
Roussos, George and Baxter, Brad J.C. (2005). Rapid evaluation of radial basis functions. *Journal of Computational and Applied Mathematics* 180 (1) 51-70.

Rapid Evaluation of Radial Basis Functions

September 30, 2004

Abstract

Over the past decade, the radial basis function method has been shown to produce high quality solutions to the multivariate scattered data interpolation problem. However, this method has been associated with very high computational cost, as compared to alternative methods such as finite element or multivariate spline interpolation. For example, the direct evaluation at M locations of a radial basis function interpolant with N cedntres requires $\mathcal{O}(MN)$ floating-point operations. In this paper we introduce a fast evaluation method based on the Fast Gauss Transform and suitable quadrature rules. This method has been applied to the Hardy Multiquadric, the Inverse Multiquadric and the Thin-plate Spline to reduce the computational complexity of the interpolant evaluation to $\mathcal{O}(M + N)$ floating-point operations. By using certain localisation properties of conditionally negative definite functions this method has several performance advantages against traditional hierarchical rapid summation methods which we discuss in detail.

1 Introduction

A problem frequently occurring in science and engineering is the approximation of a function f, the value of which is known only on a relatively small set of points. One way to obtain such an approximation is by interpolation: Given the values f_i of f at the points $x_i \in \mathbb{R}^d$, $i = 1, 2, \ldots, N$, determine a function s that satisfies the conditions

$$s(x_i) = f(x_i) = f_i, \quad i = 1, 2, \ldots, N.$$

Usually, the choice of solution method specifies a class of functions S, with the interpolant $s \in S$ uniquely identified by computing a number of free parameters, so that s satisfies the interpolation conditions, and possibly meets further restraints or has particular properties required by the application.

In the one-dimensional case, the graph of f belongs to the two-dimensional Euclidean space and the problem may be restated in geometric terms. Given a set of points p_i, $i = 1, 2, \ldots, N$, from an unknown target curve, construct a curve which approximates the original, in the sense that it passes through all the data points. A common solution for this problem is cubic spline interpolation, that is, choosing an interpolant from the space S of piecewise polynomials of degree three that have a smooth first derivative and a continuous second derivative both within and at the boundary of the interpolation interval.

For the two-dimensional case, a comparative study [16] of interpolation methods indicated that the most accurate and visually attractive results are produced by the so-called Hardy Multiquadric and the Thin-plate Spline methods. At that time, only numerical evidence supported the suitability of these methods for interpolation. Since then a significant amount of analytical work has been carried out and today radial basis functions are a well established method of multivariate scattered data interpolation.

1.1 Interpolation with Radial Basis Functions

Radial basis function interpolants have the form

$$s(y) = \sum_{i=1}^{N} \lambda_i \varphi(\|y - x_i\|), \tag{1}$$

with λ_i real coefficients, x_i points in $\mathbb{R}^d$ called *centers*, $\|\cdot\|$ the Euclidean norm and φ the *basis function*. The function $\varphi : \mathbb{R}^+ \to \mathbb{R}$ is univariate and radially symmetric with respect to the norm, in the sense that it has the symmetries of the unit ball in $\mathbb{R}^d$. The coefficients λ_i are chosen so that the interpolation conditions are satisfied. Recently, different p-norms have been considered in the literature, but here we will discuss only the Euclidean norm, since it is the one used in the majority of applications.

Useful choices of φ include the *Gauss kernel*

$$\varphi(r) = e^{-c\, r^2},$$

the *Euclidean distance*

$$\varphi(r) = r,$$

the *Hardy Multiquadric*

$$\varphi(r) = \sqrt{r^2 + c^2}, \tag{2}$$

the *Inverse Multiquadric*

$$\varphi(r) = \frac{1}{\sqrt{r^2 + c^2}}, \tag{3}$$

and the *Thin-plate spline*

$$\varphi(r) = r^2 \log r. \tag{4}$$

The radial basis function method can be thought of as an extension of univariate splines to several variables. Assuming that the points x_i are organised in ascending order, the linear spline is composed by the line segments

$$s(y) = \frac{(x_{i+1} - y) f_i + (y - x_i) f_{i+1}}{x_{i+1} - x_i}, \quad x_i \le y \le x_{i+1},$$

or else

$$s(y) = \sum_{i=1}^{N} \lambda_i |y - x_i|,$$

the λ_i's being defined by the interpolation equations.

In matrix notation, solving the radial basis function interpolation problem is equivalent to solving the system of linear equations

$$A\lambda = f, \tag{5}$$

where A is the *interpolation matrix*

$$A_{ij} = \varphi(\|x_i - x_j\|), \tag{6}$$

λ is the vector of coefficients $(\lambda_1 \ \lambda_2 \ \dots \ \lambda_N)^T$ and f is the vector of function values $(f_1 \ f_2 \ \dots \ f_N)^T$.

One of the features that makes radial basis function interpolation a useful technique is the fact that a unique interpolant is guaranteed under weak conditions on the location of the centers. These properties have been discussed in detail in [29] and are also summarized in this paper in section 2.2. One exception to this is the Thin-plate Spline whose corresponding interpolation matrix A in (5) may be singular. For example, if the centers x_i for $i = 2, \dots, N$, are points on the circle of unit radius in $\mathbb{R}^2$ and x_1 is the center of the circle, then the first row and column of A consist entirely of zeros. This difficulty may be resolved by adding a polynomial of degree one to the Thin-plate Spline interpolant and then demand that the centers are unisolvent, that is the only polynomial of degree one which vanishes at every center is the zero polynomial. Of course, extra conditions must be introduced because of the extra degrees of freedom added. The interpolation conditions become

$$\begin{array}{rcll} \sum_{j=1}^{N} \lambda_j \varphi(\|x_i - x_j\|) + p(x_i) & = & f_i, & i = 1, 2, \dots, N, \\ \sum_{j=1}^{N} \lambda_j p(x_j) & = & 0, & \text{for every } p \in \Pi_1(\mathbb{R}^2). \end{array} \tag{7}$$

with $\Pi_1(\mathbb{R}^2)$ the set of polynomials in two real variables of total degree less than or equal to one. In this case, there is a unique vector λ and a unique polynomial p satisfying (7).

For the d-dimensional case, the resulting linear system is

$$\begin{pmatrix} A & Q \\ Q^T & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ h \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} \tag{8}$$

with A defined in (6), and Q defined by

$$Q = \begin{pmatrix} p_1(x_1) & \dots & p_{d_m}(x_1) \\ \vdots & \vdots & \vdots \\ p_1(x_n) & \dots & p_{d_m}(x_n) \end{pmatrix},$$

where $\{p_k \ : \ k = 1, 2, \dots, d_m\}$ is a basis of $\Pi_m(\mathbb{R}^d)$ and

$$d_m = \dim \Pi_m(\mathbb{R}^d) = \binom{d+m}{d}.$$

Two-dimensional Thin-plate Spline interpolation has extra advantages: the resulting interpolant is optimal in the sense that it is the differentiable function minimising the integral

$$\int_{\mathbb{R}^2} |f_{xx}|^2 + 2|f_{xy}|^2 + |f_{yy}|^2 \mathrm{d}x\mathrm{d}y.$$

In addition to providing an effective way to define the degrees of freedom in the general interpolation problem, this property implies smoothness for the resulting surface. Further, the interpolant is rotation and scale-invariant and thus one obtains the same surface independent of the physical units used.

1.2 Computation with Radial Basis Functions

Although the radial basis function method has been applied successfully in a wide variety of scientific and engineering problems, its widespread adoption has been hindered by the associated high computational cost. Indeed, the solution of the interpolation equations (5) directly requires $\mathcal{O}(N^3)$ floating point operations, while the form of the interpolant (1) implies that evaluating s at M points $y_1, y_2, \ldots, y_M$, directly requires $\mathcal{O}(MN)$ operations.

Indeed, the high computational cost associated with the radial basis function method (compared against alternative methods such as finite elements or multivariate splines) has been identified as early as Franke's survey [16]. An initial attempt to reduce the computational complexity of the evaluation task was made by Powell [33]. The minimisation property of the Thin-plate Spline was used to devise a fast method to evaluate interpolants on a grid. A simpler solution to the same problem but for any radial basis function was suggested by Arad [3]: when the centers of interpolation are on a grid then there are far less than $\mathcal{O}(N^2)$ distinct inter-point distances, which may be precomputed. Then, the evaluation task may be performed by table lookups rather than floating point computation.

Of course, a new approach is required if the interpolant is not to be evaluated on a grid but at scattered points. Beatson and Newsam [8] first noted the similarities between the computational structure of the N-body problem and the evaluation of a Thin-plate Spline (and in consequence, of any other radial basis function) interpolant, thus initiating research in the construction of fast evaluation methods, based on the extensive research literature on rapid N-body simulations. In [8], the Laurent and Taylor expansions required by the Fast Multipole Method (FMM) of Greengard and Rokhlin [23, 24] were constructed. These results[1] were used by Powell [34] to construct a fast algorithm for the evaluation of Thin-plate Spline interpolants. This method uses a decomposition of the set of interpolation centers similar to Appel [2] and the method has observed computational complexity of $\mathcal{O}(N \log N)$ for fairly uniform distributions of centers. More recently, Beatson has implemented a full FMM based on the results of [8] with reported performance characteristics similar to that discussed in [23]. A variant of this method whereby the coefficients of the multipole expansion are not calculated directly but approximated is discussed by Suter [44]. Finally, in [11] the multipole expansions required for computation with generalised multiquadrics have been calculated.

The relation between the Hardy Multiquadric and N-body computations has received a physical justification. Indeed, Hardy [28] relates the solution of an interpolation problem to simulation of the Earth's geomagnetic field by a biharmonic potential. This has the advantage that the Earth is considered as a solid rather than a hollow body, as happens when using the harmonic potential.

A rapid algorithm for evaluation of radial basis function interpolants may reduce the computational cost of certain methods used for the fast solution of the interpolation equations. Indeed, Baxter [5] points out that

[1] Note that the Laurent expansions calculated in [34] are erroneous. Formula (3.7) of [34] must be corrected using the multipole expansions calculated in Theorem 4.2 of [8]: Following the notation of [34], replace $|\mathcal{J}|$ (the total number of centers) with $\sum_{i=1}^{N} \lambda_i$ (the sum of the coefficients of the interpolant). However, the numerical results reported in the final section of [34] are correct.

when the centers form a regular grid, matrix-vector products Ax with A is the interpolation matrix, may be computed in $\mathcal{O}(N \log N)$ operations via the Fast Fourier Transform. Products of this kind are required in iterative methods, like the Conjugate Gradient algorithm.

In fact, the preconditioned CG method was used by Dyn, Levin and Rippa [13, 14] to solve the interpolation equations. However, since the CG algorithm requires a positive (or negative) definite matrix, the method is unstable for most radial basis functions of interest. A remedy for this situation for the Hardy Multiquadric was proposed by Baxter [5, Chapter 6], where at the end of each iteration of the CG method the residual is projected onto the space $< e >^T$ with $e = (1\ 1\ \ldots\ 1)$, where the interpolation matrix is indeed negative definite. The modified method ensures the stability of the conjugate gradient method. More recently, in a series of papers Beatson, Powell, Goodsell and Faul [9, 10, 20, 15, 35] have developed an iterative algorithm which employs estimates of the characteristic function.

In this paper, we introduce a method for the rapid evaluation of radial basis function interpolants that can be used for all usual radial basis functions discussed in this and previous sections. In addition to its general applicability and contrary to FMM and treecodes this method is non-hierarchical. This fact allows for very scalable implementations on high performance computers, in particular multiprocessors and clusters of workstations. The remainder of this paper is organized as follows: in section two, first we discuss the Fast Gauss Transform of Greengard and Strain [25] and then we summarize some properties of radial basis functions, which form the core of our algorithm. Then, in section three we proceed to present the rapid evaluation algorithm itself as well as its implementation. The paper concludes with a report on some numerical experiments and with observations on the performance of our method.

2 Preliminaries

Our rapid evaluation method for radial basis functions is based on certain integral representations (such representations are established in Theorems 1, 2 and 3 for the most common radial basis functions) which are used with the Fast Gauss Transform and suitable quadrature rules, to estimate the value of the interpolant at a point. In the following paragraphs we discuss the individual building blocks of the algorithm.

2.1 The Fast Gauss Transform

Here, we briefly discuss the Fast Gauss Transform (FGT) of Greengard and Strain [25] as appropriate for the fast radial basis function evaluation method. In the d-dimensional FGT we require the use of multi-index notation. The d-tuple $\beta = (\beta_1, \beta_2, \ldots, \beta_d) \in \mathbb{N}^d$ is called a *multi-index* and is useful for indexing in the context of a d-dimensional Euclidean space. We use the notation $\beta \geq p$, $p \in \mathbb{N}$, if $\beta_h \geq p$ for $1 \leq h \leq d$. Thus, for any $x \in \mathbb{R}^d$, we define

$$\begin{aligned} |\beta| &= \beta_1 + \beta_2 + \ldots + \beta_d, \\ \beta! &= \beta_1! \beta_2! \ldots \beta_d!, \\ x^\beta &= x_1^{\beta_1} x_2^{\beta_2} \ldots x_d^{\beta_d}. \end{aligned}$$

The multi-dimensional Hermite functions $h_\beta(x)$ are

$$h_\beta(x) = h_{\beta_1}(x_1)h_{\beta_2}(x_2)\dots h_{\beta_d}(x_d) \tag{9}$$

where h_k with $k \in \mathbb{N}$ is a classical Hermite function and $x = (x_1, x_2, \dots, x_d)$.

The aim of the FGT is to evaluate efficiently sums of the form

$$\sum_{i=1}^{N} \lambda_i \exp\left(-s\|y - x_i\|^2\right)$$

at M distinct points $y_1, y_2, \dots, y_M$. By shifting the origin and re-scaling, we may assume that all the interpolation centers and all the evaluation points lie within the unit hypercube $B_0 = [0,1]^d$. This is a a convenient normalisation and does not restrict the generality of the method.

We may express a Gaussian in $\mathbb{R}^d$ as the Hermite expansion

$$e^{-s\|y-x\|^2} = \sum_{\beta \geq 0} \frac{1}{\beta!} \left(\sqrt{s}(x - C)\right)^\beta h_\beta \left(\sqrt{s}(y - C)\right). \tag{10}$$

For centers $x_1, x_2, \dots, , x_N \in \mathbb{R}^d$ inside box B $B = \{y \in [0,1]^d : \|y - C\|_\infty < r/\sqrt{2s}\}$ of side length $r\sqrt{2/s}$ for some $r < 1$ (cf. [25, Lemma 2.1]) centred at C, we can precompute the moments

$$A_\beta = \frac{1}{\beta!} \sum_{i=1}^{N} \lambda_i \left(\sqrt{s}(x_i - C)\right)^\beta, \tag{11}$$

which we can then use to evaluate the Gaussian sum at a point y by

$$\sum_{i=1}^{N} \lambda_i \exp\left(-s\|y - x_i\|^2\right) = \sum_{\beta \geq 0} A_\beta h_\beta \left(\sqrt{s}(y - C)\right). \tag{12}$$

By truncating the infinite series on the right of (12) after the first p^d terms, we introduce the error

$$|E_p| \quad \leq \quad \frac{Q}{(1-r)^d} \sum_{k=0}^{d-1} \binom{d}{k} (1 - r^p)^k \left(\frac{r^p}{\sqrt{p!}}\right)^{d-k}, \tag{13}$$

with $Q = \sum_{i=1}^{N} |\lambda_i|$. Note that this error estimate (calculated in detail in [38]) and the error estimate of [25, Lemma 2.1] coincide when $d = 1$ but they are distinct in higher dimensions. Of course, similar reasoning can be directly applied to Lemmata 2.2 and 2.3 of [25] to obtain the corresponding error estimates.

Thus, the first ingredient of the FGT is the approximation (12) to the Gaussian in terms of the moments (11) of the centers which when truncated after p^d terms introduces the associated error estimate (13). The second ingredient of the FGT is the decomposition of the computational space B_0 into subboxes B of side length $r\sqrt{2/s}$ parallel to the axes, for some fixed parameter r. Each centre is assigned to the subbox B that contains it and contributes only to the p^d moments of subbox B. At the end of the precomputation step, the p^d moments for each of the subboxes B have been computed. The precomputation requires $\mathcal{O}(p^d N)$ operations.

For the estimation of the FGT at a particular evaluation point y contained in subbox D, we need to consider the influence of only some of the nearest neighbour boxes of D. Indeed, due to the exponential decay

```
 1: choose r and p to guarantee the required precision
 2: subdivide B_0 into boxes B of side length at most r√(2/s)
 3: for each centre x_j do
 4:    find the box C that contains x_j
 5:    compute the contribution of x_i to the moments A_k of C defined by (11)
 6: end for
 7: for each evaluation point y_i do
 8:    find the box D that contains y_i
 9:    for all (2n + 1)^d nearest neighbours of D do
10:       accumulate the sums (12)
11:    end for
12: end for
```

Algorithm 1: The Fast Gauss Transform. Accepts as input the parameter s, N centers x_i with the associated weights λ_i and M evaluation points y_j, and returns the value of (10) at the points y_j.

of the Gauss kernel, its effect on subboxes away from its centre may be insignificant within certain accuracy. For example, taking into account only the $(2n+1)^d$ nearest neighbours to D, introduces error bounded by $Qe^{-2r^2n^2}$. Hence, for $r = 1/2$ and $n = 6$ relative accuracy of 10^{-7} is obtained. We will call the set of $(2n+1)^d$ nearest neighbours the *interaction list* of box D. Thus, in order to estimate the Gaussian sum on the left side of (12) at point y, we have to accumulate the p^d moments for each of the boxes B in the interaction list of D. Evaluation at a single point requires $\mathcal{O}((2n+1)^d p^d)$ operations.

Overall, the calculation of (10) may be performed using Algorithm 1. Step 5 requires $\mathcal{O}(p^d)$ operations per centre and thus Step 3 requires $\mathcal{O}(p^d N)$ operations overall. Step 10 requires $\mathcal{O}(p^d)$ per evaluation point per box and thus Step 7 requires $\mathcal{O}(p^d(2n+1)^d M)$ in total. Hence, the computational complexity of the algorithm is $\mathcal{O}(p^d N + p^d(2n+1)^d M)$.

2.2 Conditionally Negative Definite Functions

As noted earlier, radial basis function interpolation is possible under relatively weak conditions on the positions of the centers. To construct a radial basis function interpolant it is necessary to solve the linear system of equations (5) on page 3, but so far we have not commented on the non-singularity of the matrix A. Moreover, in the case of the Thin-plate Spline solvability of the interpolation problem requires that the matrix A in (8) on page 3 be non-singular for all vectors λ satisfying (7), and the polynomials $p \in \Pi_m(\mathbb{R}^d)$ be uniquely determined by their value on $X = \{x_1, \ldots, x_N\}$, that is if $p(x_i) = 0$ for all $x_i \in X$ then $p = 0$. Micchelli [30] answers this question by proving almost positive (or negative) definiteness of the interpolation matrix for several of the radial basis functions, including the Hardy Multiquadric, the Inverse Multiquadric and the Thin-plate Spline. In doing so, Micchelli constructs integral forms for these radial basis functions, which we will use to develop our rapid evaluation method.

To state Micchelli's results about conditionally positive (or negative) functions we recall that a function f is said to be *completely monotonic* on $(0, \infty)$ provided that it is in $C^\infty(0, \infty)$ and $(-1)^m \cdot f^{(m)}(x) \geq 0$ for $x \in$

$(0, \infty)$ and $m = 0, 1, 2, \ldots$. Also, a real valued function f of a real variable is said to be *positive definite* if the inequality $\sum_{i,j=1}^{N} \lambda_i \lambda_j f(x_i - x_j) \geq 0$ holds for every choice of the real numbers x_i and λ_i. Schoenberg generalised this definition so that for example, a radially symmetric function $\varphi(\|x\|_2) = \Phi(x)$, $\Phi : \mathbb{R}^d \to \mathbb{R}$ is said to be *positive definite on* $\mathbb{R}^d$ if the inequality $\sum_{i,j=1}^{N} \lambda_i \lambda_j \Phi(x_i - x_j) \geq 0$ holds for any $x_i \in \mathbb{R}^d$ and real numbers λ_i [41].

Let $\Pi_m(\mathbb{R}^d)$ denote the linear space of all polynomials of total degree less than or equal to m. A function $\Phi : \mathbb{R}^d \to \mathbb{R}$ is *conditionally positive (or negative) definite of order m on* $\mathbb{R}^d$, if for all sets $X = \{x_1, x_2, \ldots, x_N\} \in \mathbb{R}^d$ with N distinct points and all vectors $\lambda = (\lambda_1 \ \lambda_2 \ \ldots \ \lambda_N)^T \in \mathbb{R}^N$ subject to the conditions

$$\sum_{i=1}^{N} \lambda_i p(x_i) = 0, \quad p \in \Pi_{m-1}(\mathbb{R}^d),$$

the quadratic form

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j \Phi(x_i - x_j),$$

is non-negative (or non-positive) and vanishes only when $\lambda = 0$. Micchelli [30] related the conditional positive definiteness of the radially symmetric functions $\Phi(x) = \varphi(\|x\|_2)$ to complete monotonicity of derivatives of φ. In particular, Φ is conditionally positive (negative) definite of order m for any d if the derivative of order m of $\varphi(r^{1/2})$, $r \geq 0$, is completely monotonic.

In addition to proving the feasibility of interpolation, the relation to conditionally positive (or negative) functions implies a means to represent a radial basis function as an integral of a Gaussian by a suitable measure. For example, Schoenberg [40] proved that Bochner's Theorem [41] implies that a function φ that is radially symmetric and positive definite on all $\mathbb{R}^d$ must satisfy

$$\varphi(r) = \int_0^\infty e^{-sr^2} \mathrm{d}\mu(s), \tag{14}$$

with $\mu : [0, \infty) \to \mathbb{R}$ a finite positive Borel measure. Furthermore, due to a theorem by Schoenberg [40], φ is a conditionally negative definite function of order one on every $\mathbb{R}^d$ if and only if there exists a Borel measure $\mu : [0, \infty) \to \mathbb{R}$ such that

$$\varphi(r) = \varphi(0) + \int_0^\infty \frac{1 - e^{-sr^2}}{s} \mathrm{d}\mu(s), \tag{15}$$

and μ has the properties

$$\int_0^1 \mathrm{d}\mu(t) < \infty \text{ and } \int_1^\infty t^{-1} \mathrm{d}\mu(t) < \infty.$$

The above conditions being required for the integral to be finite. Finally, at least for some higher order conditionally negative definite functions it is possible to construct expressions similar to (14) and (15). For example, in [29] we have shown that the Thin-plate Spline may be written as

$$\varphi(r) = \frac{r-1}{2} + \frac{1}{2} \int_0^\infty e^{-s} \, \frac{e^{-s(r-1)} + s(r-1) - 1}{s^2} \mathrm{d}s. \tag{16}$$

We can use Micchelli's results to make useful observations about the interpolation matrix of several Radial Basis Functions. For example, the Gauss kernel and the Inverse Multiquadric are (radially symmetric) positive definite function on $\mathbb{R}^d$ then the corresponding interpolation matrices are positive definite. The Euclidean distance and the Hardy Multiquadric are conditionally negative definite functions of order one on $\mathbb{R}^d$ and thus the corresponding interpolation matrices have one positive and $N-1$ negative eigenvalues, provided that the centers are distinct and there are at least two of them. Finally, the Thin-plate Spline is a conditionally positive definite function of order two on $\mathbb{R}^d$. The particulars of the associated representations have been detailed elsewhere [29] and here we will only refer to the relations we need to develop our rapid summation schemes:

Theorem 1 *For centers $\{x_i\}_{i=1}^N$ and y in $\mathbb{R}^d$ we have*

$$\sum_{i=1}^{N} \frac{\lambda_i}{\sqrt{||y-x_i||^2+c^2}} = \frac{1}{\sqrt{\pi}} \int_0^\infty \frac{e^{-sc^2}}{\sqrt{s}} \sum_{i=1}^{N} \lambda_i e^{-s\|y-x_i\|^2} \mathrm{d}s.$$

Theorem 2 *For centers $\{x_i\}_{i=1}^N$ and y in $\mathbb{R}^d$ we have*

$$\begin{aligned} \sum_{i=1}^{N} \lambda_i \sqrt{\|y-x_i\|^2+c^2} &= c\sum_{i=1}^{N} \lambda_i + \\ &+ \frac{1}{2\sqrt{\pi}} \int_0^\infty \frac{e^{-sc^2}}{\sqrt{s}} \left(\frac{\sum_{i=1}^N \lambda_i - \sum_{i=1}^N \lambda_i e^{-s\|y-x_i\|^2}}{s} \right) \mathrm{d}s. \end{aligned}$$

Theorem 3 *For centers $\{x_i\}_{i=1}^N$ and the corresponding coefficients $\{\lambda_i\}_{i=1}^N$ such that $\sum_{i=1}^N \lambda_i = \sum_{i=1}^N \lambda_i x_i = 0$ and y in $\mathbb{R}^d$ we have*

$$\begin{aligned} \sum_{i=1}^{N} \lambda_i \|y-x_i\|^2 \log\|y-x_i\| &= \frac{1}{2}\sum_{i=1}^{N} \lambda_i \|x_i\|^2 + \\ &\frac{1}{2} \int_0^\infty \frac{e^{-s}}{s^2} \left(\sum_{i=1}^{N} \lambda_i e^{-s(\|y-x_i\|^2-1)} + s\sum_{i=1}^{N} \lambda_i \|x_i\|^2 \right) \mathrm{d}s. \end{aligned}$$

3 Rapid Evaluation Algorithm Description

In this section we discuss the basic steps of the rapid evaluation algorithm. Without loss of generality, we will discuss the algorithm for the Inverse Multiquadric method and we will return to the details required for other radial basis functions in the following paragraphs. We assume that we have already calculated the solution to the interpolation problem, that is we have computed the coefficients λ_i so that

$$s(y) = \sum_{i=1}^{N} \frac{\lambda_i}{\sqrt{\|y-x_i\|^2+c^2}}, \tag{17}$$

satisfies the N interpolation conditions for a set of centers $x_1, x_2, \ldots, x_N$ in $\mathbb{R}^d$ for data values $f_1, f_2, \ldots, f_N$ and we consider the task of evaluating (17) at a set of M points $y_1, y_2, \ldots, y_M$. Performing this calculation directly requires $\mathcal{O}(MN)$ operations. Our aim is to reduce the complexity

of the evaluation task by developing a rapid summation scheme. Without loss of generality we will assume that $c^2 = 1$ and that all the centers and the evaluation points are contained in the unit hypercube $B_0 = [0,1]^d$.

Applying Theorem 1 to 17 we have

$$s(y) = \frac{1}{\sqrt{\pi}} \int_0^\infty \frac{e^{-t}}{\sqrt{t}} \sum_{i=1}^N \lambda_i e^{-t\|y-x_i\|^2} \, \mathrm{d}t. \tag{18}$$

Using (18) we can approximate s using a q-term generalised Gauss-Laguerre quadrature rule

$$s(y) = \frac{1}{\sqrt{\pi}} \sum_{k=1}^q w_k f(t_k), \tag{19}$$

where

$$f(t) = \sum_{i=1}^N \lambda_i e^{-t\|y-x_j\|^2}. \tag{20}$$

Thus, rather than evaluating the sum of Inverse Multiquadrics (17) at an overall cost of $\mathcal{O}(MN)$ operations, we may evaluate q sums of Gaussians (20) (one for each quadrature node t_k) via the Fast Gauss Transform in $\mathcal{O}\big(q(N+M)\big)$ operations. Recall that the decrease in the computational complexity of the latter task is due to the decoupling of the precomputation of the moments of the points x_i and the estimation of the interpolant at points y_j through the already computed moments.

Of course, the quadrature introduces the error

$$|\varepsilon_q| = \frac{q!}{(2q)!} \, \Gamma(q+\frac{1}{2}) f^{(2q)}(\xi), \quad 0 < \xi < \infty.$$

The quadrature nodes t_k are the zeros of the generalised Laguerre polynomial $L_q^{(-1/2)}(t)$ and the weights may be computed by the formula

$$w_k = \frac{q!\Gamma(q+1/2)t_k}{\big(L_{q+1}^{-1/2}(t_k)\big)^2}.$$

Overall, the fast evaluation of Inverse Multiquadric interpolants may be performed by Algorithm 2. In order to identify exactly the structure of the iterations, we summarise the Hermite expansion required by a two-dimensional FGT here. Indeed, for a point $C \equiv (c_1, c_2)$ the Gauss kernel can be approximated by

$$e^{-t\|x-y\|^2} = \sum_{n_1,n_2=0}^p A_{n_1,n_2} h_{n_1}(\sqrt{t}(x_1-c_1)) h_{n_2}(\sqrt{t}(x_2-c_2)), \tag{21}$$

with the moments A_{n_1,n_2}

$$A_{n_1,n_2} = \frac{1}{n_1!n_2!} \sum_{j=1}^N \lambda_j \left(\sqrt{t}(x_{j_1}-z_1)\right)^{n_1} \left(\sqrt{t}(x_{j_2}-z_2)\right)^{n_2}. \tag{22}$$

Hence, in two dimensions the precomputation of the moments (Steps 9-11 of Algorithm 2) requires $\mathcal{O}(qp^2)$ operations per centre. The evaluation of the moments at a single point (Steps 17-19 of Algorithm 2) requires $\mathcal{O}(qp^2(2n+1)^2)$ operations. Overall, the two-dimensional fast evaluation algorithm requires $\mathcal{O}(qp^2N + qp^2(2n+1)^2M)$ operations.

choose q and p to guarantee the required precision
compute the weights w_k and nodes t_k of the Gauss-Laguerre quadrature rule (19)
for each quadrature node t_k **do**
  subdivide B_0 into boxes of side at most $\sqrt{2/t_k}$
end for
{start first stage: precompute moments}
for each centre x_j **do**
  for each quadrature node t_k **do**
    find the box C which contains x_j
    for $\beta < p$ **do**
      compute the contribution of x_j to the moments A_β of box C using (11) on page 6
    end for
  end for
end for
{start second stage: evaluate moments}
for each evaluation point y_i **do**
  for each quadrature node t_k **do**
    find the box B that contains y_j
    for each of the $(2n+1)^d$ nearest neighbours of B **do**
      accumulate the series (12) on page 6 truncated after p^d terms to obtain an approximation to the Gaussian with parameter t_k
    end for
    accumulate the contribution of the k-th point of the quadrature rule (19)
  end for
end for

Algorithm 2: Fast Summation of Inverse Multiquadrics. Accepts as input the N points x_i, the associated weights λ_i and M evaluation points y_j, and returns the value of $s(y) = \sum_{i=1}^{N} \lambda_i \left(\|y - x_i\|^2 + 1\right)^{-1/2}$ at the points y_j.

In the d-dimensional setting, precomputation of the moments requires $\mathcal{O}(qp^d)$ operations per centre and the evaluation at one point $\mathcal{O}(qp^d(2n+1)^d)$ operations. Overall, the d-dimensional fast evaluation algorithm requires $\mathcal{O}(qp^dN + qp^d(2n+1)^dM)$ operations.

In section 2 we established that integral representations similar to (18) may be constructed for conditionally negative (or positive) definite functions. In particular, Theorems 2 and 3 show such constructs for the Hardy Multiquadric and the Thin-plate Spline correspondingly. In the following section, we discuss suitable quadrature rules for these cases.

3.1 Numerical Integration

Perhaps the most popular interpolatory quadrature formula is the formula of Gauss type [12]

$$\int_0^\infty w(t)f(t)\mathrm{d}t \approx \sum_{k=1}^{q} w_k f(t_k), \tag{23}$$

where the t_k and the w_k have been determined so that the formula is exact for all $f \in \Pi_{2q-1}$. For the weight function $w(t)$ equal to:

$$w(t) = t^a e^{-t}, \qquad a > -1, \tag{24}$$

we obtain the generalised Laguerre formula [12]:

$$\int_0^\infty t^a e^{-t} f(t)\mathrm{d}t = \sum_{k=1}^{q} w_k f(t_k) + \frac{q!\Gamma(q+a+1)}{(2q)!} f^{(2q)}(\xi), \tag{25}$$

for some $\xi \in (0,\infty)$. The nodes t_k are the zeros of the Laguerre polynomials $L_q^{(a)}(t)$. The weights are given by the formula

$$w_k = \frac{q!\Gamma(q+a+1)t_k}{\left(L_{q+1}^a(t_k)\right)^2}$$

Let us now consider the use of Gauss-Laguerre formulae for the evaluation of the integral of Theorem 2. For simplicity let $c^2 = 1$, then the general case can be treated similarly by the change of variable $u = sc^2$. We have

$$\sum_{i=1}^{N} \lambda_i \sqrt{\|y - x_i\|^2 + 1} = c\sum_{i=1}^{N} \lambda_i + \frac{1}{2\sqrt{\pi}} \int_0^\infty \frac{e^{-s}}{\sqrt{s}} f(s)\mathrm{d}s \tag{26}$$

with

$$f(s) = \frac{\sum_{i=1}^{N} \lambda_i - \sum_{i=1}^{N} \lambda_i e^{-s\|y-x_i\|^2}}{s}.$$

Hence, setting $a = -\frac{1}{2}$ we can use (25) to estimate (26) in $\mathcal{O}(q)$ function evaluations. The quadrature introduces the error

$$|\varepsilon_q| \leq \frac{q!\Gamma(q+1/2)}{(2q)!} f^{(2q)}(\xi), \quad 0 < \xi < \infty.$$

The Gaussian quadrature nodes and the corresponding weights may be computed using one of a number of standard methods, for example using Gautschi's ORTHOPOL package [18].

Alternatively, we may consider the weight functions

$$w(t) = \frac{e^{-tc^2}}{\sqrt{t}}, \quad c \in \mathbb{R}, \tag{27}$$

and construct a Gaussian quadrature rule with respect to (27) rather than the classical Gauss-Laguerre weight function (24). However, in this case we no longer have an explicit formula for the quadrature weights and we need to employ an alternative method for their estimation, for example the so-called method of *orthogonal reduction* [17]. In this case, we can use routine D1MACH from Gautschi's ORTHOPOL package [18] and also note

that the formula below provides appropriate approximations to the inner product

$$\int_0^\infty \frac{e^{-tc^2}}{\sqrt{t}} u(t)v(t)dt = \frac{1}{c}\int_0^\infty \frac{e^{-s}}{\sqrt{s}} u(\frac{s}{c^2})v(\frac{s}{c^2})ds \approx \frac{1}{c}\sum_{k=1}^{m} \widetilde{w}_k u(\frac{s_k}{c^2})v(\frac{s_k}{c^2}),$$

as needed by the ORTHOPOL package.

Finally, let us consider the evaluation of Thin-plate Spline interpolants. Theorem 3 implies that

$$\sum_{i=1}^{N} \lambda_i \|y - x_i\|^2 \log \|y - x_i\| = \frac{1}{2}\sum_{i=1}^{N} \lambda_i \|x_i\|^2 +$$

$$+\frac{1}{2}\int_0^\infty \frac{e^{-t}}{t^2}\left(\sum_{i=1}^{N} \lambda_i e^{-t(\|y-x_i\|^2-1)} + t\sum_{i=1}^{N} \lambda_i \|x_i\|^2\right). \qquad (28)$$

Numerical evidence implies that using Gauss-Laguerre quadrature ($a = 0$) for the function

$$f(t) = \frac{\sum_{i=1}^{N} \lambda_i e^{-t(\|y-x_i\|^2-1)} + t\sum_{i=1}^{N} \lambda_i \|x_i\|^2}{t^2},$$

does not guarantee the required precision. Thus, an alternative approach is required. Indeed, first we make the change of variable $t \to \frac{1-r}{r}$ so that the semi-infinite integration interval is mapped on $[0, 1]$, that is

$$\int_0^\infty f(t)\mathrm{d}t = \int_0^1 f(\frac{1-r}{r})r^{-2}\mathrm{d}r, \qquad (29)$$

where f is the function

$$f(t) = \frac{\sum_{i=1}^{N} \lambda_i e^{-t(\|y-x_i\|^2-1)} + t\sum_{i=1}^{N} \lambda_i \|x_i\|^2}{t^2}.$$

In this case we estimate the value of interpolant s using Kronrod quadrature [12, p. 77] which overcomes one of the main shortcomings of Gaussian quadrature. Indeed,when proceeding from the n-point to the m-point Gaussian rule with $n < m$, all functional values are discarded (with the possible exception of the midpoint for n odd), since the nodes of the n-point rule do not include any of the nodes of the m-point rule. The Kronrod scheme overcomes this problem to some extent, by augmenting the n-point rule with $n + 1$ nodes selected so that the resulting quadrature is exact for polynomials in Π_{3n+1}.

We use the Kronrod scheme to estimate the Thin-plate Spline integral (28) using (29) to map the infinite interval $[0, \infty)$ to the finite $[0, 1]$. Then, we use the Kronrod scheme on a partition of $[0, 1]$ that guarantees the required accuracy. To estimate the accuracy provided by a particular quadrature rule on a certain subinterval we use the difference between the Gauss and the corresponding Kronrod scheme estimates on that interval. The Kronrod quadrature interval $[-1, 1]$ may be mapped to any interval $[a, b]$ by applying the following transformation

$$\int_a^b f(t)\mathrm{d}t = \int_{-1}^1 \frac{b-a}{2} f(\frac{b+a+(b-a)x}{2})\mathrm{d}x.$$

4 Numerical Results

In this section we discuss the performance of the fast evaluation method on a representative selection of test problems. In all test problems we report the maximum observed relative error ϵ defined as

$$\epsilon = \max_{1 \le j \le M} \frac{||f_j - s_j||_2}{||f_j||_2},$$

where f_j is the exact and s_j the computed interpolant value at evaluation point y_j. In all test problems, the interpolant coefficients λ_j have been randomly selected in the interval $[-1, 1]$.

Test Problem 1. For the first test case we consider one-dimensional Hardy Multiquadric interpolation. The N centers and the M evaluation points are chosen from the uniform distribution on the unit interval. The method scales linearly with respect to N and M (Table 1). Although of little practical significance, this test problem is included here to show the scaling of the method with respect to the number of dimensions.

Test Problem 2. The second test case is two-dimensional Inverse Multiquadric interpolation (Table 2). The evaluation points coincide with the centers and are uniformly distributed on the unit square. The observed accuracy of the computation is $\mathcal{O}(10^{-14})$. The results for this case are also shown in graphical format in Figure 1.

Test Problem 3. This is a case of two-dimensional Hardy Multiquadric interpolation with track data. The centers are positioned within one tenth of the diagonal of the unit square and within it. The observed relative error of the calculation is $\mathcal{O}(10^{-7})$ (Table 3).

Test Problem 4. This is case of two-dimensional Hardy Multiquadric interpolation we investigate the case $N << M$ which often occurs in practical applications. The N centers are positioned within 1/10 unit along the diagonal of the unit square and the M evaluation points are are uniformly distributed in the unit square. The observed relative error of the calculation is $\mathcal{O}(10^{-7})$ (Table 4). The reported timings show the approximate and direct evaluation of the Hardy Multiquadric interpolants. This test case highlights the fact that the method does not appear to be sensitive to the relative values of M and N. Indeed, the observed computational complexity of the method is linear with either M or N.

$N = M$	approximate(sec)	direct(sec)
300	0.013	0.016
500	0.015	0.028
1,000	0.027	0.108

Table 1: Performance of the fast evaluation method ($d = 1$). The N centers and the M evaluation points are uniformly distributed in the unit interval. The table shows times in seconds for the approximate and direct evaluation of the Hardy Multiquadric interpolants at observed relative accuracy of $\mathcal{O}(10^{-14})$.

$N = M$	fast(sec)	direct(sec)
1,000	1.15	1.5
2,000	1.37	6.12
4,000	1.81	24.41
16,000	4.26	364.42
20,000	5.44	576.74

Table 2: Performance of the fast evaluation method ($d = 2$). The N centers coincide with the M evaluation points and are uniformly distributed in the unit square. The table shows times in seconds for the approximate and direct evaluation of the Inverse Multiquadric interpolants.

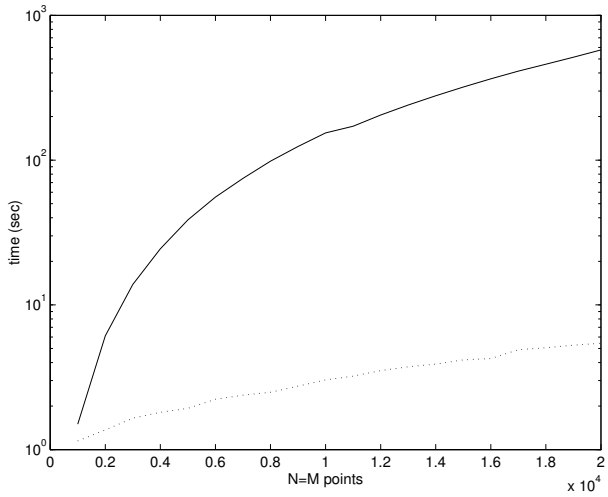

Figure 1: Graphic representation of Table 2, comparing the performance of the direct and the fast method. Dotted line is the fast method. The y axis is logarithmic.

$N = M$	approximate(sec)	direct(sec)
100	0.04	0.01
200	0.05	0.05
500	0.10	0.25
1,000	0.16	1.03
5,000	0.60	25.69
10,000	1.22	103.53
100,000	11.78	2,080.00
500,000	59.54	10,300.00

Table 3: Performance of the fast evaluation method ($d = 2$). The N centers are positioned within 1/10 unit along the diagonal of the unit square and the $M = N$ evaluation points are are uniformly distributed in the unit square. This test case highlights the fact that the method does not appear to be sensitive to the particular distribution of centres and no observable differences are noted to the uniformly distributed case. The table shows times in seconds for the approximate and direct evaluation of the Hardy Multiquadric interpolants.

M	approximate(sec)	direct(sec)
100	0.03	0.01
200	0.05	0.05
500	0.07	0.25
1,000	0.11	1.03
5,000	0.33	25.69
10,000	0.64	103.53
100,000	6.02	2,080.00
500,000	28.54	10,300.00

Table 4: Performance of the fast evaluation method for $N << M$ ($d = 2$). The N centers are positioned within 1/10 unit along the diagonal of the unit square and the $M = N$ evaluation points are are uniformly distributed in the unit square. In this case N is set to 200. The table shows times in seconds for the approximate and direct evaluation of the Hardy Multiquadric interpolants.

$N = M$	approximate(sec)	direct(sec)
700	0.92	0.83
800	1.00	1.04
900	1.01	1.33
1,000	1.07	1.62
5,000	2.73	41.24
10,000	4.87	166.96
100,000	43.31	3,340.00
300,000	130.52	10,020.00

Table 5: Performance of the fast evaluation method ($d = 3$). The N centers and the evaluation points are are uniformly distributed in the unit cube. The table shows times in seconds for the approximate ($\mathcal{O}(10^{-5})$ relative accuracy) and direct evaluation of the Hardy Multiquadric .

$N = M$	approximate(sec)	direct(sec)
1,000	6.42	1.62
5,000	16.38	41.24
10,000	29.22	166.96
100,000	259.86	3,340.00
300,000	783.12	10,020.00

Table 6: Performance of the fast evaluation method ($d = 3$). The N centers and the evaluation points are are uniformly distributed in the unit cube. The table shows times in seconds for the approximate and direct evaluation of the Inverse Multiquadric.

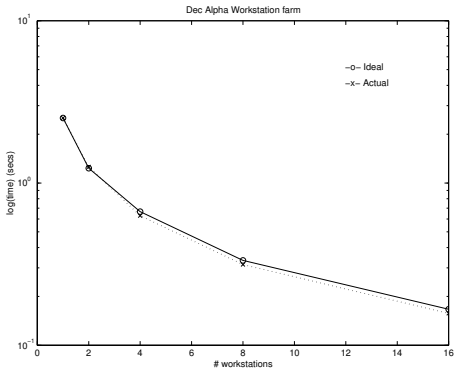

Figure 2: Performance on the DEC Alpha cluster.

Test Problem 5. This is a case of three dimensional Hardy Multiquadric interpolation. The centers and the evaluation points are distributed uniformly in the unit cube. The observed relative accuracy of the method is $\mathcal{O}(10^{-5})$ (Table 5). This test case indicates that the method scales well with d in accordance with the predicted complexity of the algorithm.

Test Problem 6. This is a case of three dimensional Inverse Multiquadric interpolation. The centers and evaluation points are uniformly distributed in the unit cube. The observed relative accuracy of the method is $\mathcal{O}(10^{-7})$. This computation may also be seen as the force calculation step of a N-body problem with the Plummer potential (Table 6). We expect thus that the algorithm developed in this paper may be used effectively in other contexts in addition to the evaluation of Radial Basis Functions.

Test Problem 7. This is a parallelized version of the rapid summation algorithm. The test environment is a high performance cluster of workstations assembled from commodity components using sixteen high end Digital Unix workstations connected over fast Ethernet and organized in a star topology and using an MPI-2 environment. Synchronisation was implemented with allreduce operations in a SPMD model. Even on relatively small problems ($N = M = 32,000$) the method scales very well. In this case, the I/O is local at the filesystem of each workstation and data are distributed and collected using standard operating system services rather than a distributed filesystem. The actual performance of the method is shown in Figure 2. Note that for a problem of similar size the best scalable implementation of a treecode achieves approximately 65% efficiency. On the other hand, the fast evaluation method examined here achieves in excess of 94% efficiency.

5 Conclusions

In this paper we have constructed a rapid evaluation method for radial basis function interpolants. The algorithm is based on a fundamental property of conditionally negative (or positive) definite functions. It employs the FGT to compute Gauss kernel sums and a suitable quadrature

rule. The method has been shown to be especially well suited for high performance computing, in particular computation on clusters of workstations.

References

[1] M. Abramowitz and I. Stegun (1970) *Handbook of Mathematical Functions*, Dover.

[2] A.W. Appel (1985) "An Efficient Program for Many-body Simulation", *SIAM J. Sci. Stat. Comp.*, Vol. 6, No. 1, pp. 85-103.

[3] N. Arad (1994) "Image Warping by Radial Basis Functions: Applications to Facial Expressions", *Computer Vision, Graphics and Image Processing*, Vol. 56, No. 2, pp.161-172.

[4] J. Barnes and P. Hut (1986) "A Hierarchical $\mathcal{O}(N \log N)$ Force Calculation Algorithm", *Nature*, 324, pp. 446-449.

[5] B.J.C. Baxter (1992) *The Interpolation Theory of Radial Basis Functions*, Ph.D. Thesis, Trinity College, University of Cambridge.

[6] B.J.C. Baxter and George Roussos (1997) "Fast Evaluation of Conditionally Negative Definite Functions", *IMACS Conference on Radial Basis Functions*, May 27-29, Pacific Grove, CA.

[7] R.K. Beatson and W.A. Light (1994) "Fast Evaluation of Radial Basis Functions: Methods for the 2-dimensional polyharmonic Splines", Department of Mathematics and Statistics, University of Canterbury, New Zealand, *Technical Report* No. 119.

[8] R.K. Beatson and G.N. Newsam (1992) "Fast Evaluation of Radial Basis Functions: Part I", *Comp. Math. Applic.*, Vol. 24, No. 12, pp. 7-19.

[9] R.K. Beatson and M.J.D. Powell (1993) "An Iterative Method for Thin-plate Spline Interpolation that Employs Approximations to Lagrange Functions" in D. Griffiths (ed.) *Proceedings of the 14th Biennial Numerical Analysis Conference*, J. Wiley, pp. 17-39.

[10] R.K. Beatson, G. Goodsell and M.J.D. Powell (1995) "On Multigrid Techniques for Thin-plate Spline Interpolation in Two Dimensions", *Lectures in Applied Mathematics*, Vol. 32, pp. 77-97.

[11] J.B. Cherrie, R.K. Beatson and G.N. Newsam (2002) "Fast Evaluation of Radial Basis Functions: Methods for Generalized Multiquadrics in $\mathcal{R}^n$", *SIAM J. Sci. Comput.*, Vol. 23, pp. 1549–1571.

[12] P.J. Davies and P. Rabinowitz (1975) *Methods of Numerical Integration*, Academic Press.

[13] N. Dyn and D. Levin (1983) "Iterative Solution of Systems Originating from Integral Equations and Surface Interpolation", *SIAM J. Numer. Anal.*, Vol. 20, No. 2, pp. 377-390.

[14] N. Dyn, D. Levin and S. Rippa (1986) "Numerical Procedures for Surface Fitting of Scattered Data by Radial Functions", *SIAM J. Sci. Stat. Comp.*, Vol. 7, No. 2, pp. 639-659.

[15] A.C. Faul and M.J.D. Powell (1998) "Proof of Convergence of an Iterative Technique for Thin Plate Spline Interpolation in Two Dimensions", Department of Applied Mathematics and Theoretical Physics, University of Cambridge, *Technical Report No.* damtp 1998/na8.

[16] R. Franke (1982) "Scattered Data Interpolation: Test of Some Methods", *Math. Comp.*, Vol. 38, pp. 181-200.

[17] M. Gander and A. Karp (2001) "A Gaussian quadrature rule for an integral from radiation transfer calculations", *J. of Quantitative Spectroscopy and Radiative Transfer*, Vol. 168, pp. 213–223.

[18] W. Gautschi (1994) "Algorithm 726: ORTHOPOL – A Package of Routines for Generating Orthogonal Polynomials and Gauss-type Quadrature Rules", *ACM Trans. Math. Soft.*, Vol. 20, No. 1, pp. 21-62.

[19] G.H. Golub and J.H. Welsch (1969) "Calculation of Gauss Quadrature Rules", *Math. Comput.*, Vol. 23, No. 106, pp. 221-230.

[20] G. Goodsell (1997) "Computing the p-nearest Neighbours for Small p", Department of Applied Mathematics and Theoretical Physics, University of Cambridge, *Technical Report No.* damtp 97/na01.

[21] G. Goodsell and M.J.D. Powell (1995) *A Multigrid-type Method for Thin-plate Spline Interpolation on a Circle*, *The* 16*th Biennial Conference on Numerical Analysis*, Dundee.

[22] W.B. Gragg and W.J. Harrod (1984) "The Numerically Stable Reconstruction of Jacobi Matrices from Spectral Data", *Numer. Math*, Vol. 44, No. 3, pp. 317-335.

[23] L. Greengard (1987) *The Rapid Evaluation of Potential Fields in Particle Systems*, The MIT Press.

[24] L. Greengard (1991) "Fast Algorithms for Classical Physics", *Science*, Vol. 255, pp. 909-914.

[25] L. Greengard and J. Strain (1991) "The Fast Gauss Transform", *SIAM J. Sci. Stat. Comp.*, Vol. 12, No. 1, pp. 79-94.

[26] L. Greengard and X. Sun (1998) "A New Version of the Fast Gauss Transform", *Documenta Mathematica*, Extra Volume ICM 1998 III, pp. 575-584.

[27] R.L. Hardy (1990) "Theory and Applications of the Multiquaric-Biharmonic Method", *Computers Math. Appl.*, Vol. 19, No. 8/9, pp. 163-208.

[28] R.L. Hardy (1997) "The Mathematical Physics of a Biharmonic Approach to Disturbing Potential based on Multiquadric Summation", *IMACS Conference on Radial Basis Functions*, May 27-29, Pacific Grove, CA.

[29] S. Hubbert, B.J.C. Baxter and George Roussos (2003) "Some Properties of Radial Basis Functions", *Num. Math.*, to appear.

[30] C.A. Micchelli (1986) "Interpolation of Scattered Data: Distance Matrices and Conditionally Positive Functions", *Constr. Approx.*, Vol. 2, pp. 11-22.

[31] C.A. Micchelli (1986) "Algebraic aspects of interpolation" in C. de Boor (ed.) *Approximation Theory*, Proceedings of Symposia in Applied Mathematics, Vol. 36, American Mathematical Society, Providence, Rhode Island, pp. 81-102.

[32] R. Piessens, E. de Doncker-Kapenga, C.W. Überhuber and D.K. Kahaner (1983) *QUADPACK: A Subroutine Package for Automatic Integration*, Springer Verlag, Berlin.

[33] M.J.D. Powell (1992) "Tabulation of Thin-plate Splines on a Very Fine Two Dimensional Grid", Department of Applied Mathematics and Theoretical Physics, University of Cambridge, *Technical Report No.* DAMTP 1992/NA2.

[34] M.J.D. Powell (1993) "Truncated Laurent Expansions for the Fast Evaluation of Thin-plate Splines", *Num. Alg.*, Vol. 5, No. 2, pp. 99-120.

[35] M.J.D. Powell (1994) "Some Algorithms for Thin-plate Spline Interpolation to Functions of Two Variables" in H.P. Dikshit and C.A. Micchelli (eds.) *Advances in Computational Mathematics, New Dehli, India*, World Scientific, Singapore, pp. 303-319.

[36] M.J.D. Powell (1997) "A New Iterative Algorithm for Thin-late Spline Interpolation in Two Dimensions", *Ann. Num. Math.*, Vol. 4, pp. 519-527.

[37] G. Roussos and B.J.C. Baxter (2001) "A Scalable Method for Many Body Computations", in Y. Cotronis, J. Dongarra (eds.) *Recent Advances in Parallel Virtual Machine and Message Passing Interface, 8th European PVM/MPI Users' Group Meeting, Santorini/Thera, Greece, September 23-26*, Lecture Notes in Computer Science, Vol. 2131, pp. 480- 488.

[38] G. Roussos and B.J.C. Baxter (2002) "A new Error Estimate for the Fast Gauss Transform", *SIAM Journal of Scientific Computing*, Vol. 24, Num. 1, pp. 257-259.

[39] R. Sibson and G. Stone (1991) "Computation of Thin-plate Splines", *SIAM J. Sci. Stat. Comp.*, Vol. 12, No. 6, 1304-1313.

[40] I.J. Schoenberg (1938) "Metric Space and Completely Monotone Functions", *Ann. of Math.*, Vol. 39, pp. 811-841.

[41] James Stewart (1976) "Positive Definite Functions and Generalizations, A Historical Survey", *Rocky Mount. J. Math.*, Vol. 6, No. 3, pp. 409-433.

[42] J. Strain (1991) "The Fast Gauss Transform with Variable Scales", *SIAM J. Sci. Stat. Comp.*, Vol. 32, No. 5, pp. 1131-1137.

[43] Gabor Szegö (1985) *Orthogonal Polynomials*, AMS Colloquium Publications.

[44] D. Suter (1993) "Multipole Methods for Visual Reconstruction", *SPIE Geometric Methods in Computer Vision II*, Vol. 2031, pp. 16-26.

[45] G.N. Watson (1959) "A Note on Gamma Functions", *Proc. Edinburgh Math. Soc.*, Vol. (2) 11 (1958/59), Edinburgh Math. Notes 42, pp. 7-9.

[46] E.T. Whittaker and G.N. Watson (1927) *A Course of Modern Analysis*, Cambridge University Press, Cambridge.