Adaptive Greedy Algorithm for Solving Large RBF Collocation Problems

Y.C. Hon, R. Schaback, and X. Zhou*

Abstract

The solution of operator equations with radial basis functions by collocation in scattered points leads to large linear systems which often are non–sparse and ill–conditioned. But one can try to use only a subset of the data for the actual collocation, leaving the rest of the data points for error checking. This amounts to finding "sparse" approximate solutions of general linear systems arising from collocation. This contribution proposes an adaptive greedy method with proven (but slow) linear convergence to the full solution of the collocation equations. The collocation matrix need not be stored, and the progress of the method can be controlled by a variety of parameters. Some numerical examples are given.

1 Introduction

Consider an abstract operator equation

$$Lu = f,\ L\ :\ X \to Y, \tag{1}$$

for a linear mapping between Hilbert spaces X and Y, where $f \in Y$ is given. The operator L may be composed of some differential equation on the domain Ω and some boundary conditions on $\partial\Omega$, but we do not care for details here. The reader may think at this stage of a classical Dirichlet problem, but should be aware that we put both a differential equation and boundary conditions into a single operator, whose domain X and range Y will then consist of cartesian products of certain Sobolev spaces.

*The work described in this paper was supported by a grant from the German Academic Exchange Service and the Research Grants Council of the Hong Kong Joint Research Scheme (Project No. G_HK025/00)

Discretization of such a problem can be done by picking a finite number of continuous functionals μ on Y and replacing (1) by

$$\mu_j(Lu) = (\mu_j \circ L)(u) = \mu_j(f) =: f_j,\ 1 \le j \le N. \tag{2}$$

Using $\lambda_j := \mu_j \circ L$, we thus have a generalized interpolation problem

$$\lambda_j(u) = f_j,\ 1 \le j \le N, \tag{3}$$

but we have to keep in mind that the functionals λ_j may be rather peculiar, comprising evaluation of a differential operator in the interior or of a boundary operator on the boundary of the domain Ω. To avoid redundancies in the discretization, we can assume that the functionals $\lambda_j := \mu_j \circ L$ are linearly independent on X.

A straightforward numerical technique for solving (3) would simply pick an N–dimensional linear subspace S_N of X on which the functionals λ_j are continuous and linearly independent. If evaluated on a basis of S_N, the system (3) will lead to a nonsingular system of linear equations, if the above assumptions are satisfied. But it is by no means trivial to assure linear independence of the functionals on the space S_N. In particular, if S_N is fixed beforehand, the linear independence of more or less arbitrary functionals cannot be expected.

A possible way out of this dilemma is to use the functionals in the set $\Lambda = \{\lambda_1, \ldots, \lambda_N\}$ for a proper definition of the space S_N. This can be done by taking a sufficiently smooth symmetric function $\Phi : \Omega \times \Omega \to \mathbb{R}$ and setting

$$S_N := S_\Lambda := \text{ span } \left\{\lambda_j^x \Phi(x, \cdot) \ : \ 1 \le j \le N\right\},$$

where the superscript x at λ_j^x stands for evaluation of the functional λ_j with respect to the variable x. The system (3) then takes the form

$$\sum_{k=1}^{N} \alpha_k \lambda_j^x \lambda_k^y \Phi(x, y) = f_j,\ 1 \le j \le N \tag{4}$$

for a linear combination

$$s := \sum_{k=1}^{N} \alpha_k \lambda_k^y \Phi(\cdot, y). \tag{5}$$

Note that (4) now has a symmetric coefficient matrix

$$A_\Lambda := \left(\lambda_j^x \lambda_k^y \Phi(x, y)\right)_{1 \le j,k \le N} \tag{6}$$

and we have to ask for functions Φ which make the matrix nonsingular or even positive definite for any choice of linearly independent functionals on X. Fortunately, such *strictly positive definite* functions exist, and they are explicitly available. They can even be found in radial form

$$\Phi(x,y) = \phi(\|x-y\|_2), \quad \phi : [0,\infty) \to I\!R$$

with some scalar function ϕ, and the Gaussian $\phi(r) = \exp(-r^2)$ is a prominent example. Surprisingly there are also radial cases where ϕ is compactly supported. A simple and useful example is Wendland's function $\phi(r) = (1-r)_+^4(1+4r)$ which works in space dimensions up to 3 and (roughly) for all functionals that can be nicely applied to both arguments of $\Phi(x,y) = \phi(\|x-y\|_2)$.

This is the setting of *symmetric collocation* that we want to work in, but some remarks seem appropriate at this point. It is by no means a trivial task to pick the "right" function Φ for a given operator equation or a given class of linear functionals, and is even less trivial to derive bounds for the error $u-s$ between an exact solution u of (1) and an approximate solution s of the form (5) to the discretized system (4). We refer the reader to the literature on collocation with "radial" basis functions [2, 4, 5, 7, 8, 9, 10]. In this contribution, we focus on the numerical techniques to solve very large systems like (4) efficiently, making use of the background of "radial" basis functions. In addition, we concentrate on approximate solutions with only few nonzero coefficients α_j. The reason is that the evaluation of a full sum in (5) on many points will be too costly, if the sum contains a term for each data value. In short, we try to approximate N data with $K << N$ terms, and we want to keep the storage and computational effort proportional to N. This implies that we try to avoid storage of the full matrix A_Λ. In the context of finite element methods, the generation of coefficient vectors with many zeros would be undesirable, but for radial basis function techniques the opposite is true. The user is free to place the centers for the basis functions, and just a subset of the actual data points may be fully sufficient to yield a prescribed accuracy.

2 Native Space Norm

We shall make use of a striking property of strictly positive definite functions: they induce norms $\|.\|_\Phi$ on functionals and functions. On functionals, this works via the inner product

$$(\lambda,\mu)_\Phi := \lambda^x \mu^y \Phi(x,y).$$

Functions s of the form (5) can be written as $s_\mu := \mu^x \Phi(x, \cdot)$ with a finitely supported functional μ, and then one defines

$$(s_\lambda, s_\mu)_\Phi = (\lambda, \mu)_\Phi = \lambda^x \mu^y \Phi(x, y) \tag{7}$$

on functions. Naturally, these definitions generalize to inner products in certain Hilbert spaces induced by Φ, but for details we refer the reader to papers on these "native" spaces. There are good reasons to view this norm as kind of an energy. We remark in passing that *conditionally positive definite functions* would require a somewhat more complicated treatment, but we do not want to overload this presentation with technical details.

But there is another important feature to be exploited. A solution $s_{\Lambda,u}$ of the form (5) to the system (4) with data $f_j := \lambda_j(u)$ has minimal norm $\|.\|_\Phi$ under all other interpolating functions $s_\mu := \mu^x \Phi(x, \cdot)$ with *arbitrary* functionals μ. By standard variational arguments this implies the orthogonality relation

$$(s_{\Lambda,u}, u - s_{\Lambda,u})_\Phi = 0$$

for all u from the native space. If this is satisfied, the Pythagorean Theorem implies

$$\|u\|_\Phi^2 = \|u - s_{\Lambda,u}\|_\Phi^2 + \|s_{\Lambda,u}\|_\Phi^2, \tag{8}$$

and we shall make frequent use of this equation.

For the above argument to work we have to make sure that the operator equation (1) has a solution u that lies in the native space of the function Φ. In practice, this requires Φ to be sufficiently "nonsmooth" with respect to the smoothness of u induced by the regularity theory for the problem (1). Here lies a hidden obstacle for applications with solutions u of limited smoothness: there may be no nice and continuous Φ such that u lies in the native space for Φ. For instance, the radial function whose native space is the Sobolev space $W_2^1(I\!R^2)$ is the Bessel function $K_0(r)$ with a logarithmic singularity at zero. This, for instance, makes it hard to handle weak solutions $u \in W_2^1(I\!R^2)$ to elliptic problems on domains with incoming corners. But this problem can very probably be overcome by future research, because the current theory has so far concentrated on at least continuous functions.

A related additional condition is that the functionals λ_j must be continuous and linearly independent on the native space for Φ. This usually does not cause any problems. But we want to point out that the discrete solution of systems like (4) will be possible in many cases where the above theoretical requirements are not met, e.g. in situations where the operator equation (1)

is ill–posed, has a singular solution or no solution at all. The technique just produces a function satisfying the discretized equations and having minimal energy norm under all such functions. In other words, symmetric collocation acts like a regularization method, and naive users have to be aware of this fact, since the technique always produces some result, even if it belongs to a question that has no answers at all. On the other hand, if no other method is applicable, setting up a system (4) with a suitably picked positive definite Φ will nearly always work somehow, even if results have to be interpreted with great care.

3 Greedy Iteration on Residuals

The orthogonality relation (8) simply says that the energy of a function u can be split up into the energy of a generalized interpolant $s_{\Lambda,u}$ with respect to $\Lambda = \{\lambda_1, \ldots, \lambda_N\}$ plus the energy of the residual $f - s_{\Lambda,u}$. We shall apply this split recursively by interpolating the residual. For pure interpolation, a thorough analysis of such techniques was provided in [13]. Here, we deal with the case of generalized interpolation and concentrate on iterative techniques to approximate $s_{\Lambda,u}$ by functions of the form (5) that contain only a small number of nonzero terms. The basic idea is to use a sequence of generalized interpolants to just a single data functional each, acting on residuals of the previous step. Since the functional with the largest residual is picked, the technique can be called *greedy.*

Algorithm 1 *Assume data* $f_j =: f_j^0$, $1 \le j \le N$ *and functionals* $\Lambda = \{\lambda_1, \ldots, \lambda_N\}$ *to be given. Fix some* $\gamma \in (0, 1]$ *and iterate over an index* $k = 0, 1, \ldots$ *as follows:*

- *Find some index* j_k *with* $1 \le j_k \le N$ *such that*

$$|f_{j_k}^k| \ge \gamma \|f^k\|_\infty := \max_{1 \le j \le N} |f_j^k|, \tag{9}$$

  e.g. by taking the actual maximum.

- *Then overwrite all values* f_j^k *for* $1 \le j_k \le N$ *by*

$$f_j^{k+1} := f_j^k - f_{j_k}^k \frac{\lambda_j^x \lambda_{j_k}^y \Phi(x,y)}{\lambda_{j_k}^x \lambda_{j_k}^y \Phi(x,y)}$$

  and repeat the iteration for $k+1$ *instead of* k.

- *During the iterations, sum the functions*

$$s_k := \lambda_{j_k}^y \Phi(\cdot, y)/\lambda_{j_k}^x \lambda_{j_k}^y \Phi(x, y)$$

by adding $1/\lambda_{j_k}^x \lambda_{j_k}^y \Phi(x, y)$ *to the coefficient* α_{j_k} *in the representation* (5) *of the sum.*

This algorithm is very primitive, but it has (at least theoretically) an unexpectedly strong convergence property that we prove in the rest of this section.

Theorem 1 *If the functionals* λ_j *are linearly independent and continuous on the native space of* Φ*, then the sum of the functions* s_k *of Algorithm 1 converges at least linearly to the solution* $s_{\Lambda,u}$ *of the full system* (4)*. The norm* $\|f^k\|_\infty$ *of the residuals converges linearly to zero.*

Proof: The hypothesis of Theorem 1 allows to conclude that the solution $s_{\Lambda,u}$ of the system (4) exists and is unique among all functions s of the form (5). For later use we note that all such functions satisfy

$$c_1\|s\|_\Phi \le \|s\|_{\Lambda,\infty} := \max_{1\le j\le N} |\lambda_j(s)| \le C_1\|s\|_\Phi \tag{10}$$

with fixed constants $0 < c_1 \le C_1$, because we have equivalence of norms on finite–dimensional vector spaces. Now we see that Algorithm 1, when viewed as working on functions in the native space, uses

$$u_0 := s_{\Lambda,u}, s_k := s_{\lambda_{j_k},u_k}, u_{k+1} := u_k - s_k.$$

Note that the functions u_k are all of the form (5), but the algorithm actually does not work on the functions, but rather on the data $\lambda_j(u_k)$ of these functions. However, equation (8) implies

$$\|u_{k+1}\|_\Phi^2 = \|u_k\|_\Phi^2 - \|s_k\|_\Phi^2 \le \|u_k\|_\Phi^2.$$

This proves that the numbers $\|u_k\|_\Phi^2$ decay weakly monotonically, but we want linear convergence to zero. ¿From (7) we get

$$\|s_k\|_\Phi^2 = \frac{(f_{j_k}^k)^2}{\lambda_{j_k}^x \lambda_{j_k}^y \Phi(x, y)} = \frac{(f_{j_k}^k)^2}{\|\lambda_{j_k}\|_\Phi^2} \tag{11}$$

and we want to bound this norm from below. Now we look at (9) and use (10) to get

$$\begin{aligned} |f_{j_k}^k| &\ge \gamma\|f^k\|_\infty \\ &= \gamma\|u_k\|_{\Lambda,\infty} \\ &\ge \gamma c_1\|u_k\|_\Phi \end{aligned}$$

With

$$c_2 := \max_{1 \le j \le N} \|\lambda_j\|_\Phi$$

we then have

$$\|s_k\|_\Phi^2 \ge \frac{\gamma^2 c_1^2}{c_2^2} \|u_k\|_\Phi^2$$

and get linear convergence $\|u_k\|_\Phi \to 0$ from

$$\|u_{k+1}\|_\Phi^2 \le \|u_k\|_\Phi^2 \left(1 - \frac{\gamma^2 c_1^2}{c_2^2}\right).$$

The functions

$$g_m := \sum_{j=0}^{m} s_j = u_0 - u_{m+1} = s_{\Lambda,u} - u_{m+1}$$

must then converge linearly to $s_{\Lambda,u}$. Since everything happens in the linear space of functions (5), the choice of norm for linear convergence is irrelevant. □

Since c_1/c_2 will often be an extremely small number, linear convergence is hard to observe in practice. A weaker convergence behaviour according to

$$\begin{aligned} \|u_0\|_\Phi^2 - \|u_{m+1}\|_\Phi^2 &= \sum_{k=0}^{m} \left(\|u_k\|_\Phi^2 - \|u_{k+1}\|_\Phi^2 \right) \\ &= \sum_{k=0}^{m} \|s_k\|_\Phi^2 \end{aligned} \tag{12}$$

is much more near to reality. In fact, the sum on the right–hand side converges to $\|s_{\Lambda,u}\|_\Phi^2$ for $m \to \infty$. It shows how the "energy" of $s_{\Lambda,u}$ gradually builds up, and it allows easy monitoring via (11). In view of (12) and (11), a comparison of the sum of squares of $\|s_k\|_\Phi$ and $\|f^k\|_\infty$ reveals some information on the constants in the error analysis.

The algorithm is "greedy" in the sense that it tries to improve on the largest available residual, and it is adaptive in the sense that it automatically concentrates on regions where the residuals are still large. For variations of greedy algorithms in similar contexts, see [1, 14] and for adaptive use of radial basis functions, see [6, 11, 13].

Convergence of the algorithm is rather slow, but its merits for extremely large problems rely on another property:

It brings in one coefficient at a time, and, at least for very large problems, it never works on approximations that have the full number of nonzero coefficients.

Let us do a very rough analysis of its performance, based on the weaker convergence behaviour like $1/k$. After k steps the order of magnitude of the residuals will be brought down by a factor of $1/k$, and this is achieved by using only k approximating functions. One can possibly expect 1% accuracy after 100 steps, using just 100 coefficients.

This strategy does not make sense if one wants an **exact** solution of a system of, say, 100,000 data points. But it often does not make sense to use all 100,000 degrees of freedom to solve such a system exactly, coming up with a "solution" with 100,000 coefficients, whose sheer size limits its usefulness. It seems to be much more reasonable to get away with 1000 nonzero parameters that fit the data to an accuracy of 0.1%. The above algorithm adaptively picks data functionals (and corresponding coefficients) that are the best candidates for further treatment, and it turns out to be extendable to an algorithm that is the first to use radial basis functions of different scales adaptively. We shall address this in the next section.

4 Adaptive Scaling

We now want to look at a modification of Algorithm 1 that uses **scaled** radial basis functions $\Phi_c(x,y) := \phi(\|x-y\|_2/c^2)$. In particular, we aim at functions ϕ that have support in $[0,1]$, such that $\Phi_c(x,y)$ vanishes for $\|x-y\|_2 > c$.

Algorithm 2 *We fix a large starting scale c and real constants $0 < \alpha, \beta < 1 < \rho$. Furthermore, we need an iteration count $K \geq 1$, a scale index $i = 0$, another integer $m \geq 0$, and a fixed number M of scales to try locally.*

Until the discrete norm $\|f^k\|_\infty$ of residuals falls below a prescribed bound, an outer iteration tries to reduce the discrete norm of residuals by a factor of α. This situation is called "success" in what follows. A middle infinite loop tries larger and larger numbers $K, K\rho, K\rho^2, \ldots$ of iterations, and an inner finite loop tries scales $c\beta^i > c\beta^{i+1} > \ldots > c\beta^{i+M-1}$ until success is reached. In case of success using scale $c\beta^j$ and $K\rho^\ell$ iterations, we may (but not need to) set $i := \max(0, j-m)$ and $K := \max(1, K\rho^{\ell-m})$ to avoid return to extremely large scales and extremely small iteration counts. After success, we do another outer iteration.

Since we know that at any fixed scale Algorithm 1 must bring the residuals to zero after sufficiently many iterations, the middle loop must terminate at each of its finitely many scales. It terminates using the scale that roughly takes the fewest number of new points to reach success. Since the middle loop reduces the residual norm by a certain factor smaller than 1, any prescribed accuracy can be reached after sufficiently many outer iterations.

Note that the algorithm tries first to get away with as few new interpolation functionals as possible, using the smallest possible iteration count that leads to a reduction of the residuals. For each iteration count, it tests large scales first, but priority is given to the iteration count over the scale. For compactly supported functions, extremely small scales will have a very local effect and will not lead to any reasonable reduction in early stages of the algorithm. This means that the algorithm tends to prefer large scales over small scales at early stages.

Test runs indicate that the built–in restriction to increasing inner iteration counts and to decreasing scales helps to save computation time, because this behaviour can be observed in many cases when running without restrictions (i.e. when starting each iteration of the outer loop with $c = c\rho^0$ and $K = 1$, which does not spoil the convergence argument if the number of actually tested scales remains finite).

If applied for compactly supported radial basis functions, the algorithm in its above form reaches smaller and smaller scales, until the calculations can be localized and parallelized. This has not yet been fully exploited in the numerical examples following in the next section. As soon as localization sets in, it does not make any sense to insist on "few" nonzero coefficients, because the coefficients in use will then be tied to local behaviour of the residuals.

But we want to point out a further generalization. One can view the inner iteration just as a trial of M different radial basis functions, ignoring scale completely. Since the middle iteration increases the number of iterations for each function, it will automatically select the radial basis function that reaches success using the fewest centers. The inner loop must be finite, but after each success one can come up with a different set of M candidates for radial basis functions. It is easy to incorporate thin–plate splines or multiquadrics at early stages, and one can go over to compactly supported functions when it comes to resolving local details.

5 Numerical Results

In this section, the adaptive greedy method is demonstrated by solving the Poisson problem

$$\begin{array}{ll} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f & \text{in } \Omega \\ u = g & \text{on } \partial\Omega \end{array} \tag{13}$$

where the solution domain Ω is contained in the unit square $[0,1] \times [0,1]$, and where $\partial\Omega$ is the boundary of Ω. We discretized both the Laplacian data provided by f and the Dirichlet data provided by g. This was done by

- describing the domain via a parametrized periodic boundary curve plus an indicator function deciding whether a point (x, y) lies inside the domain,
- providing boundary data at equidistant parameters on the boundary curve,
- using these points on the boundary also for collocation of the Laplacian values,
- superimposing the domain with a fine regular grid of meshwidth h and keeping only those grid points that fall inside the domain and do not have distance less than $h/4$ from one of the boundary points.

The regular grid in the final step can be replaced by any assignment of data points that avoid coincidence. Everything will work the same way for scattered data.

This setup was first used for large $h = h_0$ to get a starting collocation which is solved by a linear equation system. To ensure enough points on the boundary, the distribution of points was such that for $h_0 = 0.5$ and the full domain $[0,1] \times [0,1]$ one gets the standard regular 3×3 configuration, where 8 points are for Dirichlet data on the boundary and all 9 points are for the Laplacan data. All the rest is done by standard scaling.

After the starting approximation is computed, a much smaller h_1 is used in our examples to generate the actual data. We used Wendlands C^6 function $\phi(r) = (1-r)_+^8(1 + 3r + 18r^2 + 35r^3)$ and the values

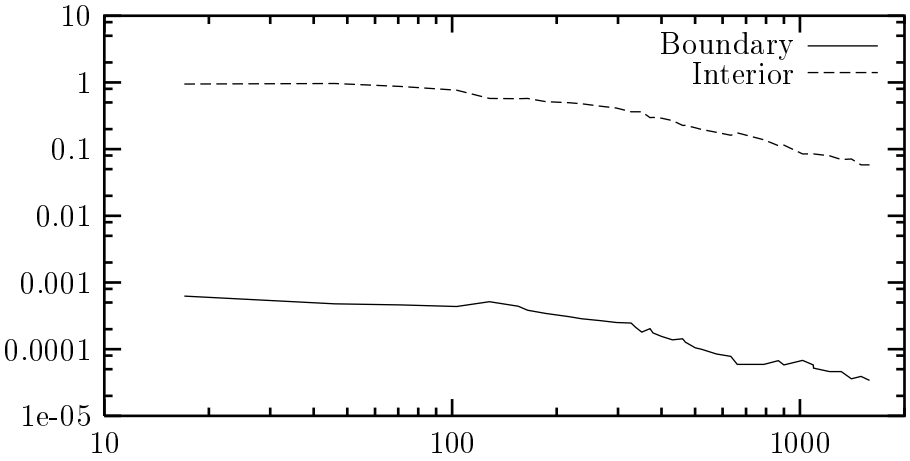
$$\begin{array}{rcl} h_1 & = & 1/128 \\ \alpha & = & 0.95 \\ \beta & = & 2^{-1/6} \\ \rho & = & 2 \\ M & = & 13 \end{array}$$

and set $K_{new} = \rho^{k-1} K_{old}$ if success within $\rho^k K_{old}$ steps occurred. This yields up to 16384 interior data points, but we want to get away with fewer nonzero coefficients. In addition, we performed a restart with the starting value of c when the residuals were reduced to 25%. This helped to prevent the method from decreasing c too quickly.

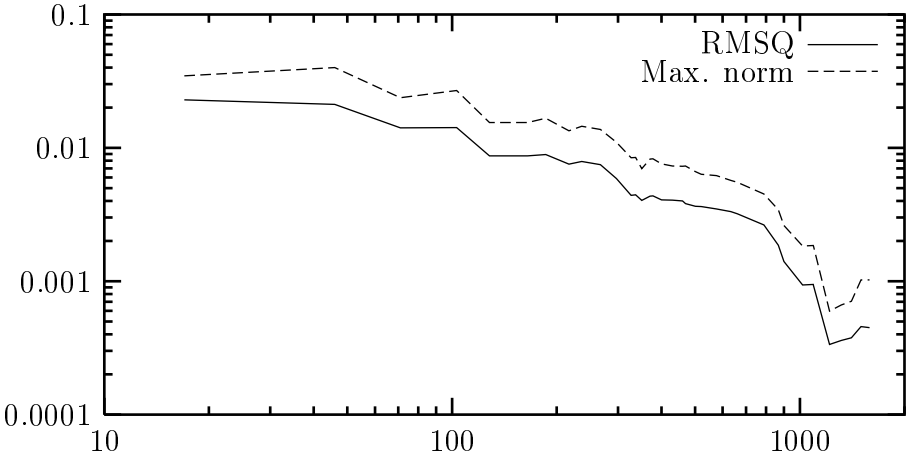
Note that the method does not directly aim at a solution of the PDE problem. Instead, it is an iterative method to calculate an approximation to the solution of a very large collocation problem, and one cannot expect that the quality of approximation to the PDE solution is better that the reproduction quality for the solution of the linear system. We thus cannot hope that the root–mean–square error with respect to the solution of the PDE tends to zero, especially if the latter is not calculated on our data locations. According to Theorem 1, the adequate way to provide supporting evidence for the method is to demonstrate the decay of the maximum norms $\|f^k\|_\infty$ of the discrete residuals during the steps of the method.

These come in two classes, Dirichlet boundary residuals and Laplace residuals inside the domain. Very often these residuals differ in orders of magnitude. We calculate their relative magnitude after the startup step and use this to introduce a weighted maximum norm which usually puts a larger weight on the quality of the boundary values.

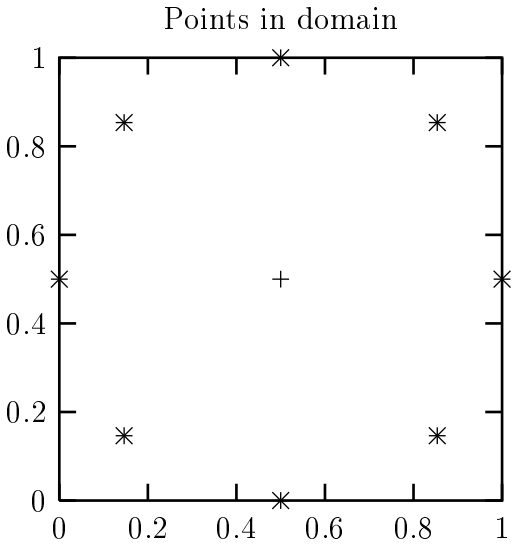
The following simple example was done on the circle with center $(0.5, 0.5)$ and radius 0.5, using $g = (x - 0.5)^2 + (y - 0.5)^2 + 1$, which also is the exact solution, and $f = 4$ with a weight of 1520 in favour of the boundary data. The following plots show the behaviour of the adaptive method.



Maximum residual against number of centers



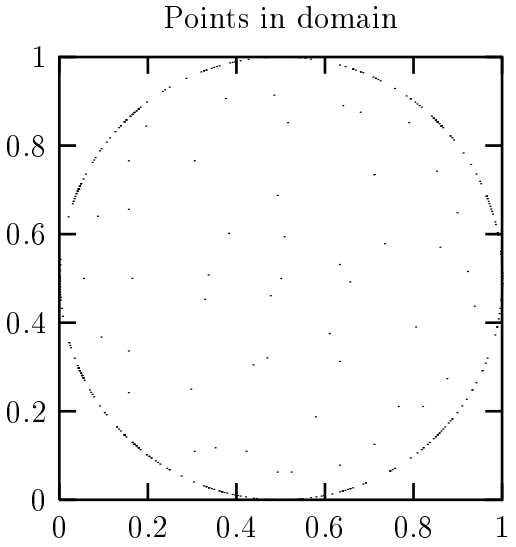
Absolute error against number of centers



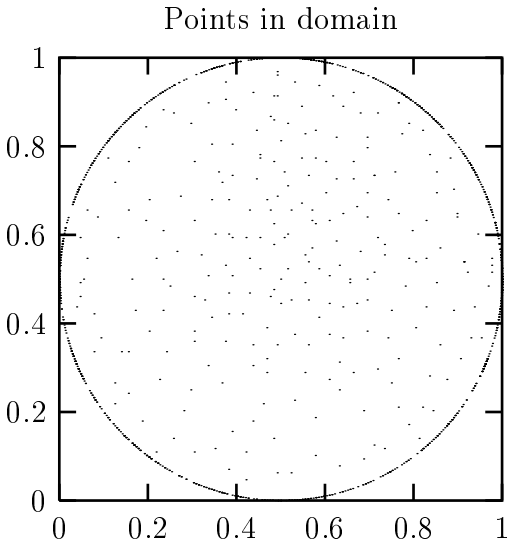
Starting approximation with 8+9 points

$*$ boundary point, also counted as "interior" point

$+$ truly interior point

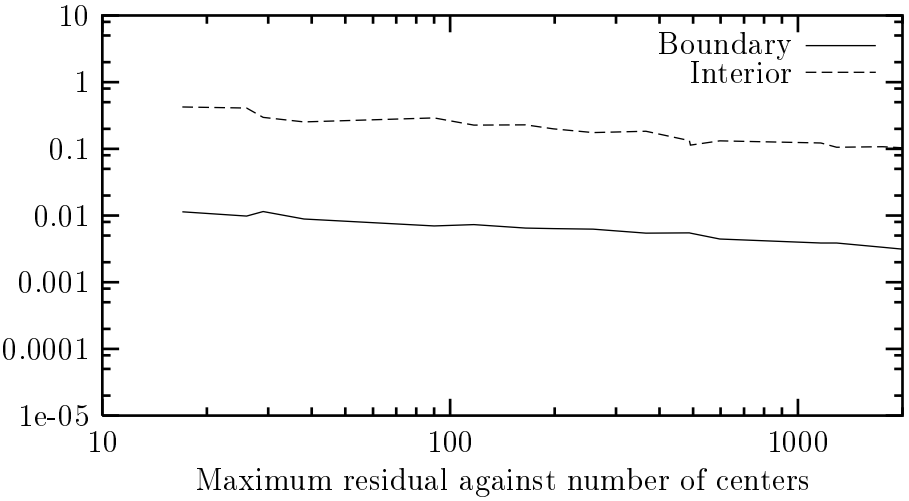


First 460 accepted centers

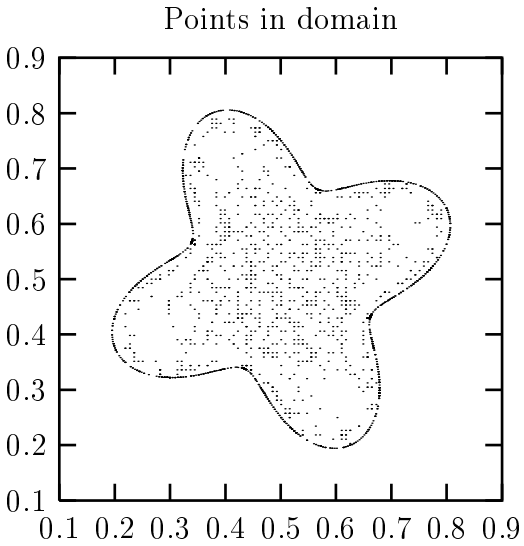


Final 1586 accepted centers

We present another example. Here, $f = g = 1$ on a smooth but irregular domain with an unknown solution to the Poisson problem. The residuals behaved like



and the final 2649 centers were



6 Conclusions

In this preliminary form, the proposed adaptive method is not considered as a serious challenge to existing methods. Its convergence behaviour needs improvement by preconditioning techniques, and the control of the various parameters deserves further study. However, our goal was to present a new idea for attacking large–scale problems with nonstandard techniques.

References

[1] DeVore, R. A., and V. N. Temlyakov, Some remarks on greedy algorithms, Advances in Computational Mathematics **5** (1996), 173-187.

[2] Fasshauer, G., Solving partial differential equations by collocation with radial basis functions, in:*Surface Fitting and Multiresolution Methods*, A. LeMéhauté, C. Rabut, and L.L Schumaker, eds., Vanderbilt University Press, Nashville, 1997, 131-138.

[3] Faul, A., and M.J.D. Powell, Proof of convergence of an iterative technique for thin–plate spline interpolation in two dimensions, Preprint DAMTP 1998/NA08

[4] Franke, C. and R. Schaback, Convergence Order Estimates of Meshless Collocation Methods using Radial Basis Functions, Advances in Computational Mathematics, Volume 8, Issue 4, June 1998, pp. 381-399

[5] Franke, C. and R. Schaback, Solving Partial Differential Equations by Collocation using Radial Basis Functions, Applied Mathematics and Computation, Volume 93, Issue 1, June 1998, pp. 73-82

[6] Hon, Y. C., multiquadric collocation method with adaptive technique for problems with boundary layer, Int. J. Appl. Sci. Comput., Volume 6, Issue 3, 1999, 173-184.

[7] Hon, Y. C., K. F. Cheung, X. Z. Mao, and E. J. Kansa, A multiquadric solution for the shallow water equations, ASCE J. Hydraulic Engineering, Volume 125, Issue 5, 1999, 524-533.

[8] Hon, Y. C. and X. Z. Mao, An efficient numerical scheme for Burgers' equations, Appl. Math. Comput., **95**, (1998), 37-50.

[9] Hon, Y. C. and Z. Wu, A quasi-interpolation method for solving stiff ordinary differential equations, Int. J. Numer. Meth. Engng., Volume 48, 2000, 1187-1197.

[10] Kansa, E. J., Multiquadrics-a scattered data approximation scheme with applications to computational fluid dynamics-II. Solutions to hyperbolic, parabolic, and elliptic partial differential equations, Comput. Math. Applic. **19** (1990), 147-161.

[11] Milroy, M.F., G.W. Vichers, and C. Bradley, An adaptive radial basis function approach to modeling scattered data, J. Appl. Sci. and Comput. **1** (1994), 319-349.

[12] Schaback, R. , Native Spaces of Radial Basis Functions I, International Series of Numerical Mathematics 132 (1999), Birkhäuser Verlag, 255-282

[13] Schaback, R., and H. Wendland, Adaptive Greedy Techniques for Approximate Solution of Large RBF Systems, preprint 2000

[14] Temlyakov, V. N. The best m-term approximation and greedy algorithms, Advances in Computational Mathematics **8** (1998) 249-265.