Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. 1, No. 4, pp. 510-528

DOI: 10.4208/aamm.09-m0916
August 2009

A Practical Algorithm for Determining the Optimal Pseudo-Boundary in the Method of Fundamental Solutions

A. Karageorghis[1],*

[1]*Department of Mathematics and Statistics, University of Cyprus, P.O.Box 20537, 1678 Nicosia, Cyprus*

Received 21 January 2009; Accepted (in revised version) 03 April 2009
Available online 18 June 2009

Abstract. One of the main difficulties in the application of the method of fundamental solutions (MFS) is the determination of the position of the pseudo-boundary on which are placed the singularities in terms of which the approximation is expressed. In this work, we propose a simple practical algorithm for determining an estimate of the pseudo-boundary which yields the most accurate MFS approximation when the method is applied to certain boundary value problems. Several numerical examples are provided.

AMS subject classifications: 65N35, 65N38, 65K10
Key words: Method of fundamental solutions, elliptic boundary value problems, function minimization.

1 Introduction

The method of fundamental solutions (MFS) [7, 8, 12] is a relatively new meshless method for the solution of certain boundary value problems. In recent years, it has become increasingly popular because of the ease with which it can be implemented for problems in complicated geometries. The MFS is applicable to problems in which a fundamental solution of the operator of the governing equation is known. The solution is then approximated by linear combinations of fundamental solutions in terms of singularities which are placed on a pseudo-boundary lying outside the domain under consideration. A significant step was achieved in the 90's by Golberg and Chen who extended the applicability of the method to boundary value problems governed by inhomogeneous partial differential equations [11]. Since then the MFS has been applied to a wide variety of physical problems ranging from computer-aided design [31] to

*Corresponding author.
URL: `http://www.mas.ucy.ac.cy/english/teachers/karagheorgis_eng.htm`
Email: andreask@ucy.ac.cy (A. Karageorghis)

electrocardiography [32]. Further, the method has proved to be particularly effective for the solution of free boundary problems [17,27] and inverse problems [2,18,22–24].

One disadvantage of the MFS is that it relies on choosing a pseudo-boundary surrounding the boundary of the domain under consideration on which the singularities are placed. The position of this pseudo-boundary is crucial as the accuracy of the method depends on it. In early applications of the method, the positions of the singularities were taken to be among the unknowns [6, 14, 15, 25]. Collocation on a set of boundary points yielded a nonlinear system of equations which had to solved by means of nonlinear minimization routines. Although effective, this approach is computationally expensive. Despite several further attempts [3,29], the optimal placement of the singularities (i.e., the pseudo-boundary) in the MFS is still an open question. In this work, we propose a simple algorithm for determining an estimate of the optimal pseudo-boundary for certain elliptic boundary value problems.

In Section 2, we present in general terms, the type of problems that we examine. In Section 3, we identify the central problem to be addressed, namely the approximate location of the optimal pseudo-boundary, and subsequently describe the algorithm to be used. Several numerical examples are presented in Section 4 which show the efficacy of the proposed algorithm. In Section 5, the algorithm is applied to a biharmonic problem and shown to yield satisfactory results. Finally, some comments and conclusions are given in Section 6.

2 The method

We consider the boundary value problem

$$\begin{cases} \Delta u = 0, & \text{in} \quad \Omega, \\ u = f, & \text{on} \quad \partial\Omega, \end{cases} \tag{2.1}$$

where $\Omega \subset \mathbb{R}^D$, $D=2,3$ is a bounded domain with boundary $\partial\Omega$, and f is a given function.

In the MFS [7,8,12], the solution u is approximated by

$$u_N(\boldsymbol{c}, \boldsymbol{Q}; P) = \sum_{n=1}^{N} c_n \, K_D(P, Q_n), \qquad P \in \overline{\Omega}, \tag{2.2}$$

where $\boldsymbol{c}=[c_1, c_2, \ldots, c_N]^T$, $\boldsymbol{Q}$ is a $N-$vector containing the coordinates of the singularities Q_n, $n = 1, \ldots, N$, which lie outside $\overline{\Omega}$, and $K_D(P, Q)$ is a fundamental solution of the Laplace operator given by

$$K_D(P,Q) = \begin{cases} -\dfrac{1}{2\pi} \log \mid P - Q \mid, & D = 2, \\[2ex] \dfrac{1}{4\pi \mid P - Q \mid}, & D = 3, \end{cases} \tag{2.3}$$

with $|P-Q|$ denoting the distance between the points P and Q. The singularities Q_n are fixed on a pseudo-boundary $\partial\tilde{\Omega}$ similar to $\partial\Omega$ in the following sense [13]. At each point of the boundary we draw the outward normal to it and fix a point at a distance d from it. The pseudo-boundary is the curve joining all these points.
A set of collocation points $\{P_m\}_{m=1}^{M}$ is placed on $\partial\Omega$ and the coefficients $\boldsymbol{c}$ are determined so that the boundary condition is satisfied at these points; that is,

$$u_N(\boldsymbol{c},\boldsymbol{Q};P_m)=f(P_m), \quad m=1,\ldots,M. \tag{2.4}$$

This yields a linear system of M equations in N unknowns which has the form

$$G_D\boldsymbol{c} \;=\; \boldsymbol{f}, \tag{2.5}$$

where

$$\boldsymbol{f}=[f(P_1),f(P_2),\ldots,f(P_M)]^T,$$

and the elements of the $M \times N$ matrix G_D, D=2,3, are given by

$$G_{D\,m,n} \;=\; K_D(P_m,Q_n).$$

When M=N, the system is solved using a standard Gaussian elimination package. If M>N, we have an over–determined linear system which is solved using linear least squares routines. Once the vector $\boldsymbol{c}$ is known, the approximation u_N can be readily calculated at any point in Ω from (2.2). In another variant of the MFS, the locations of the singularities is unknown and thus their coordinates need to be found as part of the solution as well as the coefficients $\boldsymbol{c}$. This approach, which as mentioned in the Introduction was popular in early applications of the MFS, leads to a non-linear minimization problem which is considerably more expensive than the current approach.

3 The algorithm

One of the drawbacks of the MFS is the question of where to place the pseudo-boundary $\partial\tilde{\Omega}$. Regarding its shape, as already mentioned in Section 2, a natural choice is to take it to be similar to the boundary $\partial\Omega$ and at a fixed distance d from it. In [13], it is demonstrated that this choice is superior to the approach in which the pseudo-boundary is taken to be a circle (or a spherical surface) surrounding $\partial\Omega$. In the case of problem (2.1) when $\Omega\subset\mathbb{R}^2$ is a disk, it can be shown that, theoretically, as the distance between the (circular) pseudo-boundary and the boundary $\partial\Omega$ increases, the error in the MFS approximation tends exponentially to zero [20, 21, 30]. However, in practice this behaviour can only be observed for moderate values of d. When the distance d becomes large then the MFS system (2.5) becomes exceedingly ill-conditioned leading to loss of accuracy. These phenomena have also been observed for problems in general two- and three-dimensional domains. Thus there is a value of d which leads to optimal accuracy.

A practical way of finding the optimal $d{=}d_{opt}$ which has been used in the past for the solution of (2.1) is the following; see, for example, [31]:

Algorithm A

```
Step 1: For a fixed number of boundary collocation points $M$ and singularities $N$, choose a step $\delta d$.
Step 2: For each $j = 1, \cdots, \mathcal{N}$:
    (1) Solve the MFS system (2.5) for $d = j\,\delta d$.
    (2) Evaluate the approximate solution (2.2) on a set of $\mathcal{M}$ boundary points $\{S_\ell\}_{\ell=1}^{\mathcal{M}}$, different from the collocation points $\{P_m\}_{m=1}^{M}$.
    (3) Evaluate the absolute maximum error $e_j = \max_{\ell=1,\cdots,\mathcal{M}} |u_N(\boldsymbol{c}, \boldsymbol{Q}; S_\ell) - f(S_\ell)|$ on this set.
Step 3: Select the minimum $e_j$, say $e_{j^*}$. Thus $d_{opt} = j^* \delta d$.
```

The reason for evaluating the error only on the boundary is the following. Since both the solution u of problem (2.1) and the MFS approximation u_N are harmonic so is the error $e_N{=}u_N - u$. Therefore, from the maximum principle, the maximum of $|e_N|$ will be achieved on the boundary $\partial\Omega$, so it is sufficient to examine the absolute maximum error on the boundary.

It should be noted that this algorithm is also applicable in the case where the pseudo-boundary is taken to be a circle (in $\mathbb{R}^2$) or a spherical surface (in $\mathbb{R}^3$). In these cases, the parameter to be optimized is the radius of the pseudo-boundary.

Algorithm A is, however, potentially expensive since it requires the solution of $\mathcal{N}$ systems (2.5), as well as the evaluation of the solution at the $\mathcal{M}$ boundary points, $\mathcal{N}$ times.

One way of reducing this cost is to treat the evaluation of the maximum error e_j for each j, for fixed M and N, as a continuous function of d. Thus, one may use an algorithm for the minimization of a function of a single variable.

In `MATLAB`, this is achieved by using the function `fminbnb` which uses an algorithm based on the golden section search and parabolic interpolation [9]. The golden section search [16, Section 9.2], [28, Section 10.1] corresponds to the bisection method for locating the root of a function. If the function to be minimized is F, one starts with a triplet of points $a{<}b{<}c$, such that $F(b){<}F(a)$ and $F(b){<}F(c)$. The idea is to choose a new point d in either (a,b) or (b,c). If d is chosen in the interval (b,c) then, if $F(b){<}F(d)$ the new triplet is $a{<}b{<}d$ whereas if $F(d){>}F(b)$, the new triplet is $b{<}d{<}c$. The question is how to choose d in the first place. It can be shown, that if $h_1{=}|b-a|$, and $h_2{=}|c-b|$, the optimal location of d is in the larger of the two intervals (a,b) and (b,c) and at a distance

$$\frac{3-\sqrt{5}}{2}\max\{h1, h2\},$$

from b into the larger interval. The name of this method comes from the presence of the number $(1+\sqrt{5})/2$, known as the golden ratio. Once the new triplet is found, the process is repeated until the distance between the points of the triplet reaches a prescribed tolerance. In parabolic interpolation [28, Section 10.2], a parabola is fitted through the three points $(a, F(a)), (b, F(b))$ and $(c, F(c))$ and then d is taken to be the x-coordinate of the minimum of the parabola. The process is then repeated as in the golden section search. For parabolic interpolation to be effective it needs to be used in combination with another method, such as the golden section search.

If for fixed M and N and for a particular choice of $\mathcal{M}$ the function that evaluates the maximum error e_j is `mfs(x)`, then the following call yields the minimum of the maximum error for d in the interval (x_1, x_2):

$$[\mathtt{x}, \mathtt{fval}, \mathtt{exitflag}, \mathtt{output}] = \mathtt{fminbnd}('\mathtt{mfs(x)}', \mathtt{x1}, \mathtt{x2}).$$

The number of iterations and function evaluations in `fminbnb` can be substantially reduced without significantly affecting the position of the optimal pseudo-boundary. As is shown in the examples, this can be achieved by imposing a termination tolerance `TolX` on `x` of 1.e-2 instead of the default termination tolerance `TolX`=1.e-4. This is carried out by the call:

$$[\mathtt{x}, \mathtt{fval}, \mathtt{exitflag}, \mathtt{output}] = \mathtt{fminbnd}('\mathtt{mfs(x)}', \mathtt{x1}, \mathtt{x2}, \mathtt{optimset}('\mathtt{TolX}', \mathtt{1e}-2)).$$

An obvious question concerns the choice of the bounds `x1` and `x2`. Clearly, `x1` needs to be close to the boundary so a small value needs to be chosen and in the examples this was taken to be `x1`= 0.01. The choice of `x2` is not as easy, as for an insufficiently large value there is the danger of missing the minimum. One may apply Algorithm A with a large step δd and a small $\mathcal{N}$ to see where the value of the function increases. In our numerical experiments we found that for two-dimensional problems, taking `x2`=4 and for three-dimensional problems taking `x2`=10, yielded satisfactory results. Taking larger values did not significantly affect the position of the minimum. Alternatively, one may use a simple routine for initially bracketing a minimum see, e.g., [28, Section 10.1].

Also, as is observed in the examples, in many cases there is a relatively wide range of choices of d_{opt} which yield very similar accuracy. Our goal is therefore to obtain a value of d_{opt} in that range, which explains the relaxation of the tolerance on `x`.

Note. In `Fortran` one may use the NAG Library routines `E04ABF–E04ABA` [26] which find the minimum of a function of a single variable by a method based on quadratic interpolation [10]. Alternatively, one may use the golden section search in one dimension [28, Section 10.1] or parabolic interpolation and Brent's method in one dimension [28, Section 10.2].

4 Numerical examples

4.1 Example 1

We first consider problem (2.1) with exact solution

$$u = \mathrm{e}^{4x} \cos 4y,$$

in the case when the domain $\Omega \subset \mathbb{R}^2$ is the unit disk. The pseudo-boundary $\partial\tilde{\Omega}$ is taken to be a concentric circle of radius $R=1+d$, $d>0$. We first applied Algorithm A with $\delta d=0.01$, $\mathcal{N}=400$ and $M=N=32, 48, 64$ and 80. Both the boundary points and singularities are uniformly distributed on $\partial\Omega$ and $\partial\tilde{\Omega}$, respectively. It should be noted that for this particular example, the MFS coefficient matrix is circulant [4], and thus the solution of the MFS systems can be carried out efficiently using fast Fourier transforms; see, for example, [30].

In Fig. 1, we present the maximum error calculated at $\mathcal{M}=100$ uniformly distributed points on $\partial\Omega$. In Table 1, we present the corresponding results when using the function `fminbnb` with $x_1=0.01$, and $x_2=4$. The results obtained using a relaxed tolerance on `x` of `TolX=1e-2` are presented in Table 2. As can be observed from these tables, the relaxation on the tolerance leads to no significant difference in the value of d_{opt}, while reducing significantly both the number of iterations and function evaluations. The values of d_{opt} presented in Tables 1 and 2 and which are obtained with few iterations and function evaluations, clearly agree well with the corresponding values observed in Fig. 1. In Fig. 2 we present the CPU times required for the location of the optimal pseudo-boundary using Algorithm A, minimization using the default tolerance on `x` and minimization using a relaxed tolerance of `TolX=1e-2`. As can be

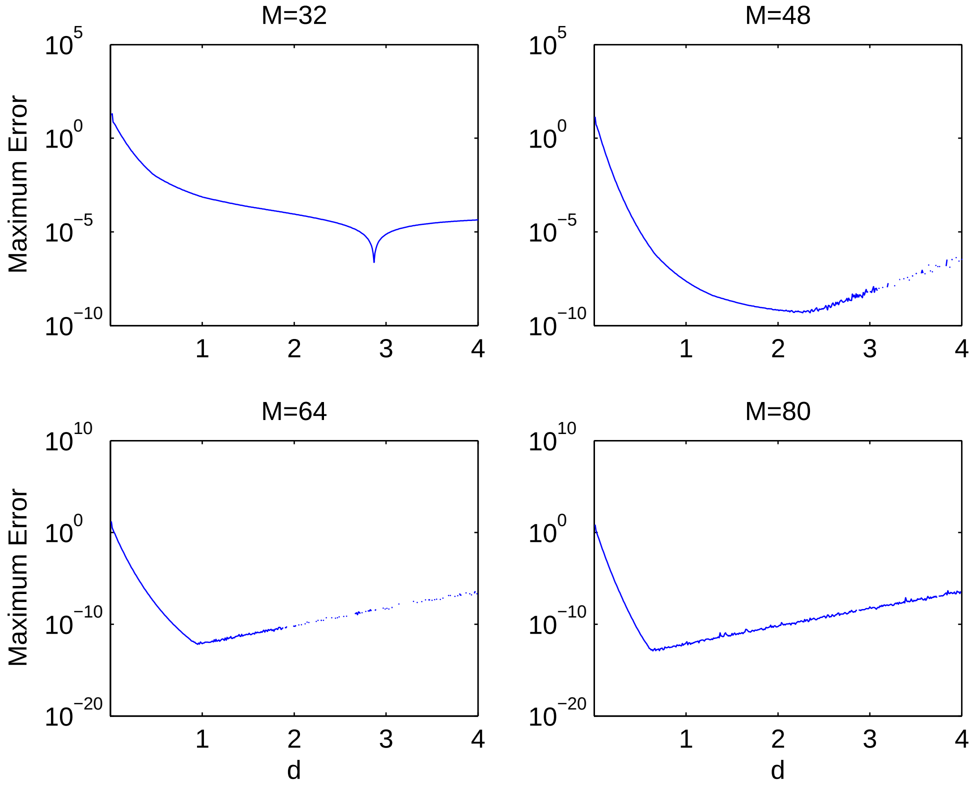

Figure 1: Results with $\delta d=0.01$, $\mathcal{N}=400$ for Example 1.

Table 1: Results for Example 1 using minimization with default tolerance. FE represents function evaluations.

$M=N$	d_{opt}	Max error	Iterations	FE
32	2.8699	2.2431(-7)	17	19
48	2.2494	4.6850(-10)	12	14
64	1.0340	7.5615(-13)	18	20
80	0.6978	1.0658(-13)	16	18

Table 2: Results for Example 1 using minimization with relaxed tolerance. FE represents function evaluations.

$M=N$	d_{opt}	Max error	Iterations	FE
32	2.8714	2.9719(-7)	10	12
48	2.2494	4.6850(-10)	10	12
64	1.0076	7.5419(-13)	11	13
80	0.6950	1.3500(-13)	13	15

observed from this figure, the use of minimization, especially with reduced tolerance, leads to substantial savings over Algorithm A in the CPU time required to locate d_{opt}.

4.2 Example 2

We next considered problem (2.1) with exact solution

$$u = e^{4x} \cos 4y,$$

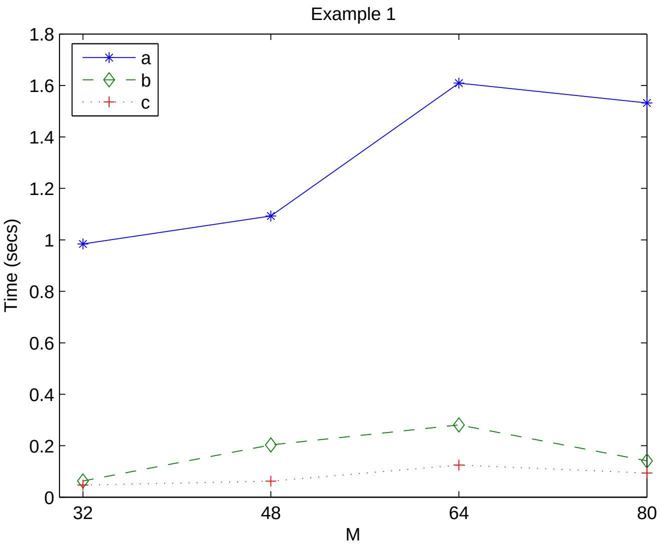

Figure 2: CPU times for Example 1: (a) Algorithm A; (b) Minimization with default tolerance; (c) Minimization with relaxed tolerance.

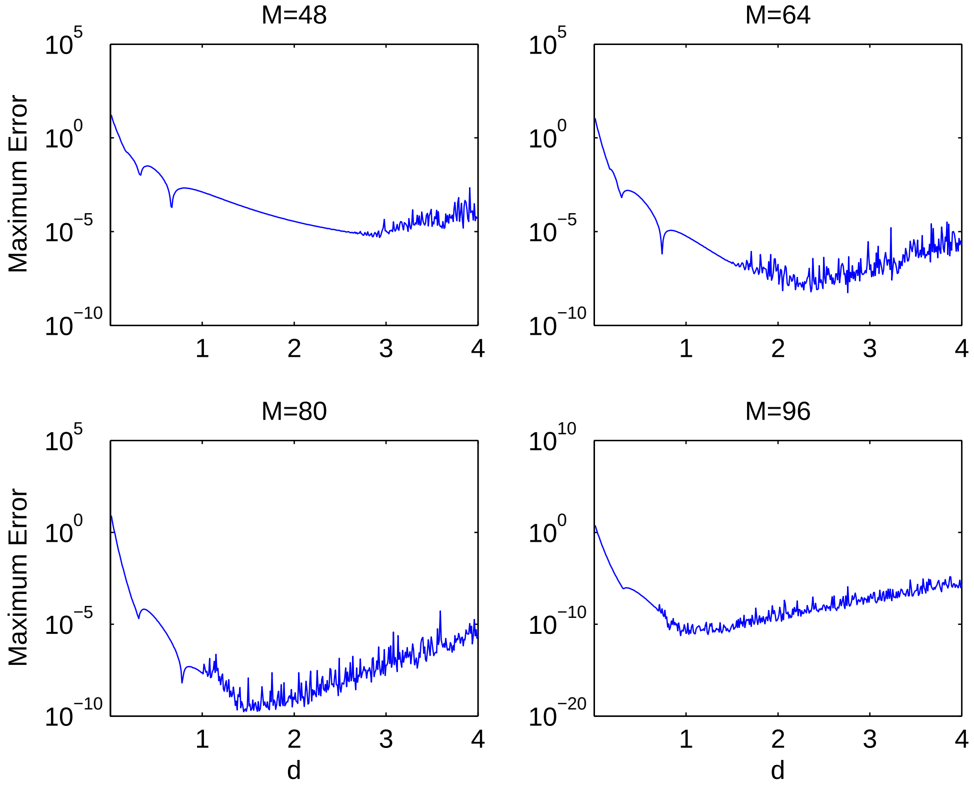

Figure 3: Results with $\delta d = 0.01$, $\mathcal{N} = 400$ for Example 2.

in the case when the domain $\Omega \subset \mathbb{R}^2$ is the square $(-1,1) \times (-1,1)$. The pseudo-boundary $\partial\tilde{\Omega}$ is taken to be a concentric square $(-R,R) \times (-R,R)$ where $R{=}1+d$, $d{>}0$. We first applied Algorithm A with $\delta d{=}0.01$, $\mathcal{N}{=}400$ and $M{=}N{=}48$, 64, 80 and 96. Both the boundary points and singularities are uniformly distributed on $\partial\Omega$ and $\partial\tilde{\Omega}$, respectively.

In Fig. 3, we present the maximum error calculated at $\mathcal{M}{=}100$ uniformly spread points on $\partial\Omega$. In Table 3 we present the corresponding results when using the function `fminbnb` with $x_1{=}0.01$, and $x_2{=}4$. The results obtained using a relaxed tolerance on `x` of `TolX=1e-2` are presented in Table 4. The same conclusions as the ones observed

Table 3: Results for Example 2 using minimization with default tolerance.

$M{=}N$	d_{opt}	Max error	Iterations	FE
48	2.9375	6.3065(-6)	20	22
64	2.4760	5.3999(-9)	22	24
80	1.6847	9.7000(-11)	22	24
96	1.1480	4.8033(-12)	22	24

Table 4: Results for Example 2 using minimization with relaxed tolerance.

$M=N$	d_{opt}	Max error	Iterations	FE
48	2.7025	7.6444(-6)	12	14
64	2.4760	5.3999(-9)	10	12
80	1.6891	2.0784(-10)	10	12
96	1.1479	5.3362(-12)	12	14

in Example 1 also apply in this case. As in Example 1, the values of d_{opt} presented in Tables 3 and 4 are in very good agreement with the corresponding values from Fig. 3. In Fig. 4, we present the CPU times required for the location of the optimal pseudo-boundary using Algorithm A, minimization using the default tolerance on `x` and minimization using a relaxed tolerance of `TolX=1e-2`. As in Example 1, this shows that the use of minimization, leads to substantial savings in CPU time.

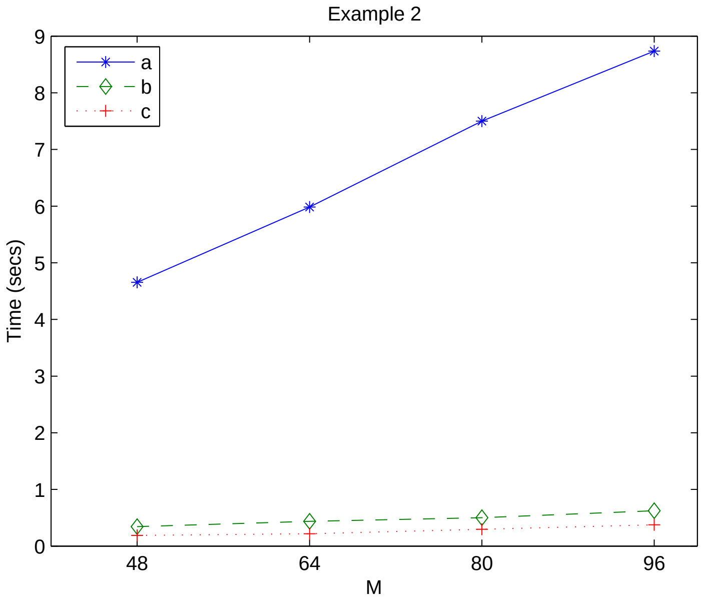

Figure 4: CPU times for Example 2: (a) Algorithm A; (b) Minimization with default tolerance; (c) Minimization with relaxed tolerance.

4.3 Example 3

We next consider problem (2.1) with exact solution

$$u = \cosh 5x \cos 4y \sin 3z,$$

in the case when the domain $\Omega \subset \mathbb{R}^3$ is the unit sphere. The pseudo-boundary $\partial\tilde{\Omega}$ is taken to be the concentric sphere of radius $R=1+d$, $d>0$. We first applied Algorithm

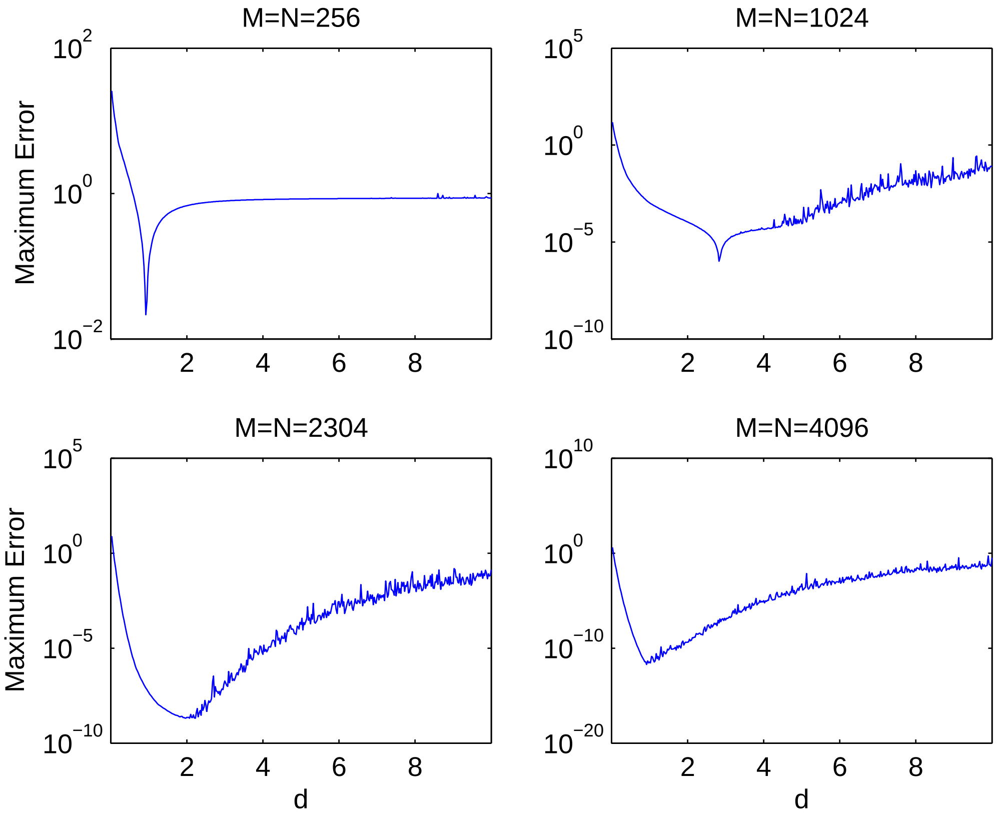

Figure 5: Results with $\delta d = 0.025$, $\mathcal{N} = 400$ for Example 3.

A with δd=0.025, $\mathcal{N}$=400 and M=N=256, 1024, 2304 and 4096. Both the boundary points and singularities are uniformly distributed on $\partial\Omega$ and $\partial\tilde{\Omega}$, respectively. In Fig. 5, we present the maximum error calculated at $\mathcal{M}$=5625 uniformly distributed points on $\partial\Omega$. It should be noted that for this particular example, the MFS systems possess a block circulant structure. Thus, the solution of these systems is carried out efficiently using a matrix decomposition algorithm and fast Fourier transforms; see, for example, [19].

In Table 5, we present the results when using the function `fminbnb` with x_1=0.01 and x_2=10, while in Table 6 we present the corresponding results with a relaxed toler-

Table 5: Results for Example 3 using minimization with default tolerance.

M=N	d_{opt}	Max error	Iterations	FE
256	0.9331	1.1786(-2)	21	24
1024	2.8318	7.8943(-7)	18	21
2304	1.9445	2.0606(-9)	24	27
4096	1.5636	3.3011(-11)	22	25

Table 6: Results for Example 3 using minimization with relaxed tolerance.

$M{=}N$	d_{opt}	Max error	Iterations	FE
256	0.9322	1.2858(-2)	13	16
1024	2.8319	8.1189(-7)	11	14
2304	1.9445	2.0606(-9)	15	18
4096	1.5642	4.0627(-11)	14	14

ance of `TolX=1e-2`. As in the two previous examples, the values of d_{opt} presented in Tables 5 and 6 are in good agreement with the corresponding values from Fig. 5. In Fig. 6, we present the CPU times required for the location of the optimal pseudo-boundary using Algorithm A, minimization using the default tolerance on `x` and minimization using a relaxed tolerance of `TolX=1e-2`. In a way considerably more pronounced than in Examples 1 and 2, we observe that the use of minimization, especially with relaxed tolerance, leads to substantial savings in CPU time.

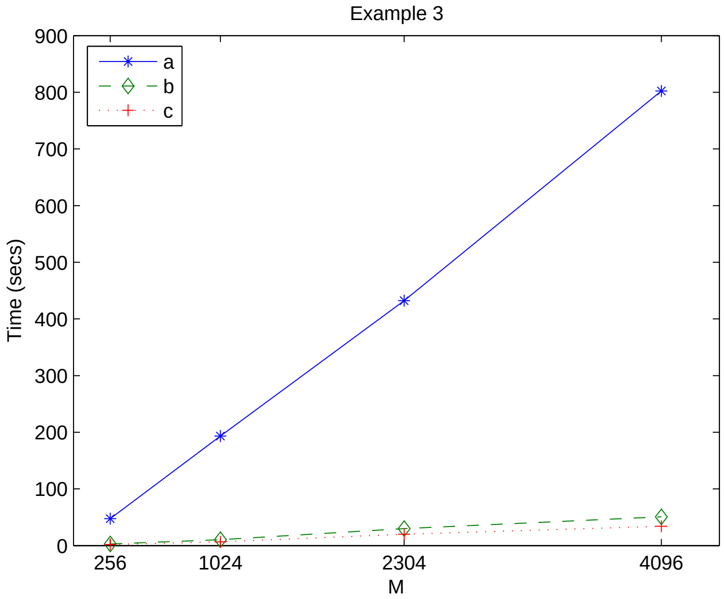

Figure 6: CPU times for Example 3: (a) Algorithm A; (b) Minimization with default tolerance; (c) Minimization with relaxed tolerance.

4.4 Example 4

Finally, we consider problem (2.1) in the case when the domain $\Omega\subset\mathbb{R}^3$ is the cube $(-1,1)\times(-1,1)\times(-1,1)$. The pseudo-boundary $\partial\tilde{\Omega}$ is taken to be the concentric cube $(-R,R)\times(-R,R)\times(-R,R)$ where $R{=}1+d$, $d{>}0$. In order to test the algorithm further, we consider the problem with a non-harmonic boundary condition, namely

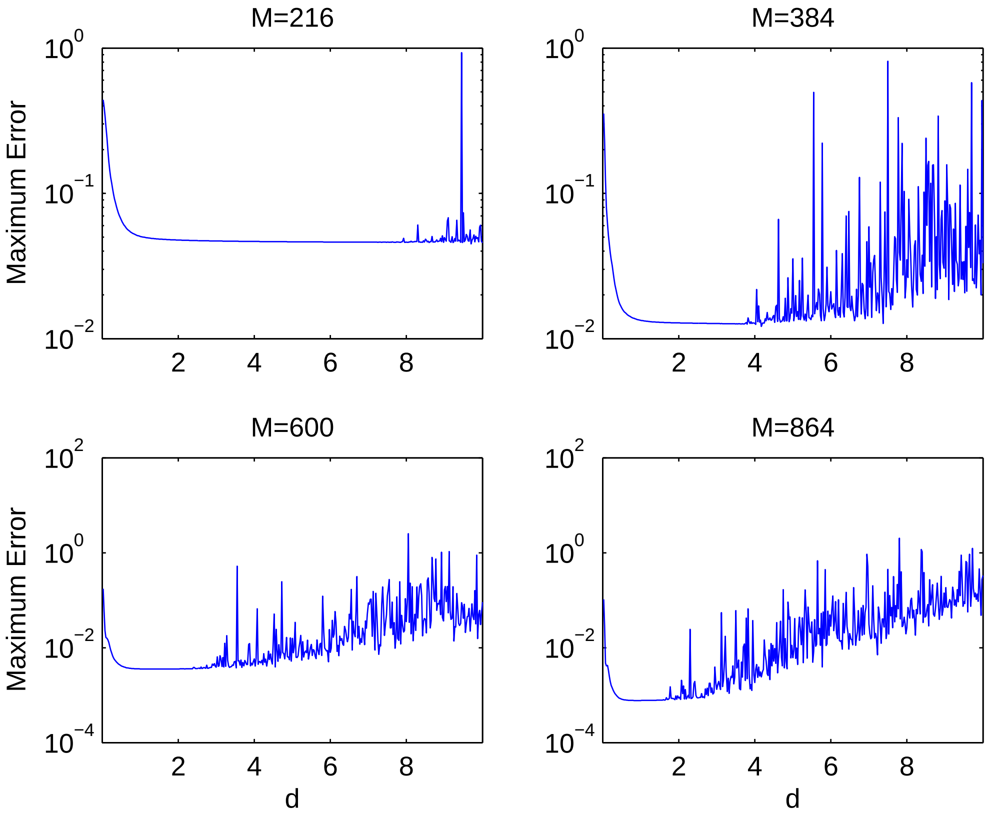

Figure 7: Results with $\delta d = 0.025$, $\mathcal{N} = 400$ for Example 4.

with

$$f(x,y,z) = x^2y^3z^4.$$

Although the exact solution is not known, the maximum principle is still valid and thus one can evaluate the maximum error. The behaviour of the MFS approximation to such problems was studied extensively in [5]. We first applied Algorithm A with δd=0.025, $\mathcal{N}$=400 and M=N=216, 384, 600 and 864. Both the boundary points and singularities are uniformly distributed on $\partial\Omega$ and $\partial\tilde{\Omega}$, respectively.

In Fig. 7, we present the maximum error calculated at $\mathcal{M}$=1176 uniformly distributed points on $\partial\Omega$. In Table 7 we present the results when using the function

Table 7: Results for Example 4 using minimization with default tolerance.

M=N	d_{opt}	Max error	Iterations	FE
216	7.5655	4.6096(-2)	19	22
384	3.8125	1.2566(-2)	25	28
600	2.4701	3.6278(-3)	17	20
864	0.8973	7.7214(-4)	17	20

Table 8: Results for Example 4 using minimization with relaxed tolerance.

M=N	d_{opt}	Max error	Iterations	FE
216	7.5653	4.6105(-2)	7	10
384	3.8122	1.2577(-2)	15	18
600	2.5013	3.5912(-3)	14	17
864	0.8973	7.7214(-4)	13	16

`fminbnb` with x_1=0.01 and x_2=9.9, while in Table 8 we present the corresponding results with a relaxed tolerance of `TolX=1e-2`. As in the three previous examples, the values of d_{opt} presented in Tables 7 and 8 are in good agreement with the corresponding values from Fig. 7. In Fig. 8 we present the CPU times required for the location of the optimal pseudo-boundary using Algorithm A, minimization using the default tolerance on `x` and minimization using a relaxed tolerance of `TolX=1e-2`. As was observed in Example 3, the use of minimization, especially with relaxed tolerance, leads to substantial savings in CPU time.

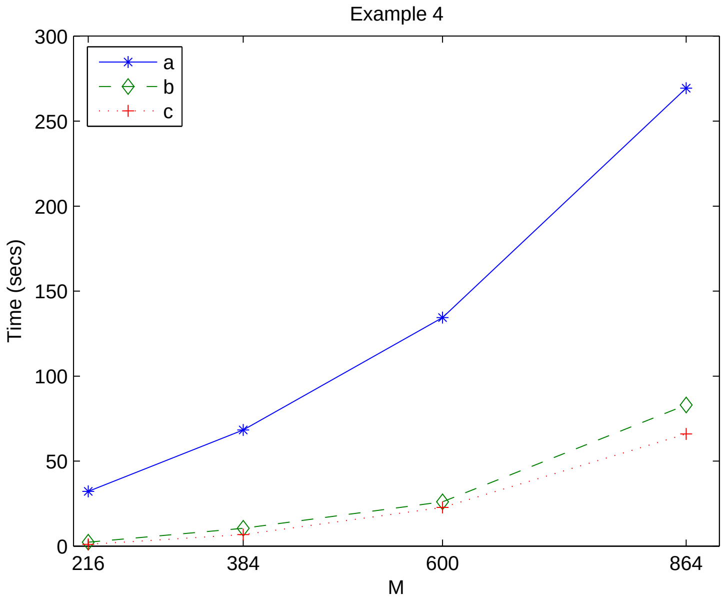

Figure 8: CPU times for Example 4: (a) Algorithm A; (b) Minimization with default tolerance; (c) Minimization with relaxed tolerance.

5 Extensions

In this section, we apply the ideas presented in Section 3 to a problem in which the differential equation does not satisfy a maximum principle. Although there is no theoretical justification, it appears natural to choose a pseudo-boundary for which the boundary conditions are satisfied as accurately as possible.

As an example, we consider the biharmonic Dirichlet problem

$$\begin{cases} \Delta^2 u = 0, & \text{in} \quad \Omega, \\ u = f, & \text{on} \quad \partial\Omega, \\ \dfrac{\partial u}{\partial \mathrm{n}} = g, & \text{on} \quad \partial\Omega, \end{cases} \tag{5.1}$$

where $\Omega \subset \mathbb{R}^2$ is a bounded domain with boundary $\partial\Omega$, $\partial/\partial\mathrm{n}$ is the derivative in the outward normal direction to the boundary and f and g are given functions.

In the MFS, we take [7]

$$u_N(\boldsymbol{c}, \boldsymbol{d}, \boldsymbol{Q}; P) = \sum_{n=1}^{N} c_n K_2(P, Q_n) + \sum_{n=1}^{N} d_n L_2(P, Q_n), \qquad P \in \overline{\Omega}, \tag{5.2}$$

where $\boldsymbol{d} = [d_1, d_2, \ldots, d_N]^T$ and $\boldsymbol{c}$, $\boldsymbol{Q}$ and $K_2(P, Q)$ are defined as in the harmonic case. Also, $L_2(P, Q)$ is a fundamental solution of the biharmonic operator given by

$$L_2(P, Q) = -\frac{1}{8\pi} |P - Q|^2 \log |P - Q|. \tag{5.3}$$

As before, a set of collocation points $\{P_m\}_{m=1}^{M}$ is placed on $\partial\Omega$ and the coefficients $\boldsymbol{c}$ and $\boldsymbol{d}$ are determined so that the boundary conditions are satisfied at these points; that is,

$$u_N(\boldsymbol{c}, \boldsymbol{Q}; P_m) = f(P_m), \quad \text{and} \quad \partial u_N(\boldsymbol{c}, \boldsymbol{Q}; P_m)/\partial\mathrm{n} = g(P_m), \quad m = 1, \ldots, M. \tag{5.4}$$

This yields a linear system of $2M$ equations in $2N$ unknowns which has the form

$$\left(\begin{array}{c|c} G_{11} & G_{12} \\ \hline G_{21} & G_{22} \end{array} \right) \left(\begin{array}{c} \boldsymbol{c} \\ \hline \boldsymbol{d} \end{array} \right) = \left(\begin{array}{c} \boldsymbol{f} \\ \hline \boldsymbol{g} \end{array} \right), \tag{5.5}$$

where

$$\boldsymbol{f} = [f(P_1), f(P_2), \ldots, f(P_M)]^T, \quad \boldsymbol{g} = [g(P_1), g(P_2), \ldots, g(P_M)]^T,$$

and the elements of the four $M \times N$ matrices $G_{11}, G_{12}, G_{21}, G_{22}$ are given by

$$\begin{aligned} (G_{11})_{m,n} &= K_2(P_m, Q_n), & (G_{12})_{m,n} &= L_2(P_m, Q_n), \\ (G_{21})_{m,n} &= \partial K_2(P_m, Q_n)/\partial\mathrm{n}, & (G_{22})_{m,n} &= \partial L_2(P_m, Q_n)/\partial\mathrm{n}. \end{aligned}$$

As in the harmonic case, we take the pseudo-boundary to be similar to the boundary $\partial\Omega$ and at a fixed distance d from it. A practical way of finding a value of $d = d_a$ which yields an accurate MFS approximation is to apply Algorithm A to the problem. Subsequently, we use an algorithm for the minimization of a function of a single variable to determine d_a. This is illustrated in the following example.

5.1 Example 5

We consider problem (5.1) with exact solution

$$u = (x^2 + y^2)\, \mathrm{e}^{4x} \cos 4y \,+\, \mathrm{e}^{2x} \cos 2y,$$

in the case when the domain $\Omega \subset \mathbb{R}^2$ is the square $(-1,1) \times (-1,1)$. As in Example 2, the pseudo-boundary $\partial\tilde{\Omega}$ is taken to be the square $(-R,R) \times (-R,R)$ where $R=1+d$, $d>0$. We applied the MFS for $d=j\,\delta d$, $j=1,\cdots,\mathcal{N}$ where $\delta d=0.01$, $\mathcal{N}=400$ and evaluated the maximum absolute error in u_N on a 25×25 uniform grid of interior points. The results are presented in Fig. 9. We then applied Algorithm A with the same parameters and the maximum error in u_N and $\partial u_N/\partial \mathrm{n}$ calculated at $\mathcal{M}=100$ uniformly spread points on $\partial\Omega$. The results are presented in Fig. 10. From Figs. 9 and 10 it is evident that the errors behave in a similar way. In Table 9 we present the corresponding results when using the function `fminbnb` with a relaxed tolerance of `TolX=1e-2`. As in the harmonic examples considered we observe that the values of d_a presented in Table 9 agree well with the corresponding values observed in Figs. 9 and 10.

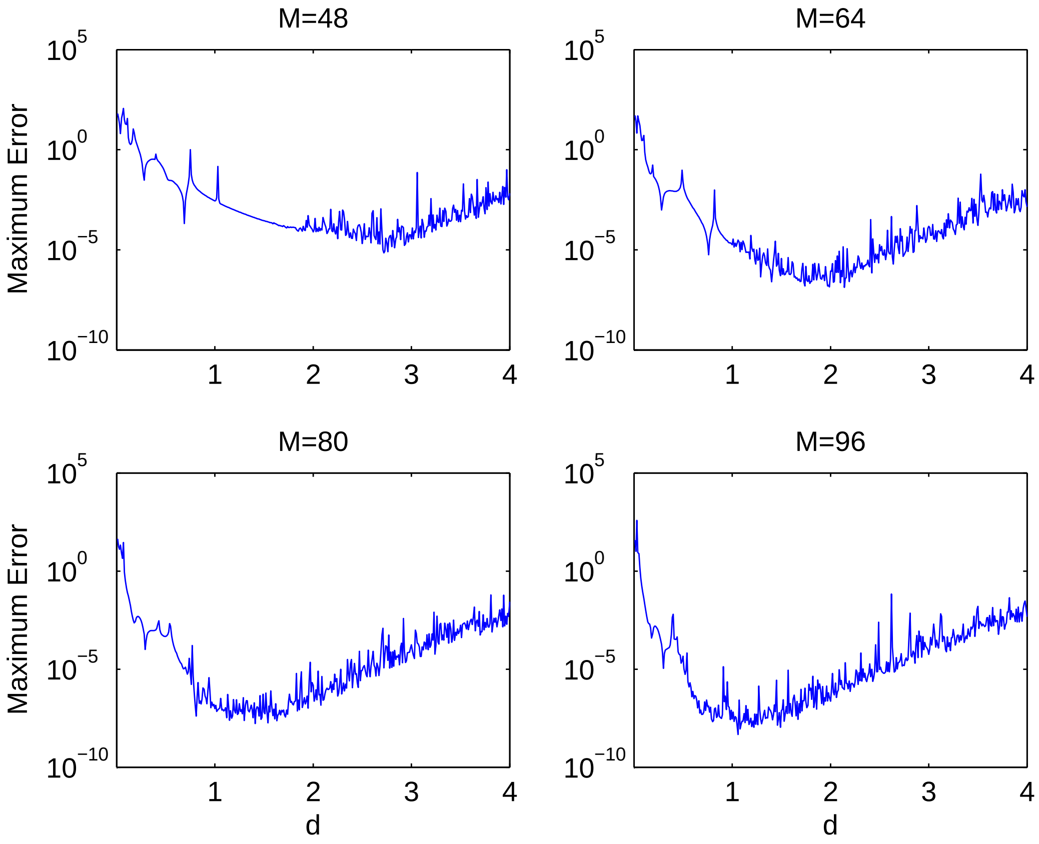

Figure 9: Maximum error in u_N on interior grid with $\delta d = 0.025$, $\mathcal{N} = 400$ for Example 5.

Table 9: Results for Example 5 using minimization with relaxed tolerance.

M=N	d_a	Max error	Iterations	FE
48	3.0581	1.1086(-4)	9	11
64	1.9479	2.5894(-7)	13	15
80	1.4083	3.2858(-8)	13	15
96	1.2032	3.6007(-9)	9	11

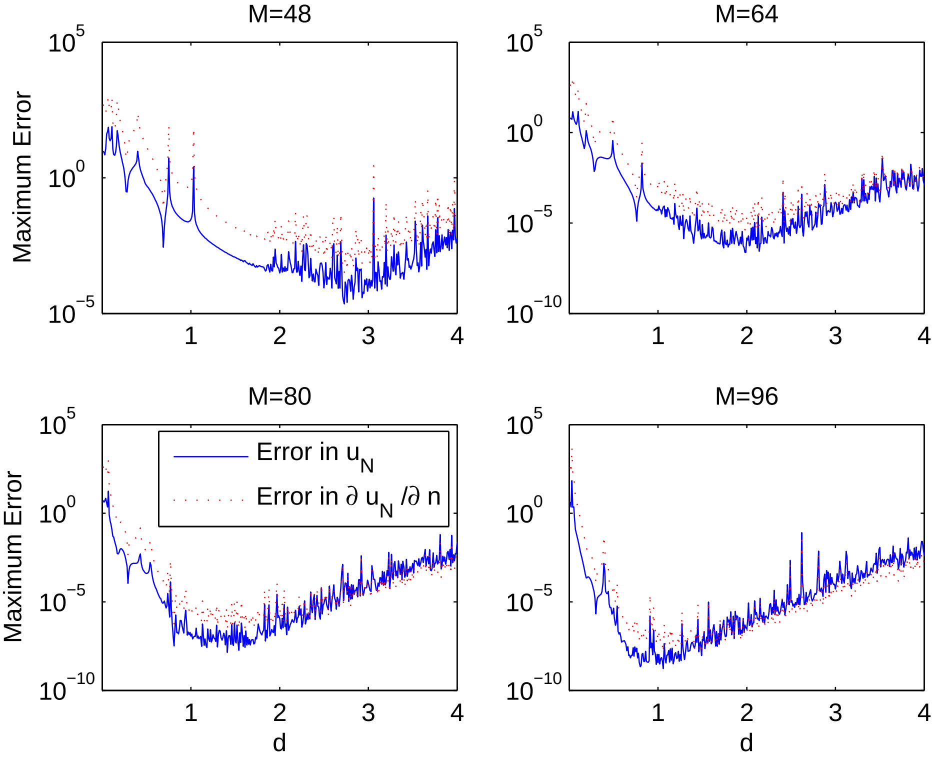

Figure 10: Maximum errors in u_N and $\partial u_N/\partial n$ on boundary with $\delta d = 0.025$, $\mathcal{N} = 400$ for Example 5.

Note. In Step 2 (3) of Algorithm A, one could have chosen

$$e_j = \max\Big\{ \max_{\ell=1,\cdots,\mathcal{M}} |u_N(\boldsymbol{c},\boldsymbol{Q};S_\ell) - f(S_\ell)|, \max_{\ell=1,\cdots,\mathcal{M}} |\frac{\partial u_N}{\partial n}(\boldsymbol{c},\boldsymbol{Q};S_\ell) - g(S_\ell)| \Big\}.$$

Such a choice was found not to significantly alter the value of d_a.

6 Conclusions

In this work we propose a simple and efficient algorithm for obtaining an approximation to the optimal location of the pseudo-boundary when the MFS is applied to harmonic boundary value problems. The algorithm, which relies on the maximum principle for harmonic functions, is based on considering the routine evaluating the maximum error in the Dirichlet boundary conditions, for fixed numbers of degrees of freedom, as a function of the distance of the pseudo-boundary from the boundary. The idea is to minimize this function by using existing software to yield an optimal value for this distance. Numerical experiments indicate that by applying this algorithm we obtain satisfactory estimates for the optimal position of the pseudo-boundary at little computational cost. The savings in CPU time are more pronounced in the case of three-dimensional problems. It is also noteworthy that from Tables 1–8, it can be observed that d_{opt} decreases as $M{=}N$ increase. It is shown that the same technique can also be applied for the determination of a pseudo-boundary which yields accurate approximations when the MFS is applied to biharmonic problems where the governing equation does not satisfy the maximum principle. The current algorithm is not restricted to the case when the pseudo-boundary is taken to be similar to the boundary of the region under consideration, but is also applicable in the case where the pseudo-boundary is taken to be a circle surrounding the region (in $\mathbb{R}^2$) or a spherical surface (in $\mathbb{R}^3$).

It should be mentioned that for problems governed by inhomogeneous equations or time-dependent problems, the optimal location of source points is not as important an issue as in the homogeneous case. This is because in these cases, usually the dominant errors come from the approximation of the inhomogeneous terms or time discretization schemes. As a result the accuracy of the homogeneous solution using the MFS has little effect on the overall accuracy.

One interesting question related to this application of the MFS is the choice of the pseudo-boundary in the case where the solution has (unknown) singularities at the points $P_k{\notin}\bar{\Omega}$, $k{=}\ldots,K$. Clearly, as the pseudo-boundary moves away from the boundary, the accuracy of the MFS will improve until it either meets the singularity located closest to the boundary or the ill-conditioning of the MFS system leads to loss of accuracy; see, for example, [29, pages 12-13] or [30]. The applicability of the current approach clearly depends on the location of the closest singularity to the boundary. Using the current algorithm or a variation of it to determine the location of this singularity could be the subject of future research.

Acknowledgements

The author would like to thank Professor Graeme Fairweather and Professor Daniel Lesnic for their useful comments as well as the two anonymous referees for their constructive comments and suggestions.

References

[1] R. P. BRENT, Algorithms for Minimization without Derivatives, Prentice-Hall, Engle-

wood Cliffs, New Jersey, 1973.

[2] D. Borman, D. B. Ingham, B. T. Johansson and D. Lesnic, *The method of fundamental solutions for detection of cavities in EIT*, J. Integral Equations Appl., to be appeared.

[3] A. P. Cisilino and B. Sensale, *Application of a simulated annealing algorithm in the optimal placement of the source points in the method of the fundamental solutions*, Comput. Mech., 28 (2002), pp. 129–136.

[4] P. J. Davis, Circulant Matrices, John Wiley & Sons, New York, 1979.

[5] T. W. Drombosky, A. L. Meyer and L. Ling, *Applicability of the method of fundamental solutions*, Eng. Anal. Bound. Elem., 27 (2009), pp. 637–643.

[6] G. Fairweather and R. L. Johnston, *The method of fundamental solutions for problems in potential theory*, in: Treatment of Integral Equations by Numerical Methods (Durham, 1982), Academic Press, London, pp. 349–359, 1982.

[7] G. Fairweather and A. Karageorghis, *The method of fundamental solutions for elliptic boundary value problems*, Adv. Comput. Math., 9 (1998), pp. 69–95.

[8] G. Fairweather, A. Karageorghis and P. A. Martin, *The method of fundamental solutions for scattering and radiation problems*, Eng. Anal. Bound. Elem., 27 (2003), pp. 759–769.

[9] G. E. Forsythe, M. A. Malcolm and C. B. Moler, Computer Methods for Mathematical Computations, Prentice-Hall, Englewood Cliffs, New Jersey, 1976.

[10] P. E. Gill and W. Murray, *Safeguarded steplength algorithms for optimization using descent methods*, NPL Report NAC 37, National Physical Laboratory, 1973.

[11] M. A. Golberg and C. S. Chen, Discrete Projection Methods for Integral Equations, Computational Mechanics Publications, Southampton, 1997.

[12] M. A. Golberg and C. S. Chen, *The method of fundamental solutions for potential, Helmholtz and diffusion problems*, in: Boundary Integral Methods and Mathematical Aspects, ed. M. A. Golberg, WIT Press/Computational Mechanics Publications, Boston, pp. 103–176, 1999.

[13] P. Gorzelańczyk and J. A. Kołodziej, *Some remarks concerning the shape of the source contour with application of the method of fundamental solutions to elastic torsion of prismatic rods*, Eng. Anal. Bound. Elem., 32 (2008), pp. 64–75.

[14] R. L. Johnston and G. Fairweather, *The method of fundamental solutions for problems in potential flow*, Appl. Math. Model., 8 (1984), pp. 265–270.

[15] R. L. Johnston and R. Mathon, *The computation of electric dipole fields in conducting media*, Int. J. Numer. Meth. Eng., 14 (1979), pp. 1739–1760.

[16] D. Kahaner, C. Moler, and S. Nash, Numerical Methods and Software, Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[17] A. Karageorghis, *The method of fundamental solutions for the solution of steady-state free boundary problems*, J. Comput. Phys., 98 (1992), pp. 119–128.

[18] A. Karageorghis and D. Lesnic, *Detection of cavities using the method of fundamental solutions*, Inverse. Probl. Sci. En., to be appeared.

[19] A. Karageorghis and Y-S. Smyrlis, *Matrix decomposition algorithms related to the MFS for axisymmetric problems*, in: Recent Advances in Boundary Element Methods, eds. G. D. Manolis and D. Polyzos, Springer, New York, pp. 223–237, 2009.

[20] M. Katsurada, *A mathematical study of the charge simulation method II*, J. Fac. Sci., Univ. of Tokyo, Sect. 1A, Math., 36 (1989), pp. 135–162, .

[21] M. Katsurada and H. Okamoto, *A mathematical study of the charge simulation method I*, Journal of the Faculty of Science. University of Tokyo. Section IA. Mathematics, 35 (1988), pp. 507–518.

[22] L. MARIN, *A meshless method for solving the Cauchy problem in three-dimensional elastostatics*, Comput. Math. Appl., 50 (2005), pp. 73–92.

[23] L. MARIN AND D. LESNIC, *The method of fundmental solutions for the Cauchy problem associated with two-dimensional Helmholtz-type equations*, Comput. Struct., 83 (2005), pp. 267–278.

[24] L. MARIN AND D. LESNIC, *The method of fundamental solutions for inverse boundary value problems associated with the two-dimensional biharmonic equation*, Math. Comput. Model., 42 (2005), pp. 261–278.

[25] R. MATHON AND R. L. JOHNSTON, *The approximate solution of elliptic boundary-value problems by fundamental solutions*, SIAM J. Numer. Anal., 14 (1977), pp. 638–650.

[26] Numerical Algorithms Group Library Mark 21, NAG(UK) Ltd, Wilkinson House, Jordan Hill Road, Oxford, UK (2007).

[27] A. POULLIKKAS, A. KARAGEORGHIS, G. GEORGIOU AND J. ASCOUGH, *The method of fundamental solutions for Stokes flows with a free surface*, Numer. Meth. Part. D. E., 14 (1998), pp. 667–678.

[28] W. H. PRESS, B. P. FLANNERY, S. A. TEUKOLSKY AND W. T. VETTERLING, Numerical Recipes: The Art of Scientific Computing (Fortran version), Cambridge University Press, Cambridge, 1989.

[29] R. SCHABACK, *Adaptive numerical solution of MFS systems*, in: The Method of Fundamental Solutions - A Meshless Method, eds. C.S. Chen, A Karageorghis and Y.S. Smyrlis, Dynamic Publishers, Inc., pp. 1–27, 2008.

[30] Y-S. SMYRLIS AND A. KARAGEORGHIS, *Numerical analysis of the MFS for certain harmonic problems*, Modélisation Mathématique et Analyse Numérique, 38 (2004), pp. 495–517.

[31] R. TANKELEVICH, G. FAIRWEATHER, A. KARAGEORGHIS AND Y-S. SMYRLIS, *Potential field based geometric modeling using the method of fundamental solutions*, Inter. J. Numer. Meth. Eng., 68 (2006), pp. 1257–1280.

[32] Y. WANG AND Y. RUDY, *Application of the method of fundamental solutions to potential-based inverse electrocardiography*, Ann. Biomed. Eng., 34 (2006), pp. 1272–1288.