt{n}$ for the 2-norm. A similar relationship exists between ΔC and C and between ΔA and A .

In the following discussion $\|\cdot\|$ is either the 2-norm or the Frobenius norm. The error in finding x is,

$$\begin{aligned}
\|\hat{x} - x\| &\leq \|\Delta R^T t\| + \|R^T \Delta C y\| + \mathcal{O}(\epsilon^2), \\
&\approx \|\Delta R^T C y\| + \|R^T \Delta C y\|, \\
&\leq \|\Delta R^T\| \|C\| \|y\| + \|R^T\| \|\Delta C\| \|y\|, \\
&\leq \alpha_\beta \gamma_R \|R\| \|C\| \|y\| + \alpha_N \gamma_C \|R\| \|C\| \|y\|, \\
&= (\alpha_\beta \gamma_R + \alpha_N \gamma_C) \|R\| \|C\| \|y\|.
\end{aligned}$$

Now $y = B^{-1}x$ implies $\|y\| \leq \|B^{-1}\| \|x\|$ which leads to the relative error

$$\frac{\|\hat{x} - x\|}{\|x\|} \lesssim \alpha_N \gamma_C \|B^{-1}\| \|R\| \|AR\| = \alpha_N b_2(X, \Phi), \quad (4.42)$$

where $b_2(X, \Phi) := \gamma_C \|B^{-1}\| \|R\| \|AR\|$. For the three stage process of finding \hat{x} a similar analysis gives the bound

$$\frac{\|\hat{x} - x\|}{\|x\|} \lesssim \alpha_N \gamma_A \|B^{-1}\| \|R\|^2 \|A\| = \alpha_N b_3(X, \Phi), \quad (4.43)$$

with $b_3(X, \Phi) := \gamma_A \|B^{-1}\| \|R\|^2 \|A\|$. Note that the bounds given in this section are from a basic analysis and as such can be improved upon. However, they are acceptable here as they show that the two stage process, $x = R^T(Cy)$, will be sufficiently accurate for most purposes. Using the notation above, equation (4.41) for the 2-norm is

$$\frac{\|\hat{x} - x\|}{\|x\|} \leq \alpha_N \gamma_B \|B^{-1}\|_2 \|B\|_2 = \alpha_N b_1(X, \Phi), \quad (4.44)$$

where $b_1(X, \Phi) := \gamma_B \|B^{-1}\|_2 \|B\|_2$.

Tables 4.9-4.11 calculate the bounds (4.42) - (4.44) for various basic functions and centres, X , in $[0, 1]^2$. Numbers in the tables are with respect to the 2-norm. These results show that the bound on the two stage process is a lot smaller than the bound on the three stage process. As expected though direct multiplication with a stored B gives the smallest bound. Of course such direct multiplication is impractical for large N .

Grid size, N	$b_3(X, \Phi)$	$b_2(X, \Phi)$	$b_1(X, \Phi)$
10×10	5.928(4)	4.465(2)	3.285(1)
20×20	3.813(5)	8.258(2)	2.361(1)
30×30	1.663(6)	1.644(3)	2.401(1)
40×40	5.258(6)	2.919(3)	2.674(1)

Table 4.9: Bounds given in this section for the multiquadric basic function with $c = 1/N$.

Grid size, N	$b_3(X, \Phi)$	$b_2(X, \Phi)$	$b_1(X, \Phi)$
10×10	1.088(6)	6.685(3)	2.404(2)
20×20	7.140(6)	1.341(4)	2.248(2)
30×30	2.769(7)	2.464(4)	2.327(2)
40×40	7.068(7)	3.608(4)	2.130(2)

Table 4.10: Bounds given in this section for the multiquadric basic function with $c = 2/N$.

Grid size, N	$b_3(X, \Phi)$	$b_2(X, \Phi)$	$b_1(X, \Phi)$
10×10	2.053(4)	6.183(2)	3.328(1)
20×20	4.105(5)	2.946(3)	4.298(1)
30×30	2.260(6)	6.933(3)	4.308(1)
40×40	7.564(6)	1.263(4)	4.159(1)

Table 4.11: Bounds given in this section for the thin-plate spline basic function.

Chapter 5

An algebraic multigrid algorithm

This chapter considers an algebraic multigrid method to solve systems like

$$\begin{bmatrix} A & P \\ P^T & O \end{bmatrix} \begin{bmatrix} \lambda \\ a \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}, \quad (5.1)$$

which arise from RBF interpolation problems. We saw in Section 4.2 that the coefficients λ can be found by solving

$$B\mu = Q^T f,$$

where Q is an $N \times (N - l)$ matrix whose columns span the orthogonal complement of P , and B is the symmetric positive definite $(N - l) \times (N - l)$ matrix $B = Q^T A Q$. Once μ is known $\lambda = Q\mu$ and a is obtained by interpolation to the residual at the special points.

In [12] a domain decomposition algorithm was presented for solving the system (5.1) with respect to the Φ basis. Their results showed that the coefficients can be found in a small number of iterations. This chapter uses the Voronoi preconditioner of Chapter 4 to construct an algebraic multigrid algorithm which is competitive with the domain decomposition method of [12]. Sections 5.1 and 5.2 consider forming fine and coarse level approximations to λ respectively. Then Section 5.3 presents numerical results which show that this multigrid method converges rapidly to λ .

5.1 Fine level approximation

Given B^{-1} the true solution for μ is $\mu = B^{-1}Q^T f$. Analogously, if M is an approximate inverse of B then our initial (fine level) approximate solution for μ is $\tilde{\mu} = MQ^T f$ and our initial (fine level) approximate solution for λ is $\tilde{\lambda} = Q\tilde{\mu} = QMQ^T f$. Hence, since A is a premultiplier that evaluates, at the centres, an RBF with coefficient values λ and without polynomial part, the residual vector arising from this fine level approximation M to B^{-1} , is $R_F f$ where

$$R_F = I - AQMQ^T. \quad (5.2)$$

We now detail a construction of an approximate inverse on the fine grid nodes. Let m be the number of domains and define the inner indices sets $\mathcal{I}_j \subset \{1, \dots, N-l\}$, $j = 1, \dots, m$ such that, the union of these m sets exhausts $\{1, \dots, N-l\}$, and \mathcal{I}_i and \mathcal{I}_j are disjoint for $i \neq j$. Corresponding to each set of inner indices is a set of outer indices, \mathcal{O}_j , such that $\mathcal{I}_j \cap \mathcal{O}_j = \emptyset$. Define $\hat{f} = Q^T f$ then the coefficients μ_k , $k \in \mathcal{I}_j$ are approximated from the data $\{\hat{f}_l : l \in \mathcal{T}_j\}$ where $\mathcal{T}_j := \mathcal{I}_j \cup \mathcal{O}_j$.

Extract the columns and rows of B corresponding to the indices \mathcal{T}_j and call this B_j . Call the corresponding vector derived from \hat{f} , $\hat{f}^{(j)}$. Now let the matrix M_j be the rows of B_j^{-1} corresponding to the inner indices, \mathcal{I}_j . Then the approximation to μ on the j th subdomain is

$$\tilde{\mu}^{(j)} = M_j \hat{f}^{(j)}. \quad (5.3)$$

Applying this procedure to each of the subdomains gives the approximation to μ . The matrix M is formed from the blocks M_j so that $\tilde{\mu} = M\hat{f}$. In practice M is not stored directly but instead the matrices M_j are, so that (5.3) is calculated for $j = 1, \dots, m$ at each iteration. The following well known lemma proves convergence of a simple iterative method if M is a sufficiently good approximate inverse of A .

Lemma 5.1.1. *Let M be an approximate inverse to $C \in \mathcal{R}^{N \times N}$ and write $R = I - CM$ where $\rho(R) < 1$. Then solving the equations*

$$C\lambda = f,$$

with the iteration

$$r_k = f - C\lambda_k, \quad \lambda_{k+1} = \lambda_k + Mr_k,$$

where $\lambda_0 = Mf$ will converge to λ for all f . After $k+1$ iterations the approximation will satisfy

$$\frac{\|\lambda - \lambda_{k+1}\|}{\|\lambda\|} \leq \text{cond}(C)\|R^{k+1}\|,$$

where $\|\cdot\|$ applied to vectors is any norm, and applied to matrices is the corresponding operator norm.

Proof. For any f , $r_1 = f - C\lambda_1 = (I - CM)f = Rf$. Now assume the residual after k iterations is $r_k = R^k f$ then

$$\begin{aligned} r_{k+1} &= f - C\lambda_{k+1} \\ &= f - C(\lambda_k + Mr_k), \\ &= r_k - CMr_k, \\ &= (I - CM)r_k = R^{k+1}f. \end{aligned} \tag{5.4}$$

Also,

$$\begin{aligned} R^{k+1}f &= f - C\lambda_{k+1}, \\ &= C\lambda - C\lambda_{k+1} = C(\lambda - \lambda_{k+1}), \end{aligned} \tag{5.5}$$

which implies

$$\|\lambda - \lambda_{k+1}\| \leq \|C^{-1}\| \|R^{k+1}\| \|f\|. \tag{5.6}$$

As $k \rightarrow \infty$, $\|R^{k+1}\| \rightarrow 0$ since $\rho(R) < 1$ (see for example Theorem 5.6.12 of Horn and Johnson [48]). The bound comes from equation (5.6) and $\|f\| \leq \|C\| \|\lambda\|$. \square

5.2 Coarse grid approximation

In this section we add a coarse grid correction to the fine grid approximation given in the previous section. The coarse grid correction is designed to correct any smooth

signal in the error. Since the previously described one level method generates an approximation without polynomial part the approximations have no hope of solving the system (5.1) in general. The coarse grid correction may be as simple as generating a purely polynomial update by interpolation at the special points. However, experience shows that in general it is better to have a denser coarse grid, and a correction consisting of an RBF generated by interpolation at more than l points.

Let R_F be the matrix updating the residuals given in (5.2). Choose from the N points $t \geq l$ points unisolvent for π_{k-1}^d . These t points include the points $\{x_{N-l+1}, \dots, x_N\}$. Taking the corresponding t rows of R_F and supplementing with l rows of zeros we obtain the matrix \tilde{R}_F which as a premultiplier takes input vector f and returns residuals at the t special points. Form the interpolation matrix (as in (5.1)) with respect to the usual basis for the coarse grid centres and label it's inverse as \tilde{T} . Then the coefficients of a coarse grid interpolant to f , or rather those entries in f corresponding to coarse level centres, are given by $\tilde{T}\tilde{R}_F f$.

Define a matrix \tilde{A} by selecting from A the columns corresponding to the coarse grid points. Then as a multiplier the matrix

$$\begin{pmatrix} \tilde{A} & P \end{pmatrix},$$

maps the coefficients of a coarse grid RBF into its values on the fine grid.

Thus the final approximation to the input values at all the centres is

$$\begin{aligned} & \text{initial fine grid approximation} + \text{coarse grid approximation} \\ &= AQMQ^T f + \begin{pmatrix} \tilde{A} & P \end{pmatrix} \tilde{T}\tilde{R}_F f. \end{aligned}$$

The matrix

$$R = I - \left\{ AQMQ^T + \begin{pmatrix} \tilde{A} & P \end{pmatrix} \tilde{T}\tilde{R}_F \right\}, \quad (5.7)$$

then maps a right hand side vector to a residual vector.

5.3 Numerical results

This section gives numerical results for the algebraic multigrid method of this chapter and compares them with the domain decomposition method given in [12]. The numerical results are for centres in $[0, 1]^2$ and the algorithms are implemented with a 2×2 grid of domains.

Tables 5.1 and 5.2 show the spectral radius of the residual matrix R using the algebraic multigrid and the domain decomposition methods respectively and for the thin-plate spline basic function. Note that the fine grid is all the nodes and the correction grid is the coarse grid nodes. The algebraic multigrid method outperforms the domain decomposition method when the overlap is small. This is expected as each function Ψ_i coming from the Voronoi preconditioner is a combination of a cluster of ϕ 's corresponding to at least local centres meaning overlap is implicitly built into the method. The performance of the two methods is closer for larger overlap. Tables 5.3 and 5.4 give similar results with the multiquadric basic function.

Correction grid	Fine grid	Number of overlap rows/columns		
		2	3	4
3 × 3	10 × 10	7.30(-2)	5.01(-2)	3.41(-2)
	20 × 20	2.05(-1)	1.05(-1)	7.65(-2)
	40 × 40	2.66(-1)	1.42(-1)	1.35(-1)
4 × 4	10 × 10	4.82(-2)	3.19(-2)	1.28(-2)
	20 × 20	1.06(-1)	6.43(-2)	4.92(-2)
	40 × 40	2.34(-1)	1.68(-1)	1.09(-1)

Table 5.1: Spectral radius of the residual matrix for the algebraic multigrid method and the thin-plate spline basic function

Residuals, $\|f - A\lambda_k\|_2$, at each iteration are also compared in Tables 5.5 and 5.6

Correction grid	Fine grid	Number of overlap rows/columns		
		2	3	4
3 × 3	10 × 10	1.39(-1)	9.15(-2)	5.29(-2)
	20 × 20	1.77(-1)	1.46(-1)	1.35(-1)
	40 × 40	3.33(-1)	1.82(-1)	1.76(-1)
4 × 4	10 × 10	1.14(-1)	5.12(-2)	3.68(-2)
	20 × 20	2.87(-1)	5.08(-2)	4.75(-2)
	40 × 40	7.14(-1)	1.61(-1)	6.22(-2)

Table 5.2: Spectral radius of the residual matrix for the method of [12] and the thin-plate spline basic function.

Correction grid	Fine grid	Number of overlap rows/columns		
		2	3	4
3 × 3	10 × 10	7.14(-2)	3.89(-2)	2.01(-2)
	20 × 20	1.46(-1)	9.09(-2)	8.28(-2)
	40 × 40	3.17(-1)	2.74(-1)	2.28(-1)
4 × 4	10 × 10	4.24(-2)	3.03(-2)	2.05(-2)
	20 × 20	5.64(-2)	5.06(-2)	3.81(-2)
	40 × 40	1.48(-1)	1.08(-1)	5.61(-2)

Table 5.3: Spectral radius of the residual matrix for the algebraic multigrid method and the multiquadric basic function

for centres in a 40×40 grid. The interpolated data comes from the Franke function, specified in equation (1.29) of the Introduction chapter. For these examples the residuals are similar between algorithms.

Correction grid	Fine grid	Number of overlap rows/columns		
		2	3	4
3×3	10×10	1.89(-1)	1.35(-1)	9.06(-2)
	20×20	2.13(-1)	1.33(-1)	1.86(-1)
	40×40	2.85(-1)	1.63(-1)	1.94(-1)
4×4	10×10	1.47(-1)	7.71(-2)	6.39(-2)
	20×20	3.71(-1)	1.39(-1)	5.25(-2)
	40×40	5.55(-1)	3.03(-1)	7.97(-2)

Table 5.4: Spectral radius of the residual matrix for the method of [12] and the multiquadric basic function.

Iteration number	Algorithm of [12]	Algebraic multigrid
1	4.488(-1)	3.658(-1)
2	3.071(-2)	2.136(-2)
3	1.295(-3)	1.678(-3)
4	8.764(-5)	7.955(-5)
5	5.977(-6)	6.000(-6)
6	4.386(-7)	4.266(-7)
7	2.691(-8)	2.939(-8)
8	1.785(-9)	1.775(-9)

Table 5.5: Two norm residuals at each iteration, for solving (5.1) by two different algorithms on a 40×40 grid. Multiquadric basic function with $c = 1/40$.

Iteration number	Algorithm of [12]	Algebraic multigrid
1	2.420(-1)	6.563(-2)
2	1.429(-2)	9.614(-3)
3	7.632(-4)	5.454(-4)
4	4.835(-5)	3.567(-5)
5	2.457(-6)	1.914(-6)
6	1.658(-7)	1.235(-7)
7	8.505(-9)	6.667(-9)
8	5.721(-10)	4.282(-10)

Table 5.6: Two norm residuals at each iteration, for solving (5.1) by two different algorithms on a 40×40 grid. Thin-plate spline basic function.

Chapter 6

RBF collocation

6.1 Introduction

In recent years radial basis function collocation has become a useful alternative to finite difference and finite element methods for solving elliptic partial differential equations. RBF collocation methods have been shown numerically (see for example [51]) and theoretically (see [41, 40]) to be very accurate even for a small number of collocation points. In application finite difference methods often have a low approximation order and consequently can require a large grid and considerable computation to obtain a sufficiently accurate solution. RBF collocation has been applied to linear elliptic PDEs in \mathcal{R}^2 and \mathcal{R}^3 [52], to time dependent problems [46, 47], and to non-linear problems [36].

In this chapter we present new numerical results for RBF collocation. These results show that collocation with a basic function from the Matern class can be more accurate than collocation with the multiquadric basic function. Also, we present and implement an algorithm which solves linear and non-linear collocation equations with the multiquadric when N is large and $c < 2/\sqrt{N}$.

Section 6.2 briefly outlines RBF collocation and discusses difficulties with the method. These include poor conditioning and full matrices when using globally

supported basic functions, and lower accuracy when using compactly supported basic functions. In Section 6.3 we give numerical results using a family of basic functions known as the Matern family. These numerical results show that this family is an effective alternative to the multiquadric basic function in many situations. Finally, in the last sections, we present a method which can be used to solve large collocation problems with the multiquadric basic function and $c < 2/\sqrt{N}$. Numerical experiments on linear PDEs show convergence to the solution for small enough values of c . It is hoped that in the future the algorithm will be able to be applied to larger values of c . This new algorithm combines the use of approximate cardinal functions and domain decomposition to iteratively find the solution of the collocation problem. Using approximate cardinal functions as a change of basis has been shown to be effective in the interpolation setting [6]. Previously solving a collocation system required $\mathcal{O}(N^3)$ operations (for globally supported Φ) and was not possible for large N . The method presented here solves the collocation system in $\mathcal{O}(N \log N)$ operations if $c < 2/\sqrt{N}$ and the PDE is suitable.

6.2 RBF collocation

This chapter considers solving a suitable elliptic PDE of the form

$$\begin{aligned} Lu &= f & \text{in } \Omega \subset \mathcal{R}^d, \\ u &= g & \text{in } \partial\Omega, \end{aligned} \tag{6.1}$$

by radial basis function collocation. In (6.1) $f, g : \mathcal{R}^d \rightarrow \mathcal{R}$ are known and $\partial\Omega$ is the boundary of the region Ω . L is a differential operator and may be linear or non-linear. If L is non-linear a multilevel Newton iteration is required and a linearized system is solved at each level.

The unknown solution, u , to the PDE is approximated by a radial basis function, u_ϕ , of the form

$$u_\phi(\cdot) = p(\cdot) + \sum_{j=1}^N \lambda_j \Phi(\cdot - x_j). \tag{6.2}$$

Here $\lambda = [\lambda_1, \dots, \lambda_N]^T$ are coefficients to be found, $p \in \pi_k^d$, and Φ is a basic function, such as the multiquadric. If L is time dependent then we let λ be a function of time and solve for $\lambda(t)$ at a finite number of discrete time steps. For more discussion on this case see [51]. For the moment assume L is time independent. Now for u_ϕ to satisfy the PDE (6.1) then

$$\begin{aligned} Lu_\phi(x) &= f(x), \quad x \in \Omega, \\ u_\phi(x) &= g(x), \quad x \in \partial\Omega. \end{aligned} \tag{6.3}$$

Clearly this cannot generally be achieved for every point in Ω . By choosing N distinct collocation points $X_I = \{x_1, \dots, x_{N_I}\} \subset \Omega$ and $X_B = \{x_{N_I+1}, \dots, x_N\} \subset \partial\Omega$ and ensuring (6.3) holds at these points we expect u_ϕ will be a good approximation to u . For the choice of u_ϕ in (6.2) the collocation equations are

$$\begin{aligned} Lp(x_i) + \sum_{j=1}^N \lambda_j L\Phi(x_i - x_j) &= f_i, \quad i = 1, \dots, N_I, \\ p(x_i) + \sum_{j=1}^N \lambda_j \Phi(x_i - x_j) &= g_i, \quad i = N_I + 1, \dots, N, \end{aligned}$$

along with the side conditions

$$\sum_{j=1}^N \lambda_j q(x_j) = 0, \quad \text{for all } q \in \pi_k^d.$$

This leads to the equivalent matrix form

$$\begin{bmatrix} W_L & P_L \\ W_B & P_B \\ P^T & O \end{bmatrix} \begin{bmatrix} \lambda \\ a \end{bmatrix} = \begin{bmatrix} f \\ g \\ 0 \end{bmatrix}, \tag{6.4}$$

where

$$\begin{aligned} (W_L)_{ij} &= L\Phi(x_i - x_j), \quad x_i \in X_I, \quad x_j \in X, \\ (W_B)_{i-N_I, j} &= \Phi(x_i - x_j), \quad x_i \in X_B, \quad x_j \in X, \\ (P_L)_{ij} &= Lp_j(x_i), \quad x_i \in X_I, \\ (P_B)_{i-N_I, j} &= p_j(x_i), \quad x_i \in X_B, \end{aligned} \tag{6.5}$$

and $\{p_1, \dots, p_{\dim(\pi_k^d)}\}$ forms a basis for π_k^d . The vector a consists of coefficients with respect to this basis. Solving this collocation system for the coefficients $[\lambda^T \ a^T]^T$, when N is large, is the emphasis of later sections of this chapter. The strategy there is to precondition the collocation matrix

$$A = \begin{bmatrix} W_L & P_L \\ W_B & P_B \\ P^T & O \end{bmatrix}, \quad (6.6)$$

so that the preconditioned system is solved quickly using an iterative method. The collocation matrix, A , in (6.6) has not been proven to be non-singular but in [75] it was shown that finding a numerically singular matrix was very rare. The positioning of the centres has an effect on the accuracy of RBF collocation. However, to keep the discussion simpler, we only consider gridded centres.

Equation (6.2) is the form of the RBF approximation that was initially presented by Kansa [51]. This form is often called unsymmetric collocation due to the matrix in (6.6) being unsymmetric. An alternative approach [34], referred to as symmetric collocation, takes the form

$$u_\phi(\cdot) = p(\cdot) + \sum_{j=1}^{N_I} \lambda_j \tilde{L}\Phi(\cdot - x_j) + \sum_{j=N_I+1}^N \lambda_j \Phi(\cdot - x_j), \quad (6.7)$$

where \tilde{L} is the operator L now applied to the second argument, x_j . Note that the absolute values of $\tilde{L}\Phi(y - x)$ and $L\Phi(y - x)$ are equal for any x and y . For the choice of u_ϕ in (6.7) the collocation equations lead to the interpolation system

$$\begin{bmatrix} W_{L\tilde{L}} & W_L & P_L \\ W_{\tilde{L}}^T & W_B & P_B \\ P_L^T & P_B^T & O \end{bmatrix} \begin{bmatrix} \lambda \\ a \end{bmatrix} = \begin{bmatrix} f \\ g \\ 0 \end{bmatrix}. \quad (6.8)$$

The matrices in (6.8) are,

$$\begin{aligned}
 (W_{L\tilde{L}})_{ij} &= L\tilde{L}\Phi(x_i - x_j), & x_i, x_j \in X_I, \\
 (W_L)_{i,j-N_I} &= L\Phi(x_i - x_j), & x_i \in X_I, x_j \in X_B, \\
 (W_{\tilde{L}})_{i,j-N_I} &= \tilde{L}\Phi(x_i - x_j), & x_i \in X_I, x_j \in X_B, \\
 (W_B)_{i-N_I,j-N_I} &= \Phi(x_i - x_j), & x_i, x_j \in X_B,
 \end{aligned} \tag{6.9}$$

and P_L and P_B are the same as in (6.4). The main advantage of this formulation is that it is provably non-singular (see [34, 91]). However the RBF in (6.7) is not as widely used as Kansa's original due to an extra application of L requiring that Φ be more differentiable. For nonlinear collocation using (6.7) also increases the complexity of the method. Some numerical results comparing the two approaches can be found in [34].

Both collocation systems are generally very badly conditioned which can restrict the use of RBF collocation to systems with only a few thousand centres. Theoretical results show that multiquadric interpolation becomes more accurate as the multiquadric parameter c increases [56]. A lot of numerical evidence agrees with this in the collocation setting. However, as c gets larger the graph of the basic function becomes flatter and this leads to bad conditioning. Thus as the accuracy of the approximation increases then often so does the ill-conditioning. Various techniques have been used with mixed success to combat this problem (see for example [52]).

The problems associated with using globally supported basic functions have led to the use of compactly supported basic functions such as the Wendland functions [89]. If the support is small then matrix-vector multiplies can be calculated in $\mathcal{O}(N)$ operations. The problem with compactly supported basic functions is that good approximations to the solution are only obtained when the support is large. For accurate results the sparsity of the matrix is lost. A multilevel approach with smoothing can improve the accuracy of the RBF approximation [33] but multiquadric basic functions are usually more accurate.

6.3 Collocation with Matern basic functions

Traditionally multiquadric or compactly supported basic functions are the preferred choice for RBF collocation. Numerical evidence has shown good results with these choices of basic functions for various types of problems. Other alternatives that are common in the RBF interpolation setting can be restricted in their use for collocation. For example, the Laplacian of the thin-plate spline is

$$\Delta\Phi(x) = 4\log(\|x\|) + 4,$$

which has a discontinuity at zero. The Laplacian of the exponential basic function also has a discontinuity at zero. This makes the use of the thin-plate spline and exponential limited in RBF collocation.

Due to the conditioning problems associated with the multiquadric we consider the use of alternative basic functions for RBF collocation. This section presents numerical results for some simple PDEs using the Matern family as basic functions. The Matern family is given by

$$\phi_\nu(r) = \frac{2^{1-\nu}}{\Gamma(\nu)}(cr)^\nu K_\nu(cr), \quad (6.10)$$

where K_ν is a modified Bessel function of order $\nu > 0$ (note that ν is also a smoothness parameter) and $c > 0$. If n is a nonnegative integer then (6.10) simplifies to

$$\phi_{n+1/2}(r) = \frac{\exp(-cr)(cr)^n}{(2n-1)!!} \sum_{k=0}^n \frac{(n+k)!}{k!(n-k)!(2cr)^k}.$$

Some examples for various values of ν are:

$$\begin{aligned} \nu = 1/2, \quad \phi(r) &= \exp(-cr), \\ \nu = 1, \quad \phi(r) &= crK_1(cr), \\ \nu = 3/2, \quad \phi(r) &= (1+cr)\exp(-cr), \\ \nu = 5/2, \quad \phi(r) &= (1+cr+c^2r^2/3)\exp(-cr). \end{aligned} \quad (6.11)$$

Although we only consider unsymmetric collocation here the motivation behind the use of the Matern class comes from the results of Franke and Schaback [41, 40]

in the symmetric collocation setting. They show that for a PDE of order m the L_∞ approximation order for RBF collocation with a Matern basic function will be $\nu - m$. Note that this result is for solutions u in the “native space” of Φ . A complete review of the work of Franke and Schaback is beyond the scope of this thesis but the reader is referred to their papers [41, 40].

Tables 6.1 and 6.2 contain condition numbers of the collocation matrix (6.6) and relative error results for the PDE

$$\begin{aligned}\Delta u &= 32 \cos(4x_1 + 4x_2), & (x_1, x_2) \in \Omega, \\ u &= \cos(4x_1 + 4x_2), & (x_1, x_2) \in \partial\Omega,\end{aligned}$$

where Ω is the unit square. The relative error is $\|s - u\|_2 / \|u\|_2$ where s is the values of the RBF and u the values of the true solution evaluated on a uniform grid of size $(2\sqrt{N} - 1) \times (2\sqrt{N} - 1)$. The basic functions we compare are the multiquadric and the Matern, $\nu = 9/2$, function.

It is clear from the tables that as the basic function becomes flatter the condition number increases for a fixed set size. In the case of the multiquadric this corresponds to c increasing, whereas for the Matern function this corresponds to a decrease in c .

Table 6.1 shows results for centres on a uniform grid in $[0, 1]^2$. The smallest relative error for the Matern function is about 9 times smaller than the smallest relative error for the multiquadric. However, both these experiments have condition numbers greater than 10^{20} . If we look at experiments with condition numbers that are about 10^{16} or less then the difference between the basic functions is even more dramatic. The best results are then approximately 1.7×10^{-5} and 4×10^{-7} for the multiquadric and Matern functions respectively. The error for the Matern function is about 40 times smaller than the error for the multiquadric!

The same experiments were repeated on a grid of shifted Chebychev nodes in $[0, 1]^2$. The results are in Table 6.2. The errors for these trials were as low as 1.13×10^{-8} for Matern collocation on 4225 centres. Overall, for this PDE, RBF collocation with the Matern basic function was more accurate than RBF collocation

with the multiquadric especially for large N . Also collocation on the Chebychev grid was more accurate than collocation on the uniform grid.

Number of centres N	Multiquadric			Matern, $\nu = 9/2$		
	c	relative error	condition number	c	relative error	condition number
9×9	15/9	<u>7.552(-5)</u>	6.555(17)	0.5	<u>1.056(-3)</u>	1.694(16)
	13/9	1.217(-4)	1.038(17)	1.0	1.079(-3)	3.069(13)
	11/9	2.582(-4)	2.037(15)	1.5	1.108(-3)	9.406(11)
	9/9	5.684(-4)	2.877(13)	2.0	1.163(-3)	9.117(10)
	7/9	1.250(-3)	2.447(11)	2.5	1.251(-3)	1.601(10)
17×17	15/17	<u>2.095(-6)</u>	1.492(19)	0.5	6.860(-4)	6.704(18)
	13/17	5.656(-6)	3.610(19)	1.0	<u>4.488(-5)</u>	1.603(17)
	11/17	1.854(-5)	5.778(18)	1.5	4.703(-5)	3.023(15)
	9/17	6.131(-5)	2.378(16)	2.0	5.175(-5)	3.109(14)
	7/17	2.152(-4)	4.151(13)	2.5	5.796(-5)	5.518(13)
33×33	15/33	1.895(-6)	2.594(20)	2.0	<u>1.628(-6)</u>	1.338(19)
	13/33	<u>9.995(-7)</u>	4.394(20)	2.5	1.952(-6)	1.484(17)
	11/33	3.345(-6)	9.664(20)	3.0	2.266(-6)	3.162(16)
	9/33	1.397(-5)	2.063(18)	3.5	2.684(-6)	9.402(15)
	7/33	5.648(-5)	1.397(15)	4.0	3.232(-6)	3.188(15)
65×65	15/65	1.536(-6)	1.862(21)	3.0	2.139(-7)	3.682(20)
	13/65	<u>1.099(-6)</u>	3.087(21)	4.0	<u>1.112(-7)</u>	1.472(20)
	11/65	1.423(-6)	8.629(20)	5.0	1.419(-7)	3.792(18)
	9/65	4.170(-6)	7.398(20)	6.0	2.290(-7)	8.410(17)
	7/65	1.697(-5)	2.086(16)	7.0	3.724(-7)	5.944(16)

Table 6.1: Radial basis function collocation of the Poisson equation with solution $\cos(4x_1 + 4x_2)$. Uniform grid.

Number of centres N	Multiquadric			Matern, $\nu = 9/2$		
	c	relative error	condition number	c	relative error	condition number
9×9	15/9	<u>3.026(-5)</u>	4.737(17)	0.5	<u>3.966(-4)</u>	1.775(16)
	13/9	4.962(-5)	2.614(16)	1.0	4.164(-4)	3.464(13)
	11/9	1.038(-4)	6.360(14)	1.5	4.342(-4)	1.085(12)
	9/9	2.307(-4)	1.102(13)	2.0	4.634(-4)	1.059(11)
	7/9	5.189(-4)	1.305(11)	2.5	5.066(-4)	1.864(10)
17×17	15/17	4.469(-5)	1.454(19)	1.0	4.136(-6)	4.806(18)
	13/17	<u>4.756(-7)</u>	7.760(18)	2.0	<u>2.498(-6)</u>	4.190(15)
	11/17	7.822(-7)	5.132(18)	3.0	3.370(-6)	1.732(14)
	9/17	2.924(-6)	9.153(16)	4.0	4.865(-6)	1.725(13)
	7/17	1.344(-5)	5.349(14)	5.0	7.407(-6)	2.721(12)
33×33	11/33	6.382(-6)	9.656(19)	4.0	1.222(-7)	1.957(20)
	9/33	2.256(-5)	6.774(19)	5.0	<u>1.554(-8)</u>	2.005(18)
	7/33	<u>1.082(-6)</u>	3.598(19)	6.0	2.389(-8)	1.843(17)
	5/33	4.329(-6)	6.523(17)	7.0	3.878(-8)	3.591(16)
	3/33	1.549(-5)	7.397(13)	8.0	6.210(-8)	1.226(16)
65×65	5/65	9.659(-7)	1.204(22)	15	1.213(-8)	6.011(20)
	4/65	<u>7.014(-7)</u>	4.365(20)	20	<u>1.127(-8)</u>	3.828(18)
	3/65	2.533(-6)	1.893(20)	25	4.414(-8)	5.882(17)
	2/65	1.105(-4)	4.786(16)	30	2.111(-7)	1.131(17)
	1/65	4.847(-3)	3.162(12)	35	9.722(-7)	3.879(16)

Table 6.2: Radial basis function collocation of the Poisson equation with solution $\cos(4x_1 + 4x_2)$. Chebychev grid.

6.4 Solving the collocation system for large N

For globally supported basic functions directly solving the collocation system (6.4) requires $\mathcal{O}(N^3)$ operations and $\mathcal{O}(N^2)$ storage without using any customised method. This section presents a new algorithm for solving this system in $\mathcal{O}(N \log N)$ operations and $\mathcal{O}(N)$ storage with the multiquadric basic function. The algorithm uses a change of basis preconditioner in conjunction with domain decomposition and a fast matrix-vector multiply. The greatest computational cost at each iteration is at least one matrix-vector multiply. Fast matrix-vector product algorithms allow this to be achieved in $\mathcal{O}(N \log N)$ operations using a suitable fast evaluation code. These fast evaluation codes exist for a variety of functions [5, 7, 14].

6.4.1 Domain decomposition

This subsection considers a domain decomposition algorithm for centres separated into two domains. Without loss of generality refer to $X_I = \{x_1, \dots, x_{N_I}\}$ as good points and $X_B = \{x_{N_I+1}, \dots, x_N\}$ as bad points. Also assume that $N_I \gg |X_B| =: N_B$. Then if an interpolant, s , is of the form

$$s(\cdot) = \sum_{j=1}^N \lambda_j \Psi(\cdot - x_j),$$

where the coefficients λ_j are to be found then the interpolation matrix is $B_{ij} = \Psi(x_i - x_j)$. This can be split into the form

$$B = \begin{bmatrix} B_{II} & B_{IB} \\ B_{BI} & B_{BB} \end{bmatrix}. \quad (6.12)$$

In (6.12) B_{jk} , $j, k \in \{I, B\}$ has size $N_j \times N_k$ and is the matrix from evaluating the Ψ s centred at X_k , at points in X_j . Now applying a simple domain decomposition algorithm to this system we can iteratively obtain a solution. This method is given by Algorithm 6.4.1. The notation r_B, r_I refers to the residuals restricted to centres X_B and X_I respectively.

Algorithm 6.4.1 *domain – decomposition*(X, f, N_I)

SETUP

1. Create a set, X_I , of good points and a set, X_B of bad points
2. Form Ψ elements for each point x in X
3. $r \leftarrow f$ and $s \leftarrow 0$

ITERATIVE SOLUTION

1. **while** $\|r\| > \epsilon$
2. Solve for the coefficients μ of a bad point approximation via direct or approximate solutions of $B_{BB}\mu = r_B$
3. $s_{\text{bad}} \leftarrow \sum_{j:x_j \in X_B} \mu_j \Psi_j$
4. Evaluate $r_I = r_I - s_{\text{bad}}(X_I)$
5. Solve for the coefficients μ of a good point approximation via approximate solution of $B_{II}\mu = r_I$
6. $s_{\text{good}} \leftarrow \sum_{j:x_j \in X_I} \mu_j \Psi_j$
7. Update the RBF $s = s + s_{\text{bad}} + s_{\text{good}}$
8. Update the residual $r = f - s(X)$
9. **end while**

At the beginning of each iteration coefficients μ are found so that

$$\sum_{j: x_j \in X_B} \mu_j \Psi_j(x_i) = r_i, \quad x_i \in X_B.$$

Because N_B is small compared to N_I this is relatively efficient to solve. The residuals are then updated and a similar system on the good points is solved. Although N_I is large we can solve the system on the good points efficiently by GMRES if the eigenvalues of B_{II} are sufficiently clustered. Each GMRES iteration will require the computational cost of a matrix-vector multiply. In our experience an exact solution at step 5 is not required. Instead reducing the residual by a few orders of magnitude will suffice. If B_{II} is an approximation to the identity then most off diagonal elements will be near zero. An approximation to B_{II} can easily be found by retaining only a small number, say σ , of the largest magnitude entries per column. A matrix-vector product will then only require $\mathcal{O}(\sigma N)$ operations instead of the $\mathcal{O}(N \log N)$ required using a fast matrix-vector code. Numerical evidence shows that this approximation increases the number of outer iterations by less than four times but significantly decreases the total number of $\mathcal{O}(N \log N)$ matrix-vector products (which is the main computational cost of the algorithm).

The final step is to update the residual which can be achieved in $\mathcal{O}(N \log N)$ operations. Note that all matrix-vector multiplies will be $\mathcal{O}(N \log N)$ only if a suitable fast evaluation algorithm exists for the basic functions Φ and $L\Phi$. For example, if L is the Laplacian and Φ the multiquadric then

$$L\Phi(\cdot) = \frac{\|\cdot\|^2 + 2c^2}{(\|\cdot\|^2 + c^2)^{3/2}} = \frac{1}{(\|\cdot\|^2 + c^2)^{1/2}} + \frac{c^2}{(\|\cdot\|^2 + c^2)^{3/2}},$$

which is a combination of two members of the multiquadric family. Fast evaluators are available for functions of this type [7].

Algorithm 6.4.1 should be modified to include a coarse grid correction at each iteration. We usually take the number of points in the coarse grid to be about N_B .

6.4.2 Approximate cardinal functions

In the previous section it was assumed that the matrix B_{II} had clustered eigenvalues. In this section we achieve this by forming Ψ elements as approximate cardinal functions. We also explain why this approach doesn't work for large values of the multiquadric parameter c . Using approximate cardinal functions as a change of basis has been shown to be effective in the interpolation setting [6, 13]. The main difference in the collocation case is that there are different operators on the interior and on the boundary. If our aim was for B to be a good approximation to the identity, as in the interpolation case, then the Ψ_j 's would be of the form,

$$\begin{aligned}\Psi_j(x_i) &\approx 0, & x_i \in X_B, \\ L\Psi_j(x_i) &\approx 0, & x_i \in X_I,\end{aligned}$$

along with one of the constraints,

$$\begin{aligned}L\Psi_j(x_j) &= 1, & \text{if } x_j \in X_I, \\ \Psi_j(x_j) &= 1, & \text{if } x_j \in X_B.\end{aligned}$$

In our experience forming approximate cardinal functions to satisfy these conditions is difficult. Instead we form approximate cardinal functions which ensure B_{II} is a good approximation to the identity and use the domain decomposition approach given in Algorithm 6.4.1. The bad points are the boundary points and the good points are the interior points. For a uniform distribution of points in \mathcal{R}^2 the number of boundary points, N_B , is proportional to $N^{1/2}$ so direct solution of a linear system on these points requires $\mathcal{O}(N^{3/2})$ operations. Calculating the LU factorisation to B_{BB} as part of the setup means this cost is only incurred once. Subsequent use of this LU decomposition to solve a system requires $\mathcal{O}(N_B^2) = \mathcal{O}(N)$ operations.

Each Ψ element is of the form

$$\Psi_j(\cdot) = p_j(\cdot) + \sum_{i \in \mathcal{S}_j} \lambda_{ji} \Phi(\cdot - x_i),$$

where the set \mathcal{S}_j is often a set of indices of the nearest β points to x_j . For interior points we construct Ψ elements so that

$$\begin{aligned} L\Psi_j(x_j) &= 1, \\ L\Psi_j(x) &= \mathcal{O}(\|x - x_j\|^{-3}) \text{ as } \|x - x_j\| \rightarrow \infty. \end{aligned}$$

Approximate cardinal functions of this type are referred to as decay element approximate cardinal functions and are found by a constrained least squares problem as mentioned in [6].

The set X_j is defined to be the centres in X such that $x_i \in X_j$ if and only if $i \in \mathcal{S}_j$. For boundary points use a pure local approach of the form

$$\begin{aligned} \Psi_j(x_j) &= 1 \\ \Psi_j(x_i) &= 0, \quad x_i \in X_j \cap X_B, \\ L\Psi_j(x_i) &= 0, \quad x_i \in X_j \cap X_I. \end{aligned}$$

The pure local approximate cardinal functions are found by solving a collocation system on $|\mathcal{S}_j|$ nodes for each j .

In our experience we have noticed that creating approximate cardinal functions is only effective for $c < 2/\sqrt{N}$ (if the centres are a uniform grid in $[0, 1]^2$). In Figure 6.1 approximate cardinal functions formed using a decaying strategy for two different values of c are compared. Clearly the Ψ element formed with the larger value of c is not a very good approximation to a cardinal function. The required rate of decay is not achieved until further away from the centre of the Ψ element. An explanation for this is the regions of validity of the far field expansions. Consider finding a Ψ centred at x_j and based on centres with maximum distance H from x_j . The Ψ element will decay in the region of validity of the far field expansion of the cluster. This expansion is given in [7] and is valid outside the circle $\|x - x_j\| = \sqrt{H^2 + c^2}$. If c is large then the radius of this circle increases and the region of validity is further from x_j (see Figure 6.2) and thus the decay of Ψ occurs further away.

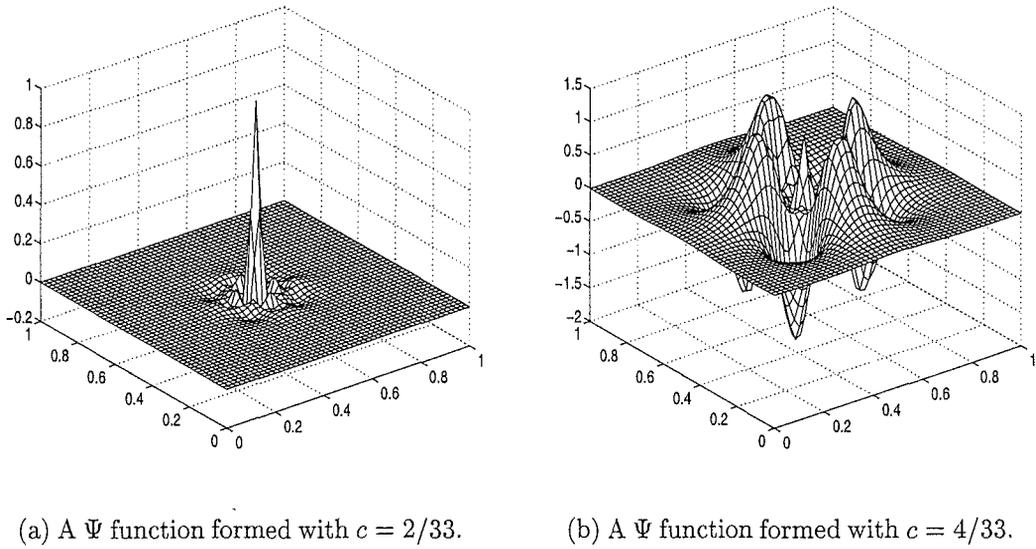


Figure 6.1: Ψ elements based on fifty local centres for two different values of c . Notice that the Ψ function decays quicker with the smaller value of c .

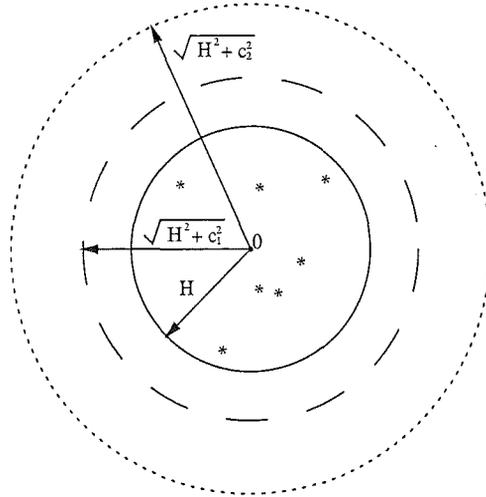


Figure 6.2: Far field expansion regions of validity for centres inside the circle $\|\cdot\| \leq H$ and multiquadric parameters c_1, c_2 with $c_2 > c_1$. The region of validity corresponding to c_2 is outside the dotted circle and for c_1 is outside the dashed circle.

6.5 Numerical results

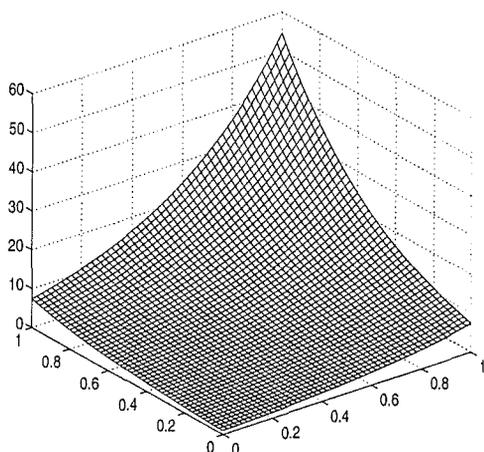
In this section we present numerical results for RBF collocation of linear PDEs of the form (6.1). These results show good convergence of the algorithm when the multiquadric parameter c is suitably small and constant. All numerical experiments are in the domain $[0, 1]^2$ with the collocation nodes forming an $n \times n$ grid. The iterations are stopped once the relative 2-norm residual is less than 10^{-8} .

To initially try this method we consider solving Poissons equation in \mathcal{R}^2 with the solutions

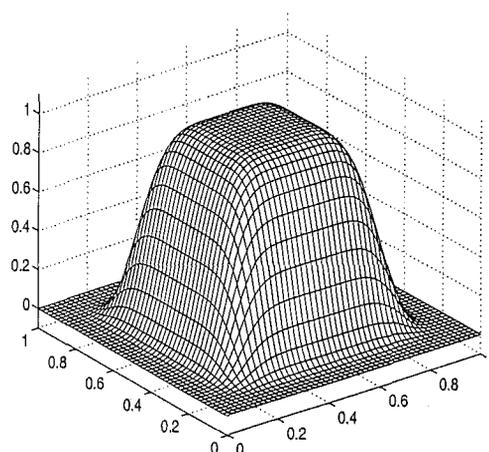
$$f_1(x_1, x_2) = \exp(2x_1 + 2x_2), \quad (6.13)$$

$$f_2(x_1, x_2) = \exp(-1000((x_1 - 1/2)^6 + (x_2 - 1/2)^6)). \quad (6.14)$$

RBF collocation solutions for these two PDEs can be found in Figure 6.3. The results



(a) Exact solution is f_1 .



(b) Exact solution is f_2 .

Figure 6.3: RBF collocation on a 33×33 grid of centres in $[0, 1]^2$ with $c = 2/33$.

from applying Algorithm 6.4.1 to f_1 and f_2 are in Tables 6.3 and 6.4 respectively. The algorithm was applied using both exact matrix-vector products and approximate matrix-vector products at step 5. We will refer to these different implementations as

Algorithm 6.4.1(a) and Algorithm 6.4.1(b) respectively. For both implementations exact matrix-vector products are always used at step 7. The “matrix-vector” column in the tables give the total number of exact matrix-vectors calculated to find the solution. For Algorithm 6.4.1(b) the total number of exact matrix-vector products is equal to the number of outer iterations. The “2-norm residual” column is the relative 2-norm residual $\|A\lambda - \bar{f}\|_2 / \|\bar{f}\|_2$ where \bar{f} is the right hand side vector $[f^T \ g^T]^T$ in (6.4). The tables show that the algorithm converges for both small and large values of N . As expected, Algorithm 6.4.1(b) requires more outer iterations to converge but the total number of exact matrix-vector products is reduced. This is a sizeable computational saving for large N .

Overall the number of outer iterations remained fairly stationary for Algorithm 6.4.1(a). When approximate matrix-vectors were used the number of outer iterations increased slightly but not dramatically as N increased. Thus it would be feasible to solve even larger systems using this algorithm.

From these experiments we can conclude that the algorithm will at least work on some simple PDEs when c is small. Usually c is required to be large for higher accuracy but in the case of f_2 we noticed that $c = 2/\sqrt{N}$ was nearly optimal for small data sets and using a Matlab \ operator to solve the systems. Carlson and Foley [19] suggest that a small shape parameter will be more accurate if the function values vary rapidly. The algorithm presented here may therefore be more applicable for solutions of this type.

A suitably modified algorithm has shown promising results for the nonlinear PDE given in [37, 36].

N	N_C	Exact matrix-vectors			Approximate matrix-vectors		
		Outer iterations	Matrix vectors	2-norm residual	Outer iterations	σ	2-norm residual
289	64	8	48	9.323(-9)	9	43.2	8.996(-9)
1089	121	8	48	1.413(-9)	14	44.6	6.745(-9)
4225	400	7	42	7.051(-9)	20	47.4	7.004(-9)
16641	625	9	54	2.409(-9)	27	48.2	8.539(-9)

Table 6.3: Results from Algorithm 6.4.1 on function f_1 . N_C is the number of coarse grid points and σ is the average number of non-zero elements per column in the approximation to B .

N	N_C	Exact matrix-vectors			Approximate matrix-vectors		
		Outer iterations	Matrix vectors	2-norm residual	Outer iterations	σ	2-norm residual
289	64	9	54	1.940(-9)	9	43.2	9.313(-9)
1089	121	7	42	1.576(-9)	16	44.6	8.487(-9)
4225	400	6	36	2.313(-9)	22	47.4	5.396(-9)
16641	625	7	42	2.174(-9)	28	48.2	6.754(-9)

Table 6.4: Results from Algorithm 6.4.1 on function f_1 . N_C is the number of coarse grid points and σ is the average number of non-zero elements per column in the approximation to B .

Bibliography

- [1] H. Akima, A method of bivariate interpolation and smooth surface fitting for irregularly distributed data points, *ACM Transactions on Mathematical Software*, **4** (1978), 148-159.
- [2] K. Ball, Eigenvalues of Euclidean distance matrices, *Journal of Approximation Theory*, **68** (1992), 74-82.
- [3] K. Ball, N. Sivakumar and J. D. Ward, On the sensitivity of radial basis interpolation to minimal data separation distance, *Constructive Approximation*, **8** (1992), 401-426.
- [4] I. Barrodale, D. Skea, M. Berkley, R. Kuwahara and R. Poeckert, Warping digital images using thin-plate splines, *Pattern Recognition*, **26** (1993), 375-376.
- [5] R. K. Beatson and E. Chacko, Fast evaluation of radial basis functions: A multivariate momentary evaluation scheme, to appear in *Curve and Surface Fitting: Saint Malo 1999*, A. Cohen, C. Rabut and L.L. Schumaker (eds), Vanderbilt Univ. Press, Nashville (2000).
- [6] 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** (1999), 253-270.

- [7] R. K. Beatson, J. B. Cherrie and G. N. Newsam, Fast evaluation of radial basis functions: Methods for generalised multiquadrics in \mathcal{R}^n . Manuscript (2001).
- [8] R. K. Beatson, J. B. Cherrie and D. L. Ragozin. Fast evaluation of radial basis functions: Methods for four-dimensional polyharmonic splines, *SIAM Journal on Mathematical Analysis*, **32** (2001), 1272-1310.
- [9] R. K. Beatson, G. Goodsell and M. J. D. Powell, On multigrid techniques for thin plate spline interpolation in two dimensions, *Lectures in Applied Mathematics*, **32** (1996), 77-97.
- [10] R. K. Beatson and L. Greengard, A short course on fast multipole methods, *Wavelets, Multilevel Methods and Elliptic PDEs*, M. Ainsworth, J. Levesley, W. A. Light and M. Marletta (eds), Oxford University Press (1997), 1-37.
- [11] R. K. Beatson and W. A. Light, Fast evaluation of radial basis functions: Methods for 2-dimensional polyharmonic splines, *IMA Journal on Numerical Analysis*, **17** (1997), 343-372.
- [12] R. K. Beatson, W. A. Light and S. Billings, Fast solution of the radial basis function interpolation equations: Domain decomposition methods, *SIAM Journal on Scientific Computing*, **22** (2000), 1717-1740.
- [13] R. K. Beatson and C. T. Mouat, Fast Kriging, *University of Canterbury Research Report number UCDMS2000/2*, (2000).
- [14] R. K. Beatson and G. N. Newsam, Fast evaluation of radial basis functions: I, *Computers and Mathematics with Applications*, **24**, No. 12 (1992), 7-19.
- [15] R. K. Beatson and G. N. Newsam, Fast evaluation of radial basis functions: moment-based methods, *SIAM Journal on Scientific and Statistical Computing*, **19** (1998), 1428-1449.

- [16] R. K. Beatson and M. J. D. Powell, An iterative method for thin-plate spline interpolation that employs approximations to the Lagrange functions, *Numerical Analysis 1993*, D. F. Griffiths and G. A. Watson eds, Longmans, Harlow, (1994).
- [17] S. D. Billings, R. K. Beatson and G. N. Newsam, Interpolation of geophysical data with continuous global surfaces. Manuscript (2000).
- [18] S. L. Campbell, I. C. F. Ipsen, C. T. Kelley and C. D. Meyer, GMRES and the minimal polynomial, *BIT*, **36** (1996), 664-675.
- [19] R. E. Carlson and T. A. Foley, The parameter R^2 in multiquadric interpolation, *Computers and Mathematics with Applications*, **21** (1991), 29-42.
- [20] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum and T. R. Evans, Reconstruction and representation of 3D objects with radial basis functions, to appear *SIGGRAPH 2001*, August 2001.
- [21] J. C. Carr, W. R. Fright and R. K. Beatson, Surface interpolation with radial basis functions for medical imaging, *IEEE Transactions on Medical Imaging*, **16** (1997), 96-107.
- [22] W. Cheney and W. Light, *A Course in Approximation Theory*, Brooks/Cole Series in Advanced Mathematics (1999).
- [23] N. H. Christ, R. Friedberg and T. D. Lee, Weights of links and plaquettes in a random lattice, *Nuclear Physics B* **210** (1982), 337-346.
- [24] C. K. Chui, *Multivariate Splines*, CBMS-NSF Regional Conference Series in applied mathematics 54, Society for Industrial and Applied Mathematics, Pennsylvania (1988).
- [25] N. Cressie, The origins of Kriging, *Mathematical Geology*, **22** (1990), 239-252.
- [26] N. Cressie, *Statistics for Spatial Data*, Wiley, New York (1993).

- [27] O. Dubrule, Two methods with different objectives: Splines and Kriging, *Mathematical Geology*, **15** (1983), 245-257.
- [28] O. Dubrule, Comparing splines and Kriging, *Computers and Geosciences*, **10** (1984), 327-338.
- [29] J. Duchon, Splines minimizing rotation-invariant seminorms in Sobolev spaces, *Constructive Theory of Functions of Several Variables, Lecture Notes in Mathematics*, **571**, W. Schempp and K. Zeller (eds), Springer-Verlag, Berlin (1977), 85-100.
- [30] N. Dyn and D. Levin, Iterative solutions of systems originating from integral equations and surface interpolation, *SIAM Journal on Numerical Analysis*, **20** (1983), 377-390.
- [31] N. Dyn, D. Levin and S. Rippa, Numerical procedures for surface fitting of scattered data by radial functions, *SIAM Journal on Scientific and Statistical Computing*, **7** (1986), 639-659.
- [32] N. Dyn, D. Levin and S. Rippa, Data dependent triangulations for piecewise linear interpolation, *IMA Journal of Numerical Analysis*, **10** (1990), 137-154.
- [33] G. E. Fasshauer, On the numerical solution of differential equations with radial basis functions. *Boundary Element Technology XIII*, C. S. Chen, C. A. Brebbia, and D. W. Pepper (eds), WIT Press, (1999), 291-300.
- [34] G. E. Fasshauer, Solving partial differential equations by collocation with radial basis functions. *Surface Fitting and Multiresolution Methods*, A. Le Mehaute, C. Rabut, and L. L. Schumaker (eds.), Vanderbilt University Press, (1997), 131-138.
- [35] G. E. Fasshauer and J. W. Jerome, Multistep approximation algorithms: improved convergence rates through postconditioning with smoothing kernels. *Advances in Computational Mathematics*, **10** (1999), 1-27.

- [36] G. E. Fasshauer, Nonsymmetric multilevel RBF collocation within an operator newton framework for nonlinear PDEs, to appear in *Trends in Approximation Theory*.
- [37] G. E. Fasshauer, E. C. Gartland and J. W. Jerome, Newton iteration for partial differential equations and the approximation of the identity, *Numerical Algorithms*, **25** (2000), 181-195.
- [38] J. Flusser, An adaptive method for image registration, *Pattern Recognition* **25** (1992), 45-54.
- [39] R. Franke, Scattered data interpolation: tests of some methods, *Mathematics of Computation*, **38** (1982), 181-200.
- [40] C. Franke and R. Schaback, Convergence order estimates of meshless collocation methods using radial basis functions, *Advances in Computational Mathematics*, **8** (1998), 381-399.
- [41] C. Franke and R. Schaback, Solving partial differential equations by collocation using radial basis functions, *Applied Mathematics and Computation*, **93** (1998), 73-82.
- [42] T. Gneiting, Closed form solutions of the two-dimensional turning bands equation, *Mathematical Geology*, **30** (1998), 379-390.
- [43] L. L. Greengard and V. Rokhlin, A fast algorithm for particle simulations, *Journal of Computational Physics*, **73** (1987), 325-348.
- [44] R. L. Hardy, Theory and applications of the multiquadric-biharmonic method, *Computers and Mathematics with Applications* **19** (1990), 163-208.
- [45] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia (1996).

- [46] Y. C. Hon, A RBFs method for solving options pricing model, *Proceedings of Advances in Scientific Computing and Modeling*, Alicante, Spain (1998), 210-230.
- [47] Y. C. Hon and Z. Wu, Error estimation on using radial basis functions for solving option pricing models. Manuscript.
- [48] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press (1990).
- [49] M. F. Hutchinson, On thin plate splines and Kriging, *Computing Science and Statistics*, **25** (1993), 55-62.
- [50] M. F. Hutchinson and P. E. Gessler, Splines - more than just a smooth interpolator, *Geoderma*, **62** (1994), 45-67.
- [51] E. J. Kansa, Multiquadrics - a scattered data approximation scheme with applications to computational fluid-dynamics-II, *Computers and Mathematics with Applications*, **19** (1990), 147-161.
- [52] E. J. Kansa and Y. C. Hon, Circumventing the ill-conditioning problem with multiquadric radial basis functions: applications to elliptic partial differential equations, *Computers and Mathematics with Applications*, **39** (2000), 123-137.
- [53] C. T. Kelley, *Iterative Methods for Linear and Non-linear Equations*, Society for Industrial and Applied Mathematics (1995).
- [54] C. L. Lawson, Software for C^1 surface interpolation, in *Mathematical Software 3*, ed. J. R. Rice, Academic Press, New York (1977).
- [55] W. Light and H. Wayne, On power functions and error estimates for radial basis function interpolation, *Journal of Approximation Theory*, **92** (1998), 245-266.
- [56] W. R. Madych, Miscellaneous error bounds for multiquadric and related interpolators, *Computers and Mathematics with Applications*, **24** (1992), 121-138.

- [57] W. R. Madych and S. A. Nelson, Multivariate interpolation: a variational theory, Manuscript, (1983).
- [58] B. Matern, *Spatial Variation*, Lecture Notes in Statistics v36, Springer-Verlag, New York (1986).
- [59] A. B. McBratney and R. Webster, Choosing functions for semi-variograms of soil properties and fitting them to sampling estimates, *Journal of Soil Science*, **37** (1986), 617-639.
- [60] C. A. Michelli, Interpolation of scattered data: Distance matrices and conditionally positive definite functions, *Constructive Approximation*, **2** (1986), 11-22.
- [61] D. E. Myers, Kriging, cokriging, radial basis functions and the role of positive definiteness, *Computers and Mathematics with Applications*, **24** (1992), 139-148.
- [62] D. E. Myers, Smoothing and interpolation with radial basis functions, in *Boundary Element Technology XIII*, C. S. Chen, C. A. Brebbia and D. W. Pepper (eds), WIT Press, Southhampton, (1999), 365-374.
- [63] F. J. Narcowich and J. D. Ward, Norms of inverses and condition numbers for matrices associated with scattered data, *Journal of Approximation Theory*, **64** (1991), 69-94.
- [64] F. J. Narcowich and J. D. Ward, Norm estimates for the inverse of a general class of scattered-data radial-function interpolation matrices, *Journal of Approximation Theory*, **69** (1992), 84-109.
- [65] F. J. Narcowich, N. Sivakumar and J. D. Ward, On condition numbers associated with radial-function interpolation, *Journal of Mathematical Analysis and Applications*, **186** (1994), 457-485.

- [66] C. C. Paige, Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix, *Journal of the Institute of Mathematics and its Applications*, **18** (1976), 341-349.
- [67] M. J. D. Powell, The theory of radial basis function approximation in 1990 in *Advances in Numerical Analysis II: Wavelets, subdivision algorithms and radial functions*, W. Light ed., Oxford University Press, Oxford, UK, (1992), 105-210.
- [68] M. J. D. Powell, The uniform convergence of thin plate spline interpolation in two dimensions, *Numerische Mathematik*, **68** (1994), 107-128.
- [69] M. J. D. Powell, Some algorithms for thin plate spline interpolation to functions of two variables, *Advances in Computational Mathematics: New Delhi, India.*, H. P. Dikshit and C. A. Micchelli (eds), World Scientific Publishing Co. (1994).
- [70] M. Powell and M. Sabin, Piecewise quadratic approximations, *ACM Transactions on Mathematical Software*, **3** (1977), 3316-325.
- [71] B. D. Ripley, *Spatial Statistics*, Wiley series in probability and mathematical statistics, Wiley, New York (1981).
- [72] Y. Saad and M. H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear equations, *SIAM Journal on Scientific and Statistical Computing*, **7** (1986), 856-869.
- [73] R. Schaback, Error estimates and condition numbers for radial basis function interpolation, *Advances in Computational Mathematics*, **3** (1995), 251-264.
- [74] R. Schaback, Multivariate interpolation and approximation by translates of a basis function, *Approximation Theory VIII*, C. K. Chui, L. L. Schumaker (eds), Vanderbilt University Press (1996).
- [75] R. Schaback and Y. C. Hon, On unsymmetric collocation by radial basis functions, *Journal of Applied Mathematics and Computation*, **119** (2001), 177-186.

- [76] I. J. Schoenberg, Metric spaces and completely monotone functions, *Annals of Mathematics*, **39** (1938), 811-841.
- [77] R. Sibson and G. Stone, Computation of Thin-Plate Splines, *SIAM Journal on Scientific and Statistical Computing*, **12** (1991), 1304-1313.
- [78] M. L. Stein, *Interpolation of Spatial Data: Some Theory for Kriging*, Springer series in Statistics, Springer (1999).
- [79] G. Szego, *Orthogonal Polynomials*, American Mathematical Society, Rhode Island (1978).
- [80] F. Trochu, A contouring program based on dual Kriging interpolation, *Engineering and Computers*, **9** (1993), 160-177.
- [81] R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, New Jersey (1962).
- [82] A. V. Vecchia, Estimation and model identification for continuous spatial processes, *Journal of the Royal Statistical Society Series B*, **50** (1988), 297-312.
- [83] G. Wahba, *Spline Models for Observational Data*, CBMS-NSF Regional Conference Series in applied mathematics 59, Society for Industrial and Applied Mathematics, Pennsylvania (1990).
- [84] H. F. Walker, Implementation of the GMRES method using Householder transformations, *SIAM Journal on Scientific and Statistical Computing*, **9** (1988), 152-163.
- [85] R. Wang, The structural characterization and interpolation for multivariate splines, *Acta Mathematica Sinica*, **18** No. 2 (1975), 10-39.
- [86] G. N. Watson, *A Treatise on the Theory of Bessel functions*, Cambridge University Press, Cambridge (1944).

- [87] G. S. Watson, Smoothing and interpolation by Kriging and with splines, *Mathematical Geology*, **16** (1984), 601-615.
- [88] H. Wendland, Error estimates for interpolation by compactly supported radial basis functions of minimal degree, *Journal of Approximation Theory*, **93** (1998), 258-272.
- [89] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, *Advances in Computational Mathematics*, **4** (1995), 389-396.
- [90] D. V. Widder, *The Laplace Transform*, Princeton University Press, Princeton (1946).
- [91] Z. Wu, Hermite-Birkhoff interpolation of scattered data by radial basis functions, *Approximation Theory and its Applications*, **8** (1992), 1-10.
- [92] Z. Wu and R. Schaback, Local error estimates for radial basis function interpolation of scattered data, *IMA Journal of Numerical Analysis*, **13** (1993), 13-27.
- [93] D. L. Zimmerman and M. B. Zimmerman, A comparison of spatial semivariogram estimators and corresponding ordinary Kriging predictors, *Technometrics*, **33** (1991), 77-91.