Multi–Level Approximation to Scattered Data Using Inverse Multiquadrics

S. J. Hales and J. Levesley

Abstract. A method of finding local approximations is used to thin data before a hierarchical iterative refinement scheme is employed in conjunction with domain decomposition. The interpolation problem on each sub–domain is solved by using the same stored inverse. The approximation power of the inverse multiquadric is exploited whilst overcoming the computational difficulties associated with globally supported basis functions.

§1. Introduction

Radial basis functions have been widely used for multivariate interpolation of scattered data, see [4] for a summary. An interpolant is generated by a linear combination of basis functions ϕ at distinct centres x_i, $i = 1, \ldots, N$;

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

constrained by $s(x_i) = f_i$, $i = 1, \ldots, N$, where $\mathcal{F} : \mathbb{R}^d \mapsto \mathbb{R}$ and $f_i = \mathcal{F}(x_i)$. The interpolation matrix $A \in \mathbb{R}^N \times \mathbb{R}^N$ is given by $A_{1 \leq i,j \leq N} = \phi(\|x_i - x_j\|)$, and λ satisfies

$$A\lambda = f, \tag{2}$$

where $\lambda = [\lambda_1 \cdots \lambda_N]^T$ and $f = [f_1 \cdots f_N]^T$.

Common choices for ϕ in this setting are given in [6],

$$\begin{aligned} \phi(\|x - x_i\|) &= \exp(-c^2\|x - x_i\|^2), && \text{Gaussian,} \\ \phi(\|x - x_i\|) &= (c^2 + \|x - x_i\|^2)^{-1/2}, && \text{Inverse multiquadric,} \end{aligned}$$

where c is a constant shape parameter. With a small modification to the scheme, the thin plate spline and multiquadric are also used.

Curve and Surface Fitting: Saint-Malo 1999
Albert Cohen, Christophe Rabut, and Larry L. Schumaker (eds.), pp. 247–254.

ISBN 0-8265-1357-3.

The above parameter–dependent functions are good at approximating data for certain values of c, but these cause inherent ill–conditioning in A. Schaback [7] explains this phenomenon by means of an "Uncertainty Relation" between upper bounds on errors for interpolants of the form (1), and lower bounds on the smallest eigenvalue of A. Iterative techniques for solving such badly conditioned systems often suffer from poor rates of convergence, and therefore computationally expensive direct methods have to be employed.

In the inverse multiquadric case, large values of c achieve good initial approximations to smooth data, whilst smaller values produce functions capable of resolving fine detail. Ideally, such properties could be exploited without having to solve (2) directly.

Since s is evaluated at *points* $y \neq x_i$ where an approximation is required, a global solution incorporating all N centres may be inappropriate. Further, it is unnecessary to find $\hat{\lambda}$ such that $\|f - A\hat{\lambda}\|_\infty \ll |\mathcal{F}(y) - s(y)|$, since the accuracy of $s(y)$ is limited by the approximating power of ϕ. Rather than searching for a complete global solution, this suggests that attention may be focussed on small regions around evaluation points. Moreover, the aim is to obtain a solution such that the residual and approximation accuracy are comparable, for little is to be gained by having a small residual, while the approximation power of the basis functions limits the final accuracy.

In Section 2, local approximations are used to convert irregular data to a regular mesh of approximate function values. Whilst the method can be generalised to $\mathbb{R}^d$, the description and examples are given in $\mathbb{R}^2$. The system of equations associated with the gridded data is inverted and used to solve subsequent systems.

Floater & Iske [1] demonstrate the benefits of a multi–level approach to approximation, and the theoretical foundation is provided by Narcowich, Schaback & Ward [5]. Section 3 describes the present hierarchical iterative refinement algorithm, and explains the computational advantages of domain decomposition and the use of a stored inverse.

§2. Local Solutions and Gridding Data

If the function $\mathcal{F}$ is not arbitrary, but arises from a physical system, then some degree of smoothness can be assumed. A smooth data set can be significantly thinned whilst retaining general information about its behaviour. Floater & Iske [1] demonstrate that Delaunay triangulation can be used to optimise the uniformity of data, and provide a good thinning algorithm. Such triangulation and assembling of data is computationally expensive for excessively large N. An $\mathcal{O}(N)$ method of finding uniform approximate data is presented.

An approximation to $\mathcal{F}$ at a point $y \in \mathbb{R}^d$ is achieved by solving a small interpolation problem centred on y. The closest q points in X to y are interpolated by inverse multiquadrics with shape parameter c_{local}, and evaluated at y. Since q can be as low as $20 \sim 30$, c_{local} can be relatively large before the matrix ill-conditioning becomes unacceptable, thus yielding a good approximation.

This method is highly parallelizable, and large data–sets can be dealt with without the need of assembling or storing the matrix A.

Finding the optimum shape parameter on regular or scattered data remains an open problem, as shown in [3]. There is no obvious correlation between point spacing and a good choice of c_{local}. The best shape parameter is generally found by increasing the value of c_{local} until just prior to machine precision breakdown.

Let $Y = \{y_1, \ldots, y_{n^2}\}$ be the set of points on the $n \times n$ regular unit grid. If the previous local approximation technique is applied to each y_i, then the irregular data can be transformed to a regular grid with approximate function values $\hat{f}_i$. The aim is to find a global approximation using the new data at the grid points.

After converting scattered data to a regular grid, certain approximation techniques become available which would otherwise have been difficult to implement. Polynomial tensor product splines can be efficiently employed to approximate a solution from the given gridded data. To find such an approximation at a point z, z must lie inside a $(d+1) \times (d+1)$ subgrid of the regular points, where d is the degree of the Lagrange polynomials to be used. Let the points of such a subgrid be labelled ξ_{ij} and have function values $\hat{f}_{ij}$ for $i, j = 1, \ldots, d+1$. The univariate Lagrange polynomials $L_i(x)$ and $L^j(y)$ are constructed such that

$$L_i(\xi_{k\bullet}) = \delta_i^k \quad \text{and} \quad L^j(\xi_{\bullet k}) = \delta_j^k.$$

The polynomial tensor product spline ϕ_{ij} is defined to be

$$\phi_{ij}(z) = L^i(z) \cdot L_j(z).$$

The approximation at z is given by

$$\sum_{i=1}^{d+1} \sum_{j=1}^{d+1} \phi_{ij}(z) \hat{f}_{ij}.$$

Alternatively, a thinned global interpolant of the form (1) can be achieved by solving

$$B\mu = \hat{f}, \tag{3}$$

where $B_{1 \le i,j \le n} = \phi(\|y_i - y_j\|)$, $\mu = [\mu_1 \cdots \mu_n]^T$ and $\hat{f} = [\hat{f}_1 \cdots \hat{f}_n]^T$.

This amounts to finding an interpolant $\hat{s}$ to a thinned approximation of the initial data. The local approximation errors $|f(y_i) - \hat{f}(y_i)|$ limit the final accuracy of $\hat{s}$.

The inverse of B need only be computed once, and then stored for future use. All scattered data problems can then be scaled and transformed to the regular grid Y, whereupon μ is given by the matrix–vector product $\mu = B^{-1}\hat{f}$. Only half of the entries of B^{-1} need to be stored since $B^{-1} = (B^{-1})^T$.

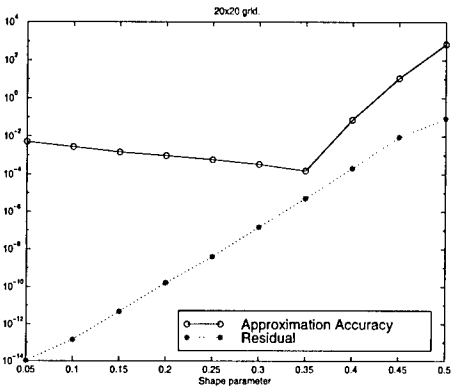

Fig. 1. An example of approximation vs. residual.

Too large a matrix B causes storage problems, and difficulties in calculating the inverse. As c increases, the approximation improves, but the residual $\|\hat{f} - B\mu\|_\infty$ grows. A value for c is chosen before the approximation begins to deteriorate due to the rise in the residual. As an example, the function $\mathcal{F} \equiv 1$ is approximated on the unit square using inverse multiquadrics by interpolating $\hat{f}_i = \mathcal{F}(y_i)$, $i = 1, \ldots, 400$ using (3). The approximation is evaluated at 1000 random points. The results in Figure 1 are typical for smooth functions, but the consequent choice of c is only a guide, and does not guarantee success for all $\mathcal{F}$.

§3. Hierarchical Iterative Refinement

The hierarchical method uses increasingly dense subsets of X to refine the current approximation; see [5]. Let $\chi_k = \{\hat{x}_1, \ldots, \hat{x}_{N_k}\} \subseteq X$, such that $N_{k+1} > N_k$. Let s_k be the current approximation, and r_k be the full global residual at the k^{th} level,

$$r_k(x_i) = f(x_i) - s_k(x_i).$$

Let $\hat{r}_k$ be the k^{th} residual over the points in χ_k. This thinned global residual is interpolated by

$$t_k(x) = \sum_{i=1}^{N_k} \gamma_i \phi_k(\|x - \hat{x}_i\|), \tag{4}$$

where $\phi_k(\|x-\hat{x}_i\|) = (c_k^2 + \|x-\hat{x}_i\|^2)^{-1/2}$, and $t_k(\hat{x}_i) = \hat{r}_k(\hat{x}_i)$, $i = 1, \ldots, N_k$.

The initial interpolant $s_1 \equiv 0$ is updated by

$$s_{k+1} = s_k + t_k. \tag{5}$$

The technique of gridding data in Section 2 is used to find an approximate function value for every point in Y_p. Therefore, (5) is replaced by $\hat{s}_{k+1} = \hat{s}_k + t_k$, where $\hat{s}_k$ is the current approximation to regular approximate data, $\hat{s}_1 \equiv 0$.

The value c_1 can be relatively large to give a good initial approximation. As N_k increases, c_k has to be reduced to ensure computational solvability. The decrease in c_k introduces tighter basis functions which improve the resolution of the approximation.

A method of data thinning is required to determine the points in χ_k. The dense systems arising from (4) have to be solved directly, but this is impractical for large N_k. To overcome such complications, domain decomposition is applied to each χ_k.

The levels of the hierarchy have to be computed sequentially, but by using domain decomposition each sub–domain can be dealt with in parallel. Moreover, each such solution only requires a single matrix–vector product.

To put this in the current context, each χ_k is constructed from overlapping square grids Y_p , where $p = 1, \ldots, m_k$. These square grids need not be the same size or of similar orientation, but must contain an equal number of points. Each sub–domain Y_p consists of an inner region, where the approximation is finally evaluated, and an overlap. Special attention has to be given to sub–domains whose edges coincide with the boundary of X.

At the k^{th} level, m_k sub–domain interpolation problems need to be solved. Since B is invariant under shifts and rotations of the centres y_i, the stored B^{-1} can be invoked. If the centres are scaled $y_i \mapsto \alpha y_i$, this amounts to a change in the shape parameter.

Recall that $B_{1\le i,j\le n} = \phi(\|y_i - y_j\|)$, where $y_i \in [0,1] \times [0,1]$. Now,

$$\begin{aligned}\phi(\|y - y_i\|) &= (c^2 + \|y - y_i\|^2)^{-1/2} \\ &= \alpha(\alpha^2 c^2 + \|\alpha y - \alpha y_i\|^2)^{-1/2}.\end{aligned}$$

Let $w_i = \alpha y_i$ and define $\psi(\|w - w_i\|) = \alpha(\alpha^2 c^2 + \|w - w_i\|^2)^{-1/2}$. Then $B_{1\le i,j\le n} = \phi(\|y_i - y_j\|) = \psi(\|w_i - w_j\|)$ where $w_i \in [0,\alpha]\times[0,\alpha]$. Therefore by using the matrix B, a new inverse multiquadric is created at scaled points with shape parameter αc.

Each of the thinned global interpolation problems (4) can be decomposed and solved by multiple applications of the stored inverse B^{-1}. Continued use of the same inverse naturally introduces tighter basis functions suitable for approximating typical residuals.

§4. Numerical Results

We give an example where the above scheme is used to approximate Franke's function [2] over 10000 scattered points in the unit square in $\mathbb{R}^2$:

$$\begin{aligned}\mathcal{F}(u,v) = \; & 0.75e^{-0.25(9u-2)^2-0.25(9v-2)^2} + 0.75e^{-(9u-2)^2/49-(9v-2)^2/10} \\ & + 0.5e^{-0.25(9u-7)^2-0.25(9v-3)^2} - 0.2e^{-(9u-4)^2-(9v-7)^2}.\end{aligned}$$

Level	No. of domains m_k	Shape parameter c_k	Overlap	Max. error in gridded data $\|f_i - \hat{f}_i\|_\infty$	Max. error in solution $\|\mathcal{F} - \hat{s}_k\|_\infty$
1	1	0.25	0	8.241×10^{-5}	1.371×10^{-4}
2	4	0.138	1/36	2.424×10^{-6}	1.838×10^{-5}
3	16	0.0688	1/72	3.189×10^{-6}	3.189×10^{-6}

Tab. 1. Error for Franke's function.

The localised interpolation problems are solved directly using Gauss Elimination, with $q = 20$ and $c_{local} = 0.2$. It is the error function at each level which is approximated locally, and not the original function $\mathcal{F}$. The square sub–domain grids Y_p are comprised of 21×21 equally spaced points. For ease of implementation, the sub–domains used for a particular level are of equal size. The overlaps between sub–domains therefore consist of one or two mesh points, depending on position. The key interpolation matrix B is constructed from inverse multiquadrics with $y_i \in [0,1]^2$, $c = 0.25$, and B^{-1} is generated using Matlab. The domain decomposition is straightforward on the unit square with $m_k = 4^{k-1}$. The thinned global interpolants $\hat{s}_k$ are evaluated at points $t_i \in [0,1]^2$. Table 1 shows the error in the approximated data at the regular grid points $|f(y_i) - \hat{f}(y_i)|$, and the error in the approximation $|\mathcal{F}(t_i) - \hat{s}_k(t_i)|$. Figures 2 and 3 show the approximation error for each level.

The error function from Level 1 clearly demonstrates the ability of the inverse multiquadric to approximate smooth data. The error near the boundary is scaled by an order of magnitude at each level, but has the same general behaviour. The final iteration leaves error near the boundary, aggravated by test points being outside the original scattered data set. Such evaluation points ought to be included since, although they require the extrapolation of $\hat{s}_k$ to evaluate, the experiment was specified to be conducted on the unit square. The original aim of finding a solution where the residual is comparable to the approximation accuracy is fulfilled at Level 3.

Example I is repeated as far as the regularization of data, and then polynomial tensor product splines are used to find the final approximation, as described in Section 2. Such splines cannot replace the inverse multiquadric approximation on the regular grid without an increase in error. Such an error is then propagated to the next level where the discrepancy is amplified. However, if the hierarchical refinement procedure is abandoned, then these basis functions efficiently yield a good approximation. Table 2 shows the approximation accuracy for such splines of different polynomial degree without iterative refinement. The grid sizes are comparable to those used in the original example.

Grid Size	Linear	Quadratic	Cubic
21×21	2.2×10^{-2}	6.4×10^{-3}	3.5×10^{-3}
41×41	5.3×10^{-3}	9×10^{-4}	2.6×10^{-4}
81×81	1.4×10^{-3}	1×10^{-4}	5.7×10^{-5}

Tab. 2. Approximation error for the various splines.

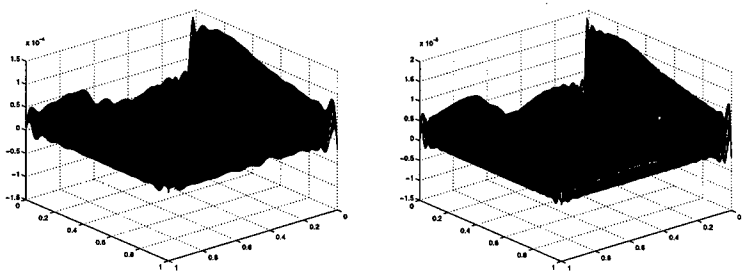

Fig. 2. Approximation Error for Levels 1 and 2.

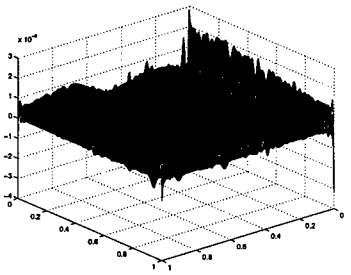

Fig. 3. Approximation Error for Level 3.

§5. Conclusion

A global solution to an interpolation problem involving a large number of data points is too expensive to compute directly if inverse multiquadrics are to be used effectively. However, if the aim is to generate approximations to a function, then such a solution is unnecessary, and an alternative method has been presented.

The underlying idea is to transform given scattered data f_i at points x_i to regular approximate data $\hat{f}_i$ at y_i, which is easier to solve for. The aim is then no longer to interpolate the initial data, but to find a good approximation to it. The final solution $\hat{s}$ is an approximation to $\hat{f}$, which is close to f. Success relies on minimising the local approximation errors $|f(y_i) - \hat{f}(y_i)|$.

The algorithm is $\mathcal{O}(N)$ since the only work related to the number of initial points is the search for the q closest points to each y_i. Such a search can be improved by making assumptions as to which x_i are unlikely to qualify.

The time required to solve each sub–domain problem is reduced due to the use of the stored $n \times n$ key inverse matrix. Solving directly would be $\mathcal{O}(n^3)$, but the required matrix–vector product is $\mathcal{O}(n^2)$.

The hierarchical iterative refinement strategy produces good approximations, and is the only sequential aspect of the method. The search for approximate regular data, and the solutions for each decomposed sub–domain are parallelizable operations, although this has yet to be implemented. These features mean that large data sets can be dealt with in acceptable computing time.

References

1. Floater, M. S. and A. Iske, Multistep scattered data interpolation using compactly supported radial basis functions, J. Comput. Appl. Math. **73** (1996), 65–78.
2. Franke, R., Scattered data interpolation: test of some methods, Math. Comp. **38** (1982), 181–200.
3. Hagan, R. E. and E. J. Kansa, Studies of the R parameter in the multiquadric function applied to groundwater pumping, Journal of Applied Science & Computations **1** (1994), 266–282.
4. Kansa, E. J. and R. E. Carlson, Radial Basis Functions: A class of grid–free, scattered data approximations, Computational Fluid Dymanics Journal **3** (1995), 479–496.
5. Narcowich, F. J., R. Schaback, and J. D. Ward, Multilevel interpolation and approximation, Appl. Comput. Harmonic Anal. to appear.
6. Powell, M. J. D., The theory of radial basis function approximation, Cambridge University, DAMTP 1990/NA11, 1990.
7. Schaback, R., Multivariate Interpolation and Approximation by Translates of a Basis Function, in *Approximation Theory VIII*, C. K. Chui & L. L. Schumaker (eds), Approximation and Interpolation, Texas, 1995, 491–514.

Stephen Hales
Mathematics & Computer Science Dept.
University of Leicester
University Road
Leicester, LE1 7RH, UK
sjh16@mcs.le.ac.uk

Jeremy Levesley
Mathematics & Computer Science Dept.
University of Leicester
University Road
Leicester, LE1 7RH, UK
jl1@mcs.le.ac.uk