Adaptive Numerical Solution of MFS Systems

Robert Schaback

February 8, 2008

Dedicated to Professor Graeme Fairweather
on the occasion of his 65th birthday

Abstract

The linear systems arising from MFS calculations share certain numerical effects with other systems involving radial basis functions. These effects concern approximation error and stability, which are closely related, and they can already be studied for simple interpolation problems without PDEs. In MFS calculations, they crucially depend on the position and density of the source points and the collocation points. In turn, the choice of these points must depend on the smoothness and possible singularities of the solution. This contribution provides an adaptive method which chooses good source points automatically. A series of examples shows that the adaptive choice of source points follows the theoretical predictions quite well.

1 Introduction

The Method of Fundamental Solutions (MFS) solves a homogeneous boundary value problem via approximation of the boundary data by traces of fundamental solutions centered at *source points* outside the domain in question. The method has been used extensively in recent years, and there are excellent surveys [6, 8, 5]. However, this contribution focuses on the linear systems arising in MFS calculations and ignores applications in engineering and science. Since our observations will easily generalize to other cases, we keep the presentation and the examples simple by restricting ourselves to the homogeneous Poisson problem

$$\begin{array}{rcll} \Delta u & = & 0 & \text{in } \Omega \subset I\!R^2 \\ u & = & \varphi & \text{on } \Gamma := \partial\Omega \end{array} \tag{1}$$

with the Laplace operator. In this case, the fundamental solution (up to a multiplicative constant) is the singular radial kernel function

$$\Phi(x,y) := \log \|x-y\|_2^2,\ x,y \in I\!R^2.$$

The source points will be taken from a curve Σ outside $\overline{\Omega}$ which is often called the "fictitious" boundary. In particular, users normally choose N points $y_1,\ldots,y_N \in \Sigma$ and take linear combinations

$$s(x) := \sum_{j=1}^{N} \alpha_j \log \|x-y_j\|_2^2,\ x \in \Omega \tag{2}$$

of fundamental solutions as *trial functions* being homogeneous solutions of the Laplacian, i.e. harmonic functions. Of course, other homogeneous solutions can also enrich the trial space, and there are plenty of such possibilities, including harmonic polynomials. Methods like this date back to Trefftz [16] in much more general form, and are currently revived under the name of *boundary knot methods* [4].

2 Error Bounds

Whatever homogeneous solutions the trial functions s are composed of, the *maximum principle* will under mild assumptions on the regularity of the domain and the boundary data [11] imply that the true solution u and the trial approximation s satisfy the error bound

$$\|u-s\|_{\infty,\overline{\Omega}} \le \|u-s\|_{\infty,\partial\Omega}.$$

This means that users only have to worry about the L_∞ approximation error on the boundary. If a fixed space of general linear combinations

$$s(x) := \sum_{j=1}^{N} \alpha_j s_j(x),\ x \in \Omega$$

of smooth homogeneous solutions s_j are admitted, the natural numerical approach induced by the Maximum Principle would be to minimize the L_∞ norm of the error on the boundary. This is a semi–infinite linear optimization problem

$$\begin{array}{rcl} & \text{Minimize } \eta & \\ -\eta & \le \varphi(x) - \displaystyle\sum_{j=1}^{N} \alpha_j s_j(x) & \le \eta, \quad x \in \Gamma \end{array} \tag{3}$$

with $N+1$ variables $\eta,\alpha_1,\ldots,\alpha_n$ and infinitely many affine–linear constraints. The literature on optimization deals with such problems [10, 9], but in many cases it suffices to come up with a cheap but suboptimal approximation. We shall focus on this situation and give examples later.

3 Linear Systems

In particular, users often try to get away with picking N *collocation points* $x_1, \ldots, x_N$ on the boundary Γ and setting up an $N \times N$ linear system

$$\sum_{j=1}^{N} \alpha_j s_j(x_k) = \varphi(x_k),\ 1 \le k \le N \tag{4}$$

for interpolation at these points. This works well in many cases, but the main theoretical problem with such systems is that the coefficient matrix with entries $s_j(x_k)$ may be singular. This clearly occurs for $N > 1$ and the MFS, because the determinant of the $N \times N$ system with matrix entries

$$s_j(x_k) = \log \|y_j - x_k\|_2^2,\ 1 \le j, k \le N$$

will be a smooth function of the source points y_j, and swapping two source points will change the sign of the determinant. Thus there are plenty of configurations of source and test points where the system is necessarily singular. Confining source points to curves may help in 2D cases, but not in 3D if source points are restricted to surfaces.

Consequently, it does not make any sense to head for theorems proving nonsingularity of the above systems. The same holds for other unsymmetric collocation–type techniques like the one introduced by E. Kansa [12, 13] for general PDE problems in strong form, or the meshless local Petrov–Galerkin method of S.N. Atluri and collaborators [1, 2].

Instead, systems like (4) should not be expected to be solvable exactly. In view of the maximum principle and the semi–infinite optimization problem (3) one can take many more collocation points than source points and solve the overdetermined linear system

$$\sum_{j=1}^{N} \alpha_j s_j(x_k) = \varphi(x_k),\ 1 \le k \le M \ge N \tag{5}$$

approximatively, e.g. by a standard least–squares solver. We shall focus on such systems from now on.

4 Choice of Test and Collocation Points

If a good linear combination s of the form (2) is found by any method whatsoever, users will check the maximum boundary error $\|\varphi - s\|_{\infty,\Gamma}$ by evaluating the error in sufficiently many *test points* on the boundary. Though this test also needs a thorough mathematical analysis in order to be safe, we ignore it here. We just remark that users will need very

many test points in case of steep gradients of the trial functions, and this inevitably occurs if the MFS is used with source points close to the boundary. Adding more test points still is computationally cheap if N is not too large, and most users will be satisfied with a simple plot of the boundary errors evaluated at test points guaranteeing graphic accuracy, i.e. at most 1000 points per plot. We shall use this rule–of–thumb in later examples.

Choosing M *collocation* points for setting up the system (4) is somewhat more difficult, but it will always stabilize the system if more points are taken. Independent of the choice of trial functions, users can repeat the calculation with more or other collocation points, if they are not satisfied with the first result. This is a simple way of introducing *adaptivity* into the numerical solution strategy:

> *Adaptivity of Testing*:
> If the evaluation of the boundary error on certain test points yields values that are intolerably large, take these test points as collocation points and repeat the calculation.

As long as the trial space S is not changed, this can improve the results, but if the trial space is poorly chosen, the final boundary error cannot be less than

$$\inf_{s\in S} \|\varphi - s\|_{\infty,\Gamma} \tag{6}$$

no matter how collocation and testing is done and how many points are used.

But there is another argument that needs consideration. If the linear optimization problem (3) is solved for a large but finite subset Γ_0 of the boundary instead of the full boundary, the Karush–Kuhn–Tucker conditions applied to the dual reformulation [3] of a linear minimax problem will imply that there is a subset Γ_1 of Γ_0 consisting of at most $N+1$ points such that

$$\inf_{s\in S} \|\varphi - s\|_{\infty,\Gamma_0} = \inf_{s\in S} \|\varphi - s\|_{\infty,\Gamma_1}.$$

This is related to the notion of *support vectors* in support vector machines, and it has the following implication:

> *Reducibility of collocation points*:
> If a system (4) with $M >> N$ has a good approximate solution, it even has a good approximate solution determined already by a subset of at most $N+1$ collocation points.

Unfortunately, these collocation points are not known beforehand, but users should be aware of the fact that a large system with a good approximate solution will have a much smaller subsystem with an equally good solution. This fact will reappear later, and we shall provide examples.

5 Choice of Trial Space

The lower bound (6) for the achievable boundary error reveals that the main design problem consists in picking good trial functions, or, in case of the MFS, in picking good source points.

Let us postpone the MFS for a while. Users can take all homogeneous solutions as trial functions, and this will work well in certain examples we shall look at later. For the Laplace operator in 2D, the real part of any differentiable function of a complex variable will be harmonic and can serve as a possible trial function. The standard fundamental solution just is a special case of a real part of a complex function with a singularity, but there are many others without singularities, e.g. harmonic polynomials or entire functions like $f(x, y) := \exp(y)\cos(x)$.

How to choose? We shall later let an algorithm decide adaptively, but there is a general though trivial rule:

> Take trial functions with similar analytic properties as the expected solution. In particular, be careful when the solution or one of its derivatives will necessarily have singularities somewhere.

6 Harmonic Polynomials

We explain this first for the case of using harmonic polynomials. If the solution u of the given Poisson problem is itself a real part of a function of a complex variable without singularities anywhere, it can be well approximated by harmonic polynomials on any curve, namely by the real part of its partial sums of its power series. The shape of the domain does not matter at all, and the background PDE problem is completely irrelevant because we only have to recover a partial power series. A full power series of an analytic function is determined by values on any countable set with an accumulation point, and thus recovery of globally harmonic functions from point evaluation data will work almost anywhere.

By analogy to certain theorems on polynomial approximations to analytic functions [7], the rate of approximation can be expected to be spectral, i.e. the error should behave like $C\lambda^n \to \infty$ as a function of the degree n of the harmonic polynomials used, and $\lambda > 0$ can be arbitrarily small. Then the choice of harmonic polynomials should be superior to all choices of fundamental solutions. In many engineering applications where MFS users report that source points of the MFS taken far away from the domain work best, the special examples usually have solutions without singularities anywhere, but users tend to ignore that harmonic polynomials will do even better in such situations.

If the solution, when viewed as a global function, is still harmonic but has a singularity at a positive distance to the boundary Γ, the rate

of approximation will again be like $C\lambda^n \to \infty$, but with $\lambda < 1$ now being bounded below, and related to the distance of the singularity to the boundary, with $\lambda \to 1$ if the singularity moves towards the boundary. Again, the shape and smoothness of the boundary is irrelevant. The crucial quantity is the distance of the closest singularity of the solution from the boundary, when the solution is extended harmonically as far as possible. Again, this case is hard to beat by the MFS, if the singularity is sufficiently far away from the domain.

The situation gets serious if the solution or one of its derivatives has a singularity directly on the boundary Γ. Note that this case occurs whenever the boundary data, however smooth, are given by a function which is not itself harmonic. In such a case, the rate of approximation of boundary values by harmonic polynomials can be very poor, depending on the smoothness of the solution when restricted to the boundary. The upshot of this discussion of harmonic polynomials is that the MFS makes sense only if the boundary data come from a non–harmonic function or if there is no harmonic extension of the solution without singularities close to the boundary. Users working in application areas do not seem to be aware of this fact.

7 Rescaling Fundamental Solutions

Before we go over to the problem of choosing good source points for the MFS, let us consider the case of far–away source points $y \in I\!R^2$ while the evaluation of a fundamental solution $\log \|x-y\|_2^2$ is at $x \in I\!R^2$ with a relatively small value of $\|x\|_2$. In such cases, the functions $\log \|x-y\|_2^2$ will not differ much if y varies, and consequently the resulting matrix gets a bad condition. But we can rewrite the function for large $\|y\|_2 \neq 0$ as

$$\begin{aligned}
\log \|x-y\|_2^2 &= \log \left(\|x\|_2^2 - 2(x,y)_2 + \|y\|_2^2 \right) \\
&= \log \left(\|y\|_2^2 \left(\frac{\|x\|_2^2}{\|y\|_2^2} - 2 \left(x, \frac{y}{\|y\|_2^2} \right)_2 + 1 \right) \right) \\
&= \log \|y\|_2^2 + \log \left(1 + \left(\frac{\|x\|_2^2}{\|y\|_2^2} - 2 \left(x, \frac{y}{\|y\|_2^2} \right)_2 \right) \right).
\end{aligned}$$

If we use the expansion

$$\log(1+z) = \sum_{j=1}^{\infty} (-1)^{j-1} \frac{z^j}{j}$$

for $|z| < 1$, we get for sufficiently large $\|y\|_2$ the expansion

$$\begin{aligned}
& \log\|x-y\|_2^2 - \log\|y\|_2^2 \\
= \;& \log\left(1 + \left(\frac{\|x\|_2^2}{\|y\|_2^2} - 2\left(x, \frac{y}{\|y\|_2^2}\right)_2\right)\right) \\
= \;& \sum_{j=1}^{\infty} \frac{(-1)^{j-1}}{j} \left(\frac{\|x\|_2^2}{\|y\|_2^2} - 2\left(x, \frac{y}{\|y\|_2^2}\right)_2\right)^j \\
= \;& \sum_{j=1}^{\infty} \frac{(-1)^{j-1}}{j} \sum_{m=0}^{j} \binom{j}{m} \left(\frac{\|x\|_2^2}{\|y\|_2^2}\right)^{j-m} \left(-2\left(x, \frac{y}{\|y\|_2^2}\right)_2\right)^m \\
= \;& \sum_{j=1}^{\infty} \frac{(-1)^{j-1}}{j} \sum_{m=0}^{j} \binom{j}{m} \frac{1}{\|y\|_2^{2j-m}} \|x\|_2^{2j-2m} \left(-2\left(x, \frac{y}{\|y\|_2}\right)_2\right)^m \\
= \;& \sum_{k=1}^{\infty} \frac{1}{\|y\|_2^k} \sum_{k/2 \le j \le k} \frac{(-1)^{j-1}}{j} \Bigg[\\
& \sum_{m=0}^{j} \binom{j}{2j-k} \|x\|_2^{2k-2j} \left(-2\left(x, \frac{y}{\|y\|_2}\right)_2\right)^{2j-k} \Bigg] \\
=: \;& \sum_{k=1}^{\infty} \frac{1}{\|y\|_2^k} p_k(x,y)
\end{aligned}$$

of the fundamental solution at y into harmonic polynomials

$$p_k(x,y) := \sum_{k/2 \le j \le k} \frac{(-1)^{j-1}}{j} \sum_{m=0}^{j} \binom{j}{2j-k} \|x\|_2^{2k-2j} \left(-2\left(x, \frac{y}{\|y\|_2}\right)_2\right)^{2j-k}$$

with respect to x of degree k. If we push the source point y to infinity by writing it as $y = rz$ for large $r > 0$ and fixed $z \in I\!R^2$ with $\|z\|_2 = 1$, we get

$$\log\|x - rz\|_2^2 \;=\; 2\log r + \sum_{k=1}^{\infty} \frac{1}{r^k} p_k(x,z)$$

and this is something like a "far field expansion" of the fundamental solution. Note that z and r are considered to be fixed, and thus users are strongly advised to include constants into the space of trial functions in order to cope with the $2\log r$ term.

Now let us look at the span of fundamental solutions based on points $y_j = rz_j$ on a circle of radius r for large r. We want to find functions which are in the span when taking the limit $r \to \infty$, and we call this the "asymptotic span". The linear combinations are

$$\begin{aligned}
s_r(x) \;&=\; \sum_{j=1}^{N} \alpha_j(r) \left(2\log r + \sum_{k=1}^{\infty} \frac{1}{r^k} p_k(x, z_j)\right) \\
&=\; 2\log r \sum_{j=1}^{N} \alpha_j(r) + \sum_{k=1}^{\infty} \frac{1}{r^k} \sum_{j=1}^{N} \alpha_j(r) p_k(x, z_j)
\end{aligned}$$

and thus have specific expansions in terms of harmonic polynomials. If constants are not added to the span, and if the MFS works at all for large r in a specific case, the sum of the coefficients $\alpha_j(r)$ will tend to zero for $r \to \infty$ while the coefficients themselves cannot stay all bounded. In all "pure MFS" examples with far-away source points, the sum of coefficients will always be close to zero while the sum of the absolute values will be extremely large.

To avoid computational crimes, we now add the constant 1 to the span of trial functions and use a coefficient α_0 for it. Then we have a span of

$$s_r(x) = 1 \left(\alpha_0(r) + 2 \log r \sum_{j=1}^{N} \alpha_j(r) \right) + \sum_{k=1}^{\infty} \frac{1}{r^k} \sum_{j=1}^{N} \alpha_j(r) p_k(x, z_j)$$

which we can analyze somewhat easier. We have the constants in the span, of course, but for arbitrary $\alpha_1(r), \ldots, \alpha_N(r)$ we can always set

$$\alpha_0(r) := -2 \log r \sum_{j=1}^{N} \alpha_j(r)$$

to cancel the first term. Now $r s_r(x)$ must be in the span, and this asymptotically is in the span of the $p_1(x, z_j)$, $1 \le j \le N$, which necessarily is a subspace V_1 of the linear polynomials. To proceed inductively, we now look at the subspace A_1 of coefficient vectors $\alpha \in I\!R^N$ with

$$\sum_{j=1}^{N} \alpha_j p_1(x, z_j) = 0.$$

If we take a vector $\alpha \in A_1$ and form the functions $r^2 s_r(x)$, we find that the aymptotic span of the fundamental solutions contains the polynomial space

$$V_2 := \left\{ \sum_{j=1}^{N} \alpha_j p_2(x, z_j) \ : \ \alpha \in A_1 \right\}$$

of maximally second–degree polynomials. Inductively we can define $A_0 := I\!R^N$ and

$$A_m := \left\{ \alpha \in I\!R^N \ : \ \sum_{j=1}^{N} \alpha_j p_i(x, z_j) = 0, \ 1 \le i \le m \right\}$$

for all $m \ge 1$ and use it for defining a space

$$V_m := \left\{ \sum_{j=1}^{N} \alpha_j p_m(x, z_j) \ : \ \alpha \in A_{m-1} \right\}$$

of polynomials of degree at most m. The spaces A_m form an inclusion chain

$$R^N = A_0 \supseteq A_1 \supseteq A_2 \supseteq \cdots$$

and if we take an appropriate orthogonal basis for that chain, we get

Theorem 1 *The asymptotic span for $r \to \infty$ of fundamental solutions with source points of the form $y_j = rz_j$ for fixed points z_j on the unit circle is a space of harmonic polynomials spanned by constants and the union of all V_m.* □

Unfortunately, it seems to be difficult to calculate the dimension of that space, because it will depend on the number and the geometry of the points z_j.

The upshot of all of this is that the MFS for far–away source points, if it works at all, is asymptotically nothing else than a fit of the boundary data by specific harmonic polynomials. Thus the MFS should not be used at all for far–away source points, but rather be replaced by use of harmonic polynomials. For this reason, we do not elaborate the above argument any further, though it would result in a way of preconditioning MFS matrices for far–away points. It does not make sense to precondition a matrix one should not use.

However, a rather primitive but still somewhat useful change of basis induced by the above argument is to add constants to the MFS span and replace the fundamental solution at $y \neq 0$ by

$$\left(\log \|x-y\|_2^2 - \log \|y\|_2^2\right) \|y\|_2$$

behaving like a linear polynomial in x when y is far away from x. A full preconditioning will use such basis changes plus coefficient vectors from an orthogonal basis of $\mathbb{R}^N$ which is compatible with the chain of the A_m spaces. Details can be worked out similarly to [19]. As an aside, we remark that it is no problem to replace the standard fundamental solutions by rational trial functions arising when taking derivatives of $\log \|x - ry\|_2^2$ with respect to r.

Finally, we present an example supporting the results of this and the previous section. In Figure 1 we show the L_∞ error $\epsilon_\infty(r)$ on the full circle when we recover the harmonic function $f(x, y) = e^x \cos(y)$ from boundary values only on a half circle. We collocate at 100 test points on the right half unit circle, using 20 source points on the right half circle of radius r. We stopped the calculation when the numerical rank of the 100×20 collocation matrix, as given by MATLAB©, was less than 20, and this occurred for $r \approx 5.5$ already. The error decreases nicely with increasing r, because the setting converges towards harmonic polynomials for $r \to \infty$, as was shown in this section, and since the discussion in the previous section showed that recovery by harmonic polynomials should work on any arc.

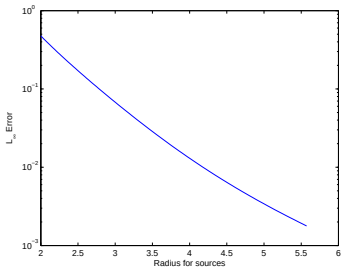

Figure 1: L_∞ error as function of r

8 Choice of Source Points

It should be clear by now that a good placement of source points will crucially depend on the distance of the closest singularity arising when extending the solution harmonically outside the domain. In many cases, the user normally does not have this information, but there are a few guidelines.

We start by an upside–down argument. If the MFS works for sufficiently many source points on a fixed curve Σ, and if the results are getting better when taking more source points, the solution will have a harmonic extension up to Σ, because the MFS constructs it. But if there necessarily is a singularity inside Σ for some reason or other, the MFS cannot work satisfactorily on Σ.

We now have to find a–priori indicators for singularities close to the boundary. The first and simplest case arises when the known boundary data are such that there is no C^∞ extension locally into R^2. This always happens if the boundary data are not C^∞ on smooth parts of the boundary. If users know where the "boundary points of data nonsmoothness" are, source points should be placed close to those. Unfortunately, there currently is no general way to guess the type of singularity beforehand, even if the position is known. Thus this case usually must be handled experimentally.

A second and partially independent case arises for incoming corners of the domain. Even if the boundary data have a C^∞ extension to $\mathbb{R}^2$, e.g. if they are non–harmonic polynomials, users must expect a singularity at the boundary, but the type of singularity is known, depending on the boundary angle. Again, users should either add the correct type of singularity or place source points close to corners in such cases. But the situation is different if the data come from an extendable harmonic function, even if corners are present. Then the MFS can ignore the corners. Note that MFS examples on domains with corners are useless as long as they consider specific boundary

data which are values of functions with a harmonic extension.

Finally, the convergence rate of the MFS when adding more and more source points will be strongly influenced by the smoothness of both the data function and the MFS trial functions on the boundary. Approximation theory proves in many situations that convergence rates are completely controlled by the minimal smoothness of the data function and the trial functions. Thus smooth boundary data on smooth boundaries will lead to good convergence rates improving with the smoothness properties. If source points can be kept at a fixed positive minimal distance from the boundary (this requires the solution to have a harmonic extension), then the trial functions are analytic and the convergence rate will be completely determined by the smoothness of the data on the boundary. But then the approximation by harmonic polynomials on the boundary will also have a good convergence rate depending on the smoothness of the boundary data, and it is not easy to predict superiority of the MFS over approximation by harmonic polynomials.

If singularities of derivatives are on the boundary or if there are incoming corners, the convergence rate of approximation of the boundary data by harmonic polynomials will deteriorate seriously, and the MFS can be competitive by placing source points closer and closer to the singularities. A general rule is not known, but there are certain adaptive techniques [18, 14, 15, 6] to handle this case. We shall provide a simple adaptive method in the next section.

9 Greedy Adaptive Techniques

Overdetermined systems like (5) can be approximately solved by a step–wise adaptive techniques even if they are huge. We applied the method of [17] to MFS problems, but it turned out to be less stable than the algorithm we describe now, because the previous one did not keep all collocation points under control.

The basic idea can be formulated independent of the MFS in terms of solving a linear unsymmetric over– or underdetermined $m \times n$ system of the form $Ax = b$. The goal is to pick useful columns of the $m \times n$ matrix A in a data–dependent way without cutting the number of rows down. This is also done by any reasonable solution algorithm, e.g. by the backslash operator in MATLAB©, and we shall present examples later. However, standard QR routines do not account for the right–hand side b, and they do not stop early when only a few columns of the matrix suffice to reproduce the right–hand side with small error. To maintain stability, we use orthogonal transformations like in any QR decomposition, but we make the choice of columns dependent on the right–hand side.

In short, our adaptive algorithm for selecting good columns works as follows:

1. Pick the column of A whose multiples approximate b best.
2. Then transform the problem to the space orthogonal to that column and repeat.

If the algorithm has selected a number of columns this way, take this column selection for a trial space and use your algorithm of choice for solving the given problem on that trial space. For instance, in MFS applications one can use L_∞ minimization of boundary errors after the selection process has provided a small set of useful source points.

The actual implementation of the algorithm needs some further explanation. Approximation of b by multiples of a single nonzero vector a is optimal in L_2, if the error vector has the form $b - a \cdot \frac{b^T a}{a^T a}$, and its squared norm then takes the minimal possible value

$$\|b\|_2^2 - \frac{(b^T a)^2}{\|a\|_2^2}.$$

If we denote the columns of A by $a^1, \ldots, a^n$, we thus can implement Step 1 by taking the maximum of

$$\frac{(b^T a^j)^2}{\|a^j\|_2^2},\ 1 \le j \le n,\ \|a^j\|_2 \neq 0$$

to pick the best column for approximation of b. If we denote this column by u, we form the normalized vector $v := u/\|u\|_2$ and transform both A and b into

$$\begin{array}{rcl} A_1 & := & A - v(v^T A) \\ b_1 & := & b - v(v^T b) \end{array}$$

to let both b_1 and the columns of A_1 be orthogonal to u and v. The new matrix has a zero column where once was u. To avoid roundoff problems, we insert exact zeros there, but we do not delete the zero column in order to avoid unnecessary storage transformations. Instead, we store the column index of u for later use and proceed. The following steps will always automatically ignore the columns we already picked. Note that we do not care for the approximate solution of the system and about accumulation of roundoff during the transformations. The L_2 norms of the vectors $b, b_1, \ldots$ will necessarily decrease, and this can be used for stopping the algorithm. Going for a strong error reduction will finally use the full matrix, while users can get away with just a few columns and a very simple numerical solution if they admit larger errors. A MATLAB© package is available from the author.

Application of this technique to MFS calculations is particularly appealing in cases where the user lets the algorithm decide which trial

function to choose. One can offer harmonic polynomials up to a fixed degree and plenty of fundamental solutions at different distances to the boundary, and the algorithm will pick suitable ones without knowing background mathematics like harmonic extendibility of the solution. Running the algorithm several times will provide the user with information about hazardous places at the boundary, and the user can offer refined choices of source points close to these when preparing the next run. The actual calculation of the solution is done after column selection in order to keep the accumulated errors small.

10 Examples

To illustrate the mathematical issues of the previous sections, we now provide a series of examples, but we have to explain the notation in the tables and figures first. Following (4), the number of collocation points will be M, and N will denote the number of source points offered to the algorithm. If only a smaller subset of source points is actually used for the calculation, we use $n \leq N$. Similarly, K denotes the number of harmonic polynomials included into the trial space, and k stands for the number actually arising in the solution. For approximate evaluation of the L_∞ norm ϵ_∞ of the error on the boundary we use $\max(1000, 5 * N)$ points for graphic accuracy. The number M of collocation points is always defined as $1 + \max(200, 2N + 2K)$. The approximate solution of the system will be done by algorithms labeled as

$L2$: the MATLAB© backslash operator, i.e. a standard least–squares solver with internal selection of columns,

$A2$: the same as above, but applied to the reduced matrix after adaptive column selection along the lines of the previous section,

$A\infty$: an L_∞ linear optimizer applied to the same adaptively reduced matrix.

Note that the solution algorithm will have quite some influence on the final error and the number of nonzero solution coefficients. The $L2$ solver always exploits maximal machine accuracy, while our adaptive solvers $A2$ and $A\infty$ are tuned for a compromise between error and complexity.

We denote the L_∞ error on our test points as ϵ_∞. It is debatable to use the RMSE error measure at all, because the Maximum Principle is guiding the error behavior, but we also include it as ϵ_2. The curves we use for boundaries or source points consist of circles C_r with radius r around zero, or they are obtained from a polar coordinate representation of the standard boundary B_0 by adding a distance h to get a source boundary B_h at polar distance h.

The first example in Table 1 concerns a more or less trivial case where the data come from a globally harmonic function $f(x, y) = \exp(x)\cos(y)$ which is the real part of the entire complex function $\exp(z) = \exp(x + iy)$. The real part of the power series of $\exp(z)$ yields a sequence of perfect approximations by harmonic polynomials on each domain whatsoever, and the recovery by collocation via harmonic polynomials even works on open arcs anywhere. As expected, the MFS cannot outperform harmonic polynomials, independent of where the sources are. The domain is a lemniscate with an incoming corner as in Figure 2, while the source points are on the circle of radius 4 around the origin.

N	M	K	Alg	n	m	k	ϵ_∞	ϵ_2
0	201	25	$L2$	0	201	13	1.16e-011	7.53e-012
0	201	25	$A2$	0	201	12	1.82e-010	1.18e-010
0	201	25	$A\infty$	0	201	12	1.81e-010	1.18e-010
200	401	0	$L2$	36	401	0	1.54e-013	6.74e-014
200	401	0	$A2$	25	401	0	3.54e-009	1.90e-009
200	401	0	$A\infty$	28	401	0	5.88e-007	2.04e-007
200	451	25	$L2$	7	451	22	5.18e-010	1.79e-010
200	451	25	$A2$	1	451	11	5.79e-010	3.73e-010
200	451	25	$A\infty$	1	451	11	7.49e-010	3.89e-010

Table 1: Recovery of harmonic function

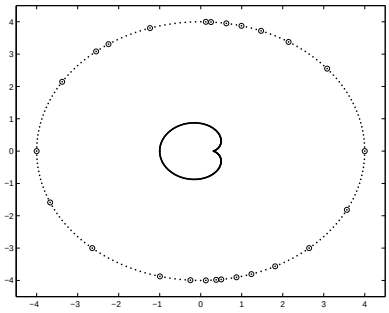

Figure 2: Lemniscate with source points

The table should be interpreted as follows. The first three lines used no source points at all ($N = 0$), but $M = 201$ collocation points and allowed $K = 25$ harmonic polynomials. Only up to $k = 13$ nonzero polynomial coefficients were calculated due to the symmetry of the data

function, and the recovery quality via the first terms of the power series is around $1.0e-10$, independent of the algorithm used. This happened for all domains tested. But the adaptive algorithms in lines 2 and 3 solve only a subproblem with 12 degrees of freedom after picking the most useful harmonic polynomials.

The next three lines are a pure MFS offering 200 source points on the circle. With the L_2 solver, the results are even better than for harmonic polynomials, and surprisingly the MATLAB backslash solver yields only 36 nonzero coefficients, i.e. only 36 source points were necessary. The adaptive solvers are satisfied with less accuracy, but also use a simpler approximation by 25 or 28 source points.

The final three lines offered the same 200 source points, but allowed also 25 harmonic polynomials. All algorithms prefer harmonic polynomials over fundamental solutions. This is to be expected, because the solution has no finite singularities. The 25 marked source points in Figure 2 belong to the situation of the fifth line of Table 1. The adaptive L_2 algorithm picks these 25 source points with no connection to the domain corner, as is to be expected. However, the error is worse than for harmonic polynomials in this case.

The L_∞ norm of the error using the $A\infty$ solver in line 3 is only slightly better than the one from the $A2$ solver in line 2. It cannot be worse because they use the same trial space. However, in some of the later cases, the $A\infty$ algorithm, after starting from the same trial space as the $A2$ algorithm, is less stable and often ends prematurely with a larger L_∞ error than the $A2$ solver. Future work should add a more sophisticated L_∞ solver.

In Table 2 we present the same situation, but with boundary data given by the function x^2y^3. This is still smooth, but the domain has an incoming corner causing problems.

N	M	K	Alg	n	m	k	L_∞	L_2
0	201	25	$L2$	0	201	12	1.17e-003	3.38e-004
0	201	25	$A2$	0	201	12	1.17e-003	3.38e-004
0	201	25	$A\infty$	0	14	12	9.28e-004	6.48e-004
200	401	0	$L2$	36	401	0	9.41e-004	2.38e-004
200	401	0	$A2$	31	401	0	1.04e-003	2.78e-004
200	401	0	$A\infty$	32	34	0	1.31e-003	8.30e-004
200	451	25	$L2$	7	451	24	1.02e-003	2.72e-004
200	451	25	$A2$	0	451	12	1.17e-003	3.38e-004
200	451	25	$A\infty$	0	26	12	9.15e-004	6.46e-004

Table 2: Smooth boundary data on lemniscate, source points on circle

With source points on the circle, none of the methods outperforms harmonic polynomials seriously.

To demonstrate that the incoming corner is the culprit, we replace the lemniscate now by the unit circle and get Table 3. Again, harmonic polynomials do best. We now go back to the lemniscate, but place the

N	M	K	Alg	n	m	k	L_∞	L_2
0	201	25	$L2$	0	201	3	1.67e-016	4.61e-017
0	201	25	$A2$	0	201	3	2.08e-016	6.84e-017
0	201	25	$A\infty$	0	201	3	1.42e-014	1.00e-014
200	401	0	$L2$	34	401	0	1.90e-008	9.64e-009
200	401	0	$A2$	33	401	0	3.31e-009	1.48e-009
200	401	0	$A\infty$	33	274	0	7.56e-006	3.38e-006
200	451	25	$L2$	17	451	11	6.47e-010	3.47e-010
200	451	25	$A2$	0	451	3	2.36e-016	9.79e-017
200	451	25	$A\infty$	0	451	3	7.31e-011	4.79e-011

Table 3: Smooth boundary data on circle

source points at a polar radial distance of 0.2 outside the boundary (i.e. we calculate them by adding 0.2 to the radius of boundary points in polar coordinates). This gives Table 4 and should be compared to Table 2. The results are not better than for sources on the circle, but

N	M	K	Alg	n	m	k	L_∞	L_2
0	201	25	$L2$	0	201	12	1.17e-003	3.38e-004
0	201	25	$A2$	0	201	12	1.17e-003	3.38e-004
0	201	25	$A\infty$	0	14	12	9.28e-004	6.48e-004
200	401	0	$L2$	200	401	0	3.09e-004	4.75e-005
200	401	0	$A2$	140	401	0	3.09e-004	4.75e-005
200	401	0	$A\infty$	140	386	0	3.66e-004	8.18e-005
200	451	25	$L2$	193	451	9	2.90e-004	4.32e-005
200	451	25	$A2$	104	451	13	3.09e-004	4.75e-005
200	451	25	$A\infty$	104	119	13	4.82e-004	1.28e-004

Table 4: Smooth boundary data on lemniscate, source points at 0.2

maybe the sources are not close enough. Thus we go for a distance of 0.02 in Table 5, and Figure 3 shows the source point distribution (circles, while the offered but unused source points are small dots) for the $A2$ technique in the eighth line of Table 5. Note that the source

points are automatically picked close to the singularity.

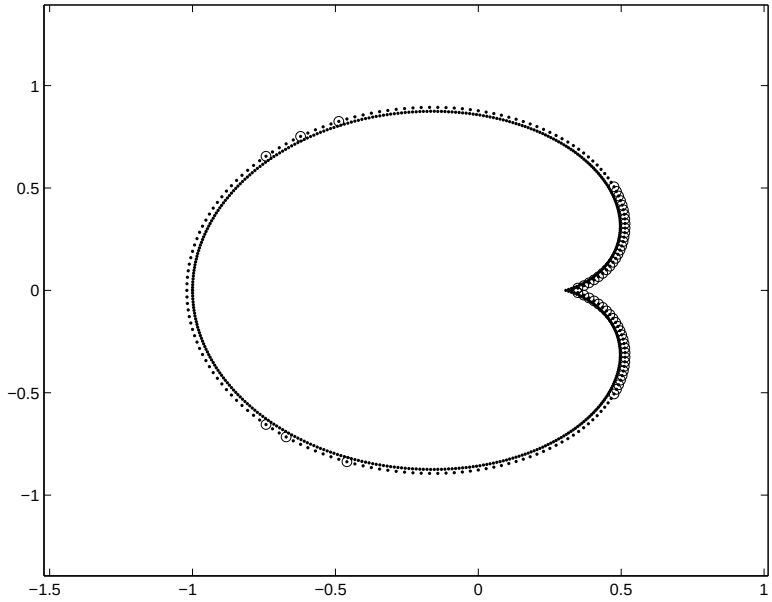

Figure 3: Lemniscate with source points

Going even closer does not pay off, unless we enhance the collocation resolution.

Now we admit source points at different distances, starting at 0.02 and ending at 10.24 after repeated multiplications by 2. At each distance, we choose random source points such that their total is roughly 201 as in the previous cases. Now the algorithms can pick source points at very different distances, and they do.

Finally, we allowed roughly 700 source points and varying distances from 0.002 to 2.048 to get Table 7. The distances were prescribed by multiplying 0.002 by powers of 2 until 2.048 was reached, while the points for fixed distance were uniformly sampled with respect to the parametrization of the boundary. Zooming in on the adaptively selected source point placements gives Figure 4 for the first line of Table 7, while Figures 5 and 6 are close-ups for line 2 with the $A2$ algorithm. Note that the small dots are the offered source points, while circles indicate the selected source points.

Users may suspect that things are better if there is no incoming corner of the domain. Thus let us take the unit circle and prescribe the continuous boundary values $\varphi(x, y) := \max(0, y)$. This leads to two derivative singularities of the harmonic solution at $(1, 0)$ and $(-1, 0)$. Numerical results are quite similar to the case on the lemniscate, and

N	M	K	Alg	n	m	k	L_∞	L_2
0	201	25	$L2$	0	201	12	1.17e-003	3.38e-004
0	201	25	$A2$	0	201	12	1.17e-003	3.38e-004
0	201	25	$A\infty$	0	14	12	9.28e-004	6.48e-004
200	401	0	$L2$	198	401	0	1.16e-004	2.28e-005
200	401	0	$A2$	196	401	0	1.16e-004	2.28e-005
200	401	0	$A\infty$	196	401	0	2.51e-004	9.65e-005
200	451	25	$L2$	198	451	12	1.17e-004	1.09e-005
200	451	25	$A2$	58	451	12	1.17e-004	1.09e-005
200	451	25	$A\infty$	58	450	12	4.32e-004	1.07e-004

Table 5: Smooth boundary data on lemniscate, source points at 0.02

N	M	K	Alg	n	m	k	L_∞	L_2
210	451	25	$L2$	111	451	21	3.64e-004	5.75e-005
210	451	25	$A2$	50	451	12	3.63e-004	5.75e-005
210	451	25	$A\infty$	50	447	12	6.12e-004	1.45e-004

Table 6: Smooth boundary data on lemniscate, about 200 source points at varying distances

thus we confine ourselves to offering about 700 source points on circles with distances 0.002 to 2.048 in Table 8. The $L2$ solver does not care about the right–hand side of the system, and thus it does not realize the two singularities. This is shown in a close–up in Figure 7, while the $A2$ algorithm (see Figure 8) selects source points close to the problematic boundary locations. Note that this case is offered 729 degrees of freedom and uses maximally 490 of these.

If we drop the MFS completely and offer 751 harmonic polynomials on the circle instead, we get Table 9. Note that this performs slightly better than the MFS and uses 377 of the possible degrees of freedom.

N	M	K	Alg	n	m	k	L_∞	L_2
704	1451	25	$L2$	416	1451	11	7.55e-005	5.31e-006
704	1451	25	$A2$	208	1451	12	6.97e-005	4.93e-006
704	1451	25	$A\infty$	211	1451	12	4.80e-003	6.97e-004

Table 7: Smooth boundary data on lemniscate, about 700 source points at varying distances

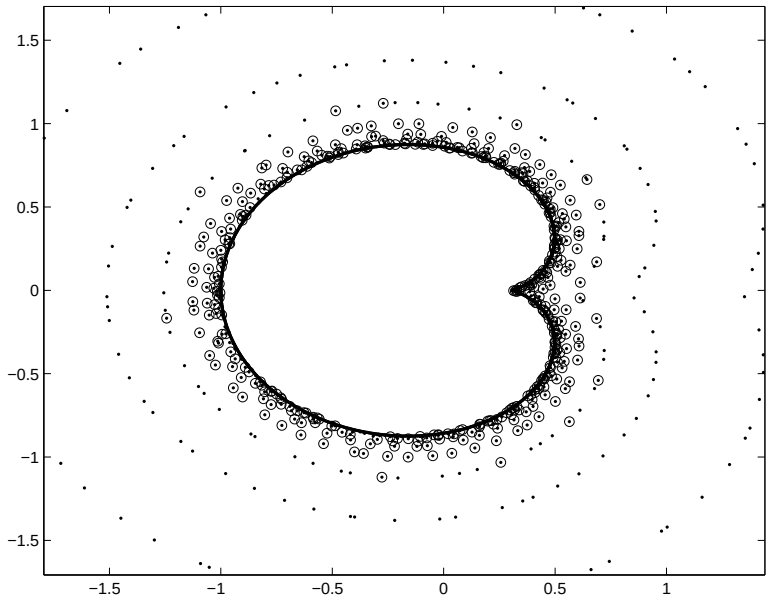

Figure 4: Lemniscate with source points, $L2$ algorithm

N	M	K	Alg	n	m	k	L_∞	L_2
704	1451	25	$L2$	471	1451	19	7.69e-004	3.41e-005
704	1451	25	$A2$	350	1451	13	8.13e-004	3.03e-005
704	1451	25	$A\infty$	356	1449	14	6.92e-003	2.01e-003

Table 8: MFS results for $\max(0, y)$

Finally, Table 10 shows how the sum of coefficients and the L_1 norm of coefficients vary with the radius r of the distance of the source points to the boundary. To avoid symmetries, we took the boundary data function $f(x, y) := \max(0, |y|)$ on the unit circle, offered 200 source points on a circle of radius r and used 401 collocation points. The solution method was the standard backslash L_2 solver from MATLAB©. Note that the increase of condition is counteracted by the solver in a very nice way, using fewer and fewer source points. This effect is even more significant when using the $A2$ or $A\infty$ methods.

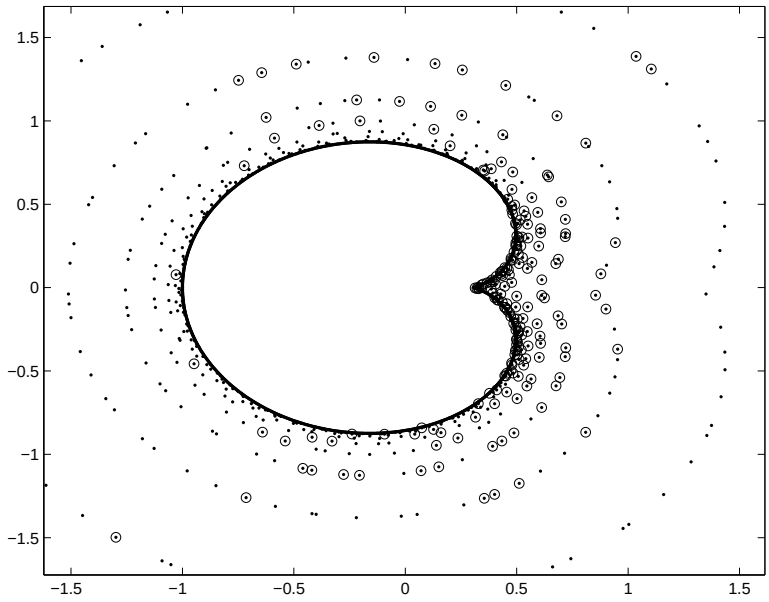

Figure 5: Lemniscate with source points, $A2$ algorithm

N	M	K	Alg	n	m	k	L_∞	L_2
0	1503	751	$L2$	0	1503	377	6.67e-004	4.08e-005
0	1503	751	$A2$	0	1503	377	6.67e-004	4.08e-005
0	1503	751	$A\infty$	0	379	377	1.47e-003	1.90e-004

Table 9: Results for $\max(0, y)$, harmonic polynomials only

References

[1] S. N. Atluri and T. L. Zhu. A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics*, 22:117–127, 1998.

[2] S. N. Atluri and T. L. Zhu. A new meshless local Petrov-Galerkin (MLPG) approach to nonlinear problems in computer modeling and simulation. *Computer Modeling and Simulation in Engineering*, 3:187–196, 1998.

[3] I. Barrodale and A. Young. Algorithms for best L_1 and L_∞ linear approximations on a discrete set. *Numer. Math.*, 8:295–306, 1966.

[4] W. Chen. Boundary knot method for Laplace and biharmonic problems. In *Proc. of the 14th Nordic Seminar on Computational*

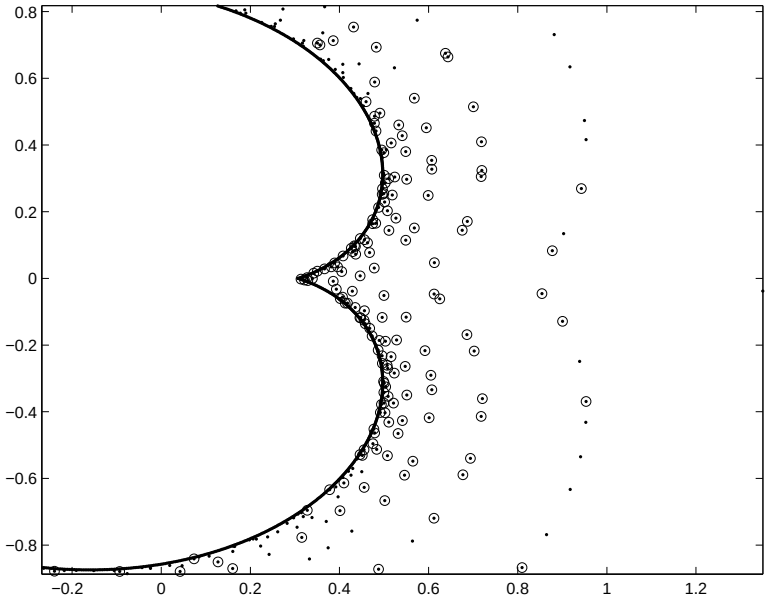

Figure 6: Lemniscate with source points, $A2$ algorithm

Mechanics, pages 117–120. Lund, Sweden, 2001.

[5] H.A. Cho, M.A. Golberg, A.S. Muleshkov, and X. Li. Trefftz methods for time dependent partial differential equations. *Computers, Materials, and Continua*, 1:1–38, 2004.

[6] G Fairweather and A. Karageorghis. The method of fundamental solution for elliptic boundary value problems. *Advances in Computatonal Mathematics*, 9:69–95, 1998.

[7] D. Gaier. *Lectures on Complex Approximation*. Birkhäuser, Boston, 1987.

n	r	ϵ_∞	sum	L_1 norm
57	1	1.13e-002	2.30e-001	7.23e+009
53	2	1.15e-002	1.45e-001	2.09e+011
35	4	1.86e-002	9.89e-002	6.44e+009
22	8	2.87e-002	7.24e-002	1.56e+011
18	16	3.50e-002	5.62e-002	3.33e+009
18	32	3.89e-002	4.57e-002	1.62e+012

Table 10: Sums of coefficients as functions of r

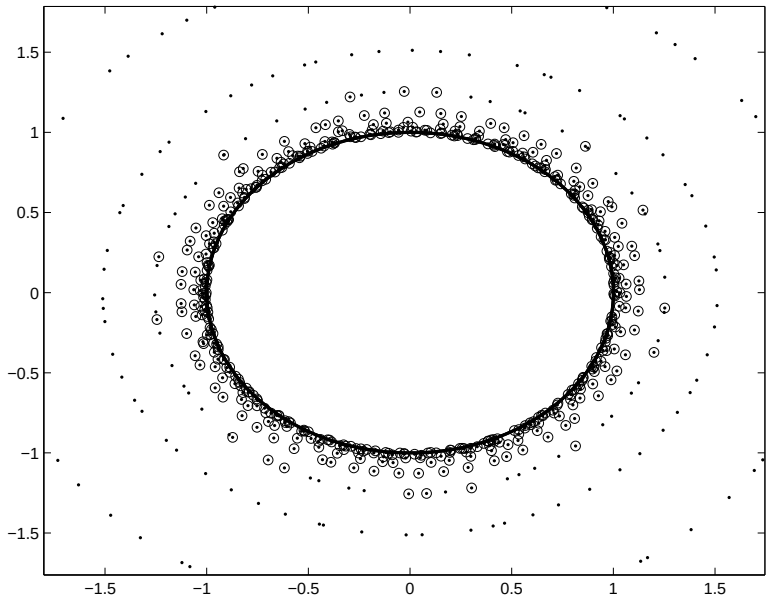

Figure 7: Source points selected by $L2$ algorithm

[8] M.A. Golberg and C.S. Chen. The method of fundamental solutions for potential, Helmholtz and diffusion problems. In M.A. Golberg, editor, *Boundary Integral Methods: Numerical and Mathematical Aspects*, pages 103–176. WIT Press, 1998.

[9] R. Hettich and K. O. Kortanek. Semi-infinite programming: theory, methods, and applications. *SIAM Rev.*, 35(3):380–429, 1993.

[10] Rainer Hettich and Peter Zencke. *Numerische Methoden der Approximation und semi-infiniten Optimierung*. B. G. Teubner, Stuttgart, 1982. Teubner Studienbücher Mathematik. [Teubner Mathematical Textbooks].

[11] Jürgen Jost. *Partial differential equations*, volume 214 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2002. Translated and revised from the 1998 German original by the author.

[12] E. J. Kansa. Application of Hardy's multiquadric interpolation to hydrodynamics. In *Proc. 1986 Simul. Conf., Vol. 4*, pages 111–117, 1986.

[13] E. J. Kansa. Multiquadrics - a scattered data approximation scheme with applications to computational fluid-dynamics - I: Surface approximation and partial dervative estimates. *Comput. Math. Appl.*, 19:127–145, 1990.

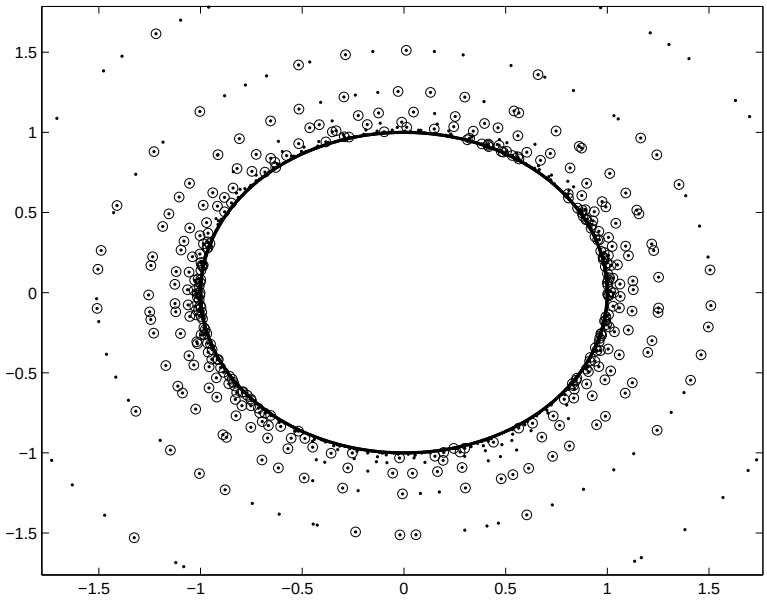

Figure 8: Source points selected by $A2$ algorithm

[14] A. Karageorghis and G. Fairweather. The method of fundamental solutions for the solution of nonlinear plane potential problems. *IMA Journal of Numerical Analysis*, 9:231–242, 1989.

[15] A. Karageorghis and G. Fairweather. The simple layer potential method of fundamental solutions for certain biharmonic problems. *International Journal of Numerical Methods in Fluids*, 9:1221–1234, 1989.

[16] Zi-Cai Li, Tzon-Tzer Lu, Hung-Tsai Huang, and Alexander H.-D. Cheng. Trefftz, collocation, and other boundary methods—a comparison. *Numer. Methods Partial Differential Equations*, 23(1):93–144, 2007.

[17] L.L. Ling, R. Opfer, and R. Schaback. Results on meshless collocation techniques. *Engineering Analysis with Boundary Elements*, 30:247–253, 2006.

[18] R. Mathon and R.L. Johnston. The approximate solution of elliptic boundary-value problems by fundamental solutions. *SIAM J. Numer. Anal.*, 14:638–650, 1977.

[19] R. Schaback. Limit problems for interpolation by analytic radial basis functions. Preprint, to appear in J. of Computational and Applied Mathematics, 2006.