TITLE: Fast Evaluation of Radial Basis Functions: A Multivariate
Momentary Evaluation Scheme

DISTRIBUTION: Approved for public release, distribution unlimited

This paper is part of the following report:

TITLE: International Conference on Curves and Surfaces [4th], Saint-Malo,
France, 1-7 July 1999. Proceedings, Volume 2. Curve and Surface Fitting

To order the complete compilation report, use: ADA399401

The following component part numbers comprise the compilation report:
ADP011967 thru ADP012009

Fast Evaluation of Radial Basis Functions:
A Multivariate Momentary Evaluation Scheme

R. K. Beatson and E. Chacko

Abstract. This paper presents a scheme for fast evaluation of splines, or "radial" basis functions, of the form $s(\cdot) = p(\cdot) + \sum_{i=1}^{N} \lambda_i \Phi(\cdot - z_i)$. Here p is a low degree polynomial and $\Phi : \mathbb{R}^n \to \mathbb{R}$ is a function that need not be radial. This multivariate momentary evaluation scheme is a generalization of the fast multipole method in which calculations with far field expansions are replaced by calculations involving moments of the data. The primary advantage of this new algorithm is that it is highly adaptive to changes in Φ.

§1. Introduction

This paper presents a scheme for fast evaluation of splines, or "radial" basis functions, of the form

$$s(\cdot) = p(\cdot) + \sum_{i=1}^{N} \lambda_i \Phi(\cdot - z_i). \qquad (1)$$

Here p is a low degree polynomial, and $\Phi : \mathbb{R}^n \to \mathbb{R}$ is a function that need not be radial. This multivariate momentary evaluation scheme is a generalization of the fast multipole method in which calculations with far field expansions are replaced by calculations involving moments of the data. The primary advantage of this new algorithm is that it is highly adaptive to changes in Φ. More precisely, changing to a new basic function Φ only requires coding a one or two line function for the (slow) evaluation of Φ. In contrast, adapting a conventional fast multipole code to a different Φ requires considerable analysis of appropriate expansions and transformation theorems, followed by writing a very specific code. The current algorithm reduces the incremental cost of a single extra evaluation from $\mathcal{O}(N)$ to $\mathcal{O}(1)$ operations, and the cost of a matrix-vector product (that is, evaluation at all centres) from $\mathcal{O}(N^2)$ to

$\mathcal{O}(N \log N)$ operations. The algorithm can be viewed as a spline fitter in
that the approximation it produces is a piecewise polynomial supplemented
by appropriate direct evaluations. The method described is a multivariate
generalisation of the method of [4].

In outline the setup phase of method is as follows. Firstly, space is divided
in a hierarchical manner. For example, in a 2-D setting an initial square
could be divided into a quadtree. Then centres are associated with the panels
they lie in. Next, proceeding up the tree level by level, the moments of the
coefficients (the λ_i's about panel centres are calculated. Next working down
the tree for each level, a number of approximations to Φ are formed. Then for
each panel within a level, polynomial approximations to that part of s due to
far away centres are formed by combining moments and the approximations
to Φ. The evaluation phase first identifies the childless panel to which the
evaluation point belongs. Then it approximates the far field by evaluating
the polynomial associated with that panel, and adds to that approximation
the directly calculated near field part of $s(x)$. For reasons of space, we will
not detail suitable methods for subdividing space, or the process of evaluation.
These matters are well understood in the context of the fast multipole method,
see for example [2].

The paper is organized as follows. The necessary mathematics for form-
ing polynomial approximations to s from moments and approximations to Φ,
and for translating moments is given in Secton 3 below. Section 4 contains
symmetry results that can substantially reduce the amount of work required to
form approximations to Φ at each level. Section 5 contains numerical results
obtained with a preliminary implementation of the method.

§2. Notation

We will need the following notation. A multi-index $\alpha = (\alpha_1, \dots, \alpha_n)$ is an
n-tuple of nonegative integers. If x is an element of $\mathbb{R}^n$, we will write its
components as $x_1, x_2, \dots, x_n$. We will also need sequences of points in $\mathbb{R}^n$.
In an effort to make the meaning of all subscripted symbols transparent, we
will write all such sequences of vectors as $\{z_m\}$, and z will never be used
unsubscripted to denote a single point in $\mathbb{R}^n$.

If a and b are elements of $\mathbb{R}^n$, then we will say a is less than or equal to b,
and write $a \leq b$, if $a_i \leq b_i$ for all $1 \leq i \leq n$. We also define for $x \in \mathbb{R}^n$ and α, β
multi-indices $|\alpha| = \alpha_1 + \alpha_2 + \cdots + \alpha_n$, $\alpha! = \alpha_1! \alpha_2! \cdots \alpha_n!$, $x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$,
and for $0 \leq \beta \leq \alpha$ take $\binom{\alpha}{\beta} = \frac{\alpha!}{(\alpha-\beta)!\beta!}$. The (multivariate) Binomial
Theorem then assumes the form

$$(x + t)^\alpha = \sum_{0 \leq \beta \leq \alpha} \binom{\alpha}{\beta} x^{\alpha-\beta} t^\beta,$$

for all multi-indices $0 \leq \alpha \in \mathcal{Z}^n$ and points $x, t \in \mathbb{R}^n$. We further define the
normalized monomial

$$V_\alpha(x) = x^\alpha / \alpha!, \tag{2}$$

and the α-th normalized moment about t of the data $\{(z_m, \kappa_m)\}_{m=1}^{M}$

$$\sigma_{t,\alpha} = \sum_{m=1}^{M} \kappa_m V_\alpha(t - z_m). \tag{3}$$

Also define $[a, b]$ to be the n dimensional box $\{x \in \mathbb{R}^n : a \le x \le b\}$, and e to be the n-vector $[1, 1, \ldots, 1]^T$.

§3. Moment Expansions

Lemma 1 below shows that we can form a polynomial approximation s_1 to a spline s of the form (1) by combining moments of the weights λ with the coefficients of a polynomial approximation q to Φ on a "double width" panel.

Lemma 1. Approximation via moments - correlations. *Let c, $d \in \mathbb{R}^n$ with c, $d > 0$. Let $t \in \mathbb{R}^n$ with $|t_i| > c_i + d_i$ for $1 \le i \le n$. Let $\epsilon > 0$ and Φ be a function in $C[t - (c + d), t + (c + d)]$. Let*

$$q(x) = \sum_{\{\alpha : 0 \le \alpha \le ke\}} a_\alpha V_\alpha(x - t)$$

be a polynomial of coordinate degree k such that

$$\|\Phi - q\|_{L^\infty[t-(c+d), t+(c+d)]} \le \epsilon .$$

Given centres $z_1, z_2, \ldots z_M$ with $z_m \in [-d, d] \subset \mathbb{R}^n$ for $1 \le m \le M$, and weights $\kappa_1, \kappa_2, \ldots, \kappa_m \in \mathbb{R}$, let the corresponding "radial" basis function

$$s(x) = \sum_{m=1}^{M} \kappa_m \Phi(x - z_m), \tag{4}$$

be approximated by

$$s_1(x) = \sum_{m=1}^{M} \kappa_m q(x - z_m) . \tag{5}$$

Then

$$\|s - s_1\|_{L^\infty[t-c, t+c]} \le \epsilon \|\kappa\|_1 . \tag{6}$$

Moreover,

$$s_1(x) = \sum_{\{\beta : 0 \le \beta \le ke\}} b_\beta V_\beta(x - t), \tag{7}$$

where

$$b_\beta = \sum_{\{\alpha : \beta \le \alpha \le ke\}} a_\alpha \sigma_{0, \alpha - \beta}. \tag{8}$$

Remark: Often in our applications of this lemma, Φ will be truly radial, i.e. of the form $\Phi(\cdot) = \phi(\| \cdot \|_2)$, for some function ϕ of one variable.

Proof: If $-c \le x - t \le c$, then for $1 \le m \le M$ we have

$$-(c + d) \le (x - z_m) - t \le c + d.$$

This shows all but the expression for s_1 in terms of the moments of the data. To see the latter, write

$$s_1(x) = \sum_{m=1}^{M} \kappa_m q(x - z_m)$$

$$= \sum_{m=1}^{M} \kappa_m \sum_{0 \le \alpha \le ke} a_\alpha V_\alpha(x - z_m - t)$$

$$= \sum_{m=1}^{M} \kappa_m \left\{ \sum_{0 \le \alpha \le ke} \frac{a_\alpha}{\alpha!} ((x - t) - z_m)^\alpha \right\}$$

$$= \sum_{m=1}^{M} \kappa_m \left\{ \sum_{0 \le \alpha \le ke} \frac{a_\alpha}{\alpha!} \sum_{0 \le \beta \le \alpha} \binom{\alpha}{\beta} (x - t)^\beta (-z_m)^{\alpha-\beta} \right\}$$

$$= \sum_{m=1}^{M} \kappa_m \left\{ \sum_{0 \le \alpha \le ke} a_\alpha \sum_{0 \le \beta \le \alpha} \frac{(x - t)^\beta}{\beta!} \frac{(-z_m)^{\alpha-\beta}}{(\alpha - \beta)!} \right\}$$

$$= \sum_{0 \le \alpha \le ke} a_\alpha \left\{ \sum_{0 \le \beta \le \alpha} V_\beta(x - t)\sigma_{0,\alpha-\beta} \right\}$$

$$= \sum_{0 \le \beta \le ke} \left\{ \sum_{\beta \le \alpha \le ke} a_\alpha \sigma_{0,\alpha-\beta} \right\} V_\beta(x - t). \quad \square$$

An efficient way to form the approximation s_1 is to use real FFTs to compute the correlation of equation (8). Nominally, such a process involves three FFTs per correlation. However, things may be arranged so that the transforms of moments of panels, and those of the required approximations to Φ, are computed once and used many times. Also sequences of coefficients can be summed in the transform domain rather than the function domain. This lowers the average number of FFTs per correlation dramatically. Thus, in 2 dimensions the work per correlation is reduced to approximately $\mathcal{O}(k^2) + bk^2 \log k$ operations, where b is small.

A proof similar to that of Lemma 1 shows the following total degree version of the approximation via moments lemma.

Lemma 2. *Let $c, d \in \mathbb{R}^n$ with $c, d > 0$. Let $t \in \mathbb{R}^n$ with $|t_i| > c_i + b_i$, $1 \leq i \leq n$. Let $\epsilon > 0$ and Φ be a function in $C[t - (c + d), t + (c + d)]$. Let*

$$q(x) = \sum_{\{\alpha : 0 \leq \alpha \text{ and } |\alpha| \leq k\}} a_\alpha V_\alpha(x - t),$$

be a polynomial of total degree k such that

$$\|\Phi - q\|_{L^\infty[t-(c+d), t+(c+d)]} \leq \epsilon .$$

Given centres $z_1, z_2, \ldots, z_M$ with $z_m \in [-d, d]$ for $1 \leq m \leq M$ and weights $\kappa_1, \kappa_2, \ldots, \kappa_M \in \mathbb{R}$, let the corresponding "radial" basis function

$$s(x) = \sum_{m=1}^{M} \kappa_m \Phi(x - z_m),$$

be approximated by

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

Then

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

Moreover,

$$s_1(x) = \sum_{\{\beta : 0 \leq \beta, \ |\beta| \leq k\}} b_\beta V_\beta(x - t),$$

where

$$b_\beta = \sum_{\{\alpha : \beta \leq \alpha, |\alpha| \leq k\}} a_\alpha \sigma_{0, \alpha - \beta} .$$

The next lemma shows that shifted moments can be expressed as a convolution of moments about a given point. This result will be used in generating the moments corresponding to a larger panel of centres indirectly from the moments corresponding to subpanels. The indirect process will be more efficient than direct formation when the number of centres is large because the operation count for the indirect shift depends on the order of the moments, not on the number of centres.

Lemma 3. *Indirect shifting of moments – convolutions. Let $z_1, \ldots, z_M$ be given points in $\mathbb{R}^n$ and $\kappa_1, \ldots, \kappa_M$ be corresponding weights. Let $\sigma_{v, \alpha}$ be the α-th normalized moment of the data defined in equation (3). Then for all $v, u \in \mathbb{R}^n$ and multi-integers α*

$$\sigma_{v, \alpha} = \sum_{0 \leq \beta \leq \alpha} V_\beta(v - u)\sigma_{u, \alpha - \beta} . \tag{9}$$

Proof:

$$\sigma_{v,\alpha} = \frac{1}{\alpha!} \sum_{m=1}^{M} \kappa_m (v - z_m) = \frac{1}{\alpha!} \sum_{m=1}^{M} \kappa_m (v - u + u - z_m)^\alpha$$

$$= \frac{1}{\alpha!} \sum_{m=1}^{M} \kappa_m \sum_{0 \le \beta \le \alpha} \binom{\alpha}{\beta} (v - u)^\beta (u - z_m)^{\alpha - \beta}$$

$$= \sum_{m=1}^{M} \kappa_m \sum_{0 \le \beta \le \alpha} V_\beta (v - u) V_{\alpha - \beta} (u - z_m)$$

$$= \sum_{0 \le \beta \le \alpha} V_\beta (v - u) \sigma_{u, \alpha - \beta}. \quad \square$$

In using Lemma 3 to translate moments we can reduce the operation count by using either FFT convolution or a tensor product approach.

We discuss firstly the tensor product approach. The tensor product approach may be viewed as making a shift from u to v not in a single step, but as a series of shifts in the coordinate directions. For simplicity we will discuss only the 2-dimensional case.

Consider the formula (9). If $v - u = (x, 0)$, then we see immediately that $V_\beta(v - u)$ is nonzero only when the second component of β, β_2, is zero. Thus for v having the same second component as u,

$$\sigma_{v,\alpha} = \sum_{0 \le \beta_1 \le \alpha_1} V_{\beta_1}(v_1 - u_1) \sigma_{u, \alpha - (\beta_1, 0)} . \tag{10}$$

Considering $\sigma_{u,\gamma}$ as an array indexed by γ and the calculation of moments of degree not exceeding k, equation (10) above implies that each row of $\sigma_{v,\alpha}$ may be calculated in $\mathcal{O}(k^2)$ flops. Thus if v and u have the same second component all moments of degree not exceeding k can be shifted in $\mathcal{O}(k^3)$ flops. Similar remarks apply to $\{\sigma_{w,\alpha}\}$ and $\{\sigma_{v,\alpha}\}$ when w and v differ only in their second components. Thus, in the 2-dimensional case this tensor product strategy reduces the flop count for a single shift of all moments of degree not exceeding k from $\mathcal{O}(k^4)$ to $\mathcal{O}(k^3)$.

An alternative is to use FFT convolution to compute the transformation of the moment shifting lemma, Lemma 3. The corresponding operation count is $\mathcal{O}(k^2 \log k)$ in the 2-dimensional case.

§4. Symmetry and Approximations to Φ

In this section we will show how symmetry considerations can greatly reduce the number of approximations to Φ that need to be computed. In the 2-dimensional situation, with a quad tree subdivision of space, and without clumping, there are 40 different geometries of source and target for each level. The method requires approximations to Φ on all the corresponding double rectangles. However, for most choices of Φ the number of approximations that need to be computed from scratch is reduced to 7 by the symmetry relations of Lemma 4 below. Related symmetry considerations for the fast multipole method are discussed in Wang and LeSar [7].

Lemma 4. Symmetries and approximations of "radial" functions. Let $\mathcal{J}_1, \ldots, \mathcal{J}_n$, be subsets of $\mathbb{R}$ and $1 \leq p \leq \infty$, and let Φ and q be functions in $L^p(\mathcal{J}_1 \times \cdots \times \mathcal{J}_n)$.

- Suppose Φ is an even function of the k-th component of x, x_k. Define a function $\tilde{q}$ by

$$\tilde{q}(x_1, \ldots, x_k, \ldots, x_n) = q(x_1, \ldots, -x_k, \ldots, x_n), \qquad (11)$$

for all $(x_1, \ldots, x_n) \in \mathcal{J}_1 \times \ldots \times -\mathcal{J}_k \times \ldots \times \mathcal{J}_n$. Then

$$\|\Phi - \tilde{q}\|_{L^p(\mathcal{J}_1 \times \cdots \times -\mathcal{J}_k \times \cdots \times \mathcal{J}_n)} = \|\Phi - q\|_{L^p(\mathcal{J}_1 \times \cdots \times \mathcal{J}_k \times \cdots \times \mathcal{J}_n)}. \qquad (12)$$

- Suppose $\Phi(y)$ is unchanged by permutation of the components $y_1, \ldots, y_n$, of y. Let π be any permutation of the integers $\{1, 2, \ldots, n\}$, let π^{-1} be its inverse, and define $\tilde{q}$ by

$$\tilde{q}(y_1, \ldots, y_n) = q(y_{\pi^{-1}(1)}, \ldots, y_{\pi^{-1}(n)}), \qquad (13)$$

for all $y \in \mathcal{J}_{\pi(1)} \times \cdots \times \mathcal{J}_{\pi(n)}$. Then

$$\|\Phi - \tilde{q}\|_{L^p(\mathcal{J}_{\pi(1)} \times \cdots \times \mathcal{J}_{\pi(n)})} = \|\Phi - q\|_{L^p(\mathcal{J}_1 \times \cdots \times \mathcal{J}_n)}. \qquad (14)$$

Proof: Let Φ, q and $\tilde{q}$ be as in the statement of the first part of the lemma. Let $x = (x_1, \cdots, x_k, \cdots x_n)$ be a point in $\mathcal{J}_1 \times \ldots \times -\mathcal{J}_k \times \ldots \times \mathcal{J}_n$. Using the evenness of Φ in the k-th component of x,

$$|\Phi(x_1, \ldots, x_k, \ldots, x_n) - \tilde{q}(x_1, \ldots, x_k, \ldots, x_n)|$$
$$= |\Phi(x_1, \ldots, -x_k, \ldots, x_n) - q(x_1, \ldots, -x_k, \ldots, x_n)|$$
$$= |\Phi(x_1, \ldots, u, \ldots, x_n) - q(x_1, \ldots, u, \ldots, x_n)|,$$

where $(x_1, \cdots, u, \cdots, x_n)$ is a point in $\mathcal{J}_1 \times \cdots \times \mathcal{J}_n$. The first part of the lemma follows by using this equality in the appropriate p^{th} power integrals and essential supremum.

We turn now to the second part of the lemma. Let Φ, q, and $\tilde{q}$ be as in the statement of that part of the lemma. Let $y = (y_1, \ldots, y_n)$ be a point in $\mathcal{J}_{\pi(1)} \times \cdots \times \mathcal{J}_{\pi(n)}$. Using that $\Phi(y)$ is unchanged by permutations of the components of y, and defining $x = (x_1, \ldots, x_n) = (y_{\pi^{-1}(1)}, \ldots, y_{\pi^{-1}(n)})$,

$$|\Phi(y_1, \ldots, y_n) - \tilde{q}(y_1, \ldots, y_n)| = |\Phi(y_1, \ldots, y_n) - q(y_{\pi^{-1}(1)}, \ldots, y_{\pi^{-1}(n)})|$$
$$= |\Phi(x_1, \ldots, x_n) - q(x_1, \ldots, x_n)|,$$

where $(x_1, \ldots, x_n)$ is a point in $\mathcal{J}_1 \times \cdots \times \mathcal{J}_n$. The result follows by using this equality in the appropriate p^{th} power integrals and essential supremum. $\square$

Remark: Suppose $\Phi(\cdot) = \phi(\|\cdot\|_p)$ for some function of one variable ϕ and some p-norm for $\mathbb{R}^n$, $1 \leq p \leq \infty$. Then Φ is even in all the components of x,

33 (0y)	34 (1y)	35 (2y)	36 (3y)	37 (7y)	38 (8y)	39 (9y)
26 (10y)	27 (4y)	28 (5y)	29 (6y)	30 (11y)	31 (12y)	32 (13y)
22 (14y)	23 (15y)				24 (16y)	25 (17y)
18 (3t)	19 (6t)		■		20 (19x)	21 (18x)
14 (2t)	15 (5t)				16 (15x)	17 (14x)
10 (1t)	4	5	6	11 (5x)	12 (4x)	13 (10x)
0	1	2	3	7 (2x)	8 (1x)	9 (0x)

Fig. 1. Symmetry in the 2D setting.

and is also unchanged by permutations of the components of x. Hence, this lemma applies to all such *generalized radial* functions and allows us to use symmetry to obtain approximations to Φ on new regions from those on old.

Figure 1 shows the geometry of source panels to target panels in $\mathbb{R}^2$ when we use a quad tree subdivision of $\mathbb{R}^2$ without clumping. The solid black square is the target, or evaluation panel, and the possible source panels are numbered 0 through 39. Actually, the sources in the left-most column and bottom-most row would not be used for the illustrated position of the evaluation panel within its parent. However, they would be used for different positions of the target. If the source is $[-d, d]$ and the target is $[t-c, t+c]$, then Lemma 1 requires an approximation q to Φ on the "double width" rectangle $[t - (c + d), t + (c + d)]$. If Φ has all the symmetries of Lemma 4, then at each level only the seven approximations corresponding to source panels $0, \ldots, 6$ need be calculated directly. The 33 other approximations are easily obtained by symmetry. The relevant symmetries to use on a previously calculated approximation are indicated in parentheses in panels 7 to 39. For example, the notation $(2x)$ in source panel 7 indicates that the approximation for source panel 7 is obtained from that for source panel 2 by symmetry in x. In the function domain this corresponds to negating the coefficients of odd powers of x. In the Fourier domain it corresponds to a block rearrangement of columns or rows, depending on which correspond to x. Similarly, the notation $(1t)$ in source panel 10 indicates that the approximation for source panel 10 can be obtained from that for source panel 1 by symmetry in x and y. This corresponds to a transpose operation on the coefficients in both the function and Fourier domains.

§5. Numerical Results

Some numerical results from a primitive implementation of the algorithm are given in Tables 1–4. In this implementation the core tensor product polynomial approximations q to Φ, employed as in Lemma 1, are formed by interpolation at shifted and scaled Chebyshev nodes followed by economisation. The moments, and the coefficients of the approximations to Φ, typically have a wide dynamic range when the polynomial degree is 15 or more. Consequently, some extra device is needed in order to make the algorithm stable, especially when the FFT is used. In the current code the method used is a scaling of moments and polynomial coefficients analogous to that suggested by Greengard and Rokhlin [6] for the 2–D fast multipole method. The device may be viewed as scaling every panel at every level to be $[-2, 2]^2$.

In the calculations reported, the centres are approximately uniformly distributed on $[0, 1]^2$. If the number of centres is N, then in Table 1 the number of evaluation points is the smallest perfect square bigger than N, and in Table 2 the number of evaluation points is the smallest perfect square bigger than $10N$. In both cases $\Phi([x, y]) = \sqrt{c^2 + x^2 + y^2}$, where $c = 1/\sqrt{N}$, and all the coefficients κ_m of the spline (4) are 1. The piecewise tensor product bivariate polynomials used are of coordinate degree 7.

Tab. 1. Moment based method versus direct evaluation.

# of centres	Ord. Alg. time	FFT Alg. time	Direct time	Ratio	Abs. error	Rel. error
4000	0.41	0.27	3.57	13.16	3.99E-04	1.30E-07
8000	0.84	0.58	14.90	25.51	2.42E-03	3.95E-07
16000	1.78	1.23	59.98	48.82	1.71E-03	1.40E-07
32000	3.64	2.65	237.47	89.75	9.30E-03	3.79E-07

Tab. 2. Moment based method versus direct evaluation.

# of centres	Ord. Alg. time	FFT Alg. time	Direct time	Ratio	Abs. error	Rel. error
4000	1.37	0.97	35.14	36.28	5.49E-04	1.79E-07
8000	2.88	2.00	148.98	74.49	4.39E-03	7.16E-07
16000	5.87	4.17	593.40	142.23	2.07E-03	1.69E-07
32000	12.51	8.98	2356.92	262.33	1.82E-02	7.43E-07

The timings in the tables are in seconds on an Intel pentium based machine. Timings are given for direct evaluation and for the algorithm both with and without the speed benefits of FFT convolutions and correlations. Tables 3 and 4 repeat the runs with polynomials of coordinate degree 15.

Tab. 3. Moment based method versus direct evaluation.

# of centres	Ord. Alg. time	FFT Alg. time	Direct time	Ratio	Abs. error	Rel. error
4000	1.19	0.81	3.57	4.42	3.47E-08	1.13E-11
8000	2.97	1.70	14.90	8.74	1.74E-07	2.84E-11
16000	5.93	3.55	59.98	16.88	1.35E-07	1.10E-11
32000	13.19	7.74	237.54	30.69	6.17E-07	2.52E-11

Tab. 4. Moment based method versus direct evaluation.

# of centres	Ord. Alg. time	FFT Alg. time	Direct time	Ratio	Abs. error	Rel. error
4000	4.20	2.19	35.14	16.06	2.49E-08	8.11E-12
8000	9.32	4.51	148.37	32.89	1.09E-07	1.77E-11
16000	19.16	9.32	593.40	63.65	8.48E-08	6.91E-12
32000	39.70	19.52	2356.92	120.76	3.86E-07	1.57E-11

References

1. Beatson, R. K. and G. N. Newsam, Fast evaluation of radial basis functions I, Comput. Math. Applic., **24** (1992), 7–19.

2. Beatson, R. K. and L. L. Greengard, A short course on fast multipole methods, in *Wavelets, Multilevel Methods and Elliptic PDEs*, M. Ainsworth and J. Levesley and W. A. Light and M. Marletta (eds), Oxford University Press, 1997, 1–37.

3. Beatson, R. K. and W. A. Light, Fast evaluation of radial basis functions: methods for two-dimensional polyharmonic splines, IMA J. Numer. Anal., **17** (1997), 343–372.

4. Beatson, R. K. and G. N. Newsam, Fast evaluation of radial basis functions: Moment based methods, SIAM J. Sci. Comput., **19** (1998), 1428–1449.

5. Greengard, L. L. and V. Rokhlin, A fast algorithm for particle simulations, J. Comput. Phys., **73** (1987), 325–348.

6. Greengard, L. L. and V. Rokhlin, On the efficient implementation of the fast multipole algorithm, Research report 602, Yale University Department of Computer Science, 1988.

7. Wang, H. Y. and R. LeSar, An efficient fast multipole algorithm based on expansion in solid harmonics, J. Chem. Phys., **104** (1995), 4173–4179.

Department of Mathematics and Statistics,
University of Canterbury,
Private Bag 4800, Christchurch, New Zealand
{R.Beatson,E.Chacko}@math.canterbury.ac.nz