Numer Algor (2018) 79:399–421
https://doi.org/10.1007/s11075-017-0443-5

ORIGINAL PAPER

Kansa-RBF algorithms for elliptic problems in regular polygonal domains

Andreas Karageorghis[1] ·
Malgorzata A. Jankowska[2] · **C. S. Chen**[3,4]

Received: 20 September 2017 / Accepted: 6 November 2017 / Published online: 6 December 2017

Abstract We propose matrix decomposition algorithms for the efficient solution of the linear systems arising from Kansa radial basis function discretizations of elliptic boundary value problems in regular polygonal domains. These algorithms exploit the symmetry of the domains of the problems under consideration which lead to coefficient matrices possessing block circulant structures. In particular, we consider the Poisson equation, the inhomogeneous biharmonic equation, and the inhomogeneous Cauchy-Navier equations of elasticity. Numerical examples demonstrating the applicability of the proposed algorithms are presented.

Keywords Kansa method · Radial basis functions · Poisson equation · Biharmonic equation · Cauchy-Navier equations · Matrix decomposition algorithms

✉ Andreas Karageorghis
andreask@ucy.ac.cy

Malgorzata A. Jankowska
malgorzata.jankowska@put.poznan.pl

C. S. Chen
cschen.math@gmail.com

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

[2] Institute of Applied Mechanics, Faculty of Mechanical Engineering and Management, Poznan University of Technology, 60-965 Poznan, Poland

[3] Department of Mathematics, University of Southern Mississippi, Hattiesburg, MS 39406, USA

[4] School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, China

Mathematics Subject Classification (2010) Primary 65N35 · Secondary 65N22

1 Introduction

Radial basis function (RBF) methods are meshless methods which, due to their simplicity and accuracy, have recently become very popular for the solution of a large variety of boundary value problems; see, e.g. [7, Chapters 38-45]. The simplest and most popular such methods are RBF collocation methods [3] and, in particular, the so-called Kansa method [16]. One of the disadvantages of RBF collocation methods is the fact that the resulting coefficient matrix in the discretization is dense and poorly conditioned. This means that RBF collocation methods are not suitable for the solution of problems where large numbers of degrees of freedom are required. These difficulties may be alleviated by using the local Kansa-RBF method [23, 25] which, in contrast to the global Kansa-RBF method, leads to sparse systems. However, as is the case with other local methods such as the finite element and finite difference methods, the accuracy of the local Kansa-RBF method is considerably lower than that of the global Kansa-RBF method due to the fact that only a small number of neighbouring points are used in the approximation process. Furthermore, for a large number of collocation points, the search for the neighbouring points becomes very time consuming.

In the past, various approaches have been proposed to tackle large-scale problems in which large numbers of interior and boundary collocation points are involved. For example, Greengard and Rokhlin [11] proposed the fast multipole method for handling very large linear systems and speeding up the computational process. Another very popular approach is the domain decomposition method (DDM) in which the domain of the problem in question is split into subdomains, each of which contains an almost equal number of points. At the interfaces of these subdomains, appropriate continuity conditions are imposed while on the boundaries of the subdomains which coincide with the physical boundary of the problem, the physical boundary conditions are imposed. This approach which reduces the problem to solving a sequence of smaller problems has been particularly effective; see, for example, [10, 15, 17, 24, 26]. Instead of decomposing the domain into smaller ones as is done in the DDM, there is another well-established numerical technique in which the (large) global system is decomposed into a series of smaller systems. This, as will be demonstrated in this work, leads to considerable savings in both computational time and storage and, in addition, alleviates to a great extent the ill-conditioning issue. In this work, we shall consider the discretization of elliptic boundary value problems in regular polygonal domains using a Kansa-RBF method. For any choice of RBF, for an appropriate choice of collocation points, such discretizations lead to linear systems in which the coefficient matrices possess block circulant structures and which are solved efficiently using matrix decomposition algorithms (MDAs) [1] in conjunction with fast Fourier transforms (FFTs). Such MDAs have been used in the past for Kansa-RBF discretizations of problems possessing radial symmetry in two and three dimensions in [20] and [28], respectively, as well as in the approximation of functions using RBFs [14, 21, 22]. The ideas developed in this paper are extensions of the ideas developed in [19] for the method of fundamental solutions.

An additional difficulty with Kansa-RBF collocation methods is the selection of an appropriate value of shape parameter of RBFs leading to optimal accuracy. Although many algorithms have been proposed for the selection of such a shape parameter [4, 8, 9, 12, 18, 30], this remains an open issue. In the context of MDAs, the leave-one-out cross validation (LOOCV) algorithm [8, 28, 30] and the average distance functions [6, 9, 12] have been modified and recently proposed for the selection of an appropriate value of the shape parameter; see [20, 27, 28] for further details. In this study, we shall propose another simple method for obtaining a value of the shape parameter that can produce good accuracy, using the normalized multiquadric (MQ).

In Section 2, we describe the Kansa-RBF method for the solution of Poisson problems in regular polygonal domains and outline the proposed MDA for the solution of the resulting linear system. The application of the method and MDA to inhomogeneous biharmonic problems is presented in Section 3. In Section 4, a description of the method and related MDA for inhomogeneous Cauchy-Navier problems are provided. Finally, in Section 6, we present some conclusions and ideas about future work.

2 The Poisson equation

We consider the Poisson equation in $\mathbb{R}^2$

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y) \quad \text{in} \quad \Omega, \tag{2.1a}$$

subject to the Dirichlet boundary conditions

$$u = g_1(x, y) \quad \text{on} \quad \partial\Omega_1, \tag{2.1b}$$

$$u = g_2(x, y) \quad \text{on} \quad \partial\Omega_2, \tag{2.1c}$$

or the mixed Neumann/Dirichlet boundary conditions

$$\frac{\partial u}{\partial \mathrm{n}} = g_1(x, y) \quad \text{on} \quad \partial\Omega_1, \tag{2.1d}$$

$$u = g_2(x, y) \quad \text{on} \quad \partial\Omega_2, \tag{2.1e}$$

where the domain Ω is the region between two concentric regular N-gons. The boundary $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2$, $\partial\Omega_1 \cap \partial\Omega_2 = \emptyset$ where $\partial\Omega_1$ and $\partial\Omega_2$ are regular N-gons. A typical domain considered is presented in Fig. 1a. Note that the application of the methods presented in this paper to regular polygonal domains without a hole is trivial. In (2.1d), $\partial/\partial\mathrm{n}$ denotes the derivative along the outward unit normal vector to the boundary which we denote by $\mathbf{n} = (\mathrm{n}_x, \mathrm{n}_y)$. Problem (2.1a), (2.1b)–(2.1c) is a *Dirichlet boundary value problem*, whereas problem (2.1a), (2.1d)–(2.1e) is a *mixed Neumann/Dirichlet boundary value problem*.

2.1 Kansa-RBF method

Let us assume that the radius of the circle circumscribing the inner regular polygon $\partial\Omega_1$ is ϱ_1 and that the radius of the circle circumscribing the outer regular polygon

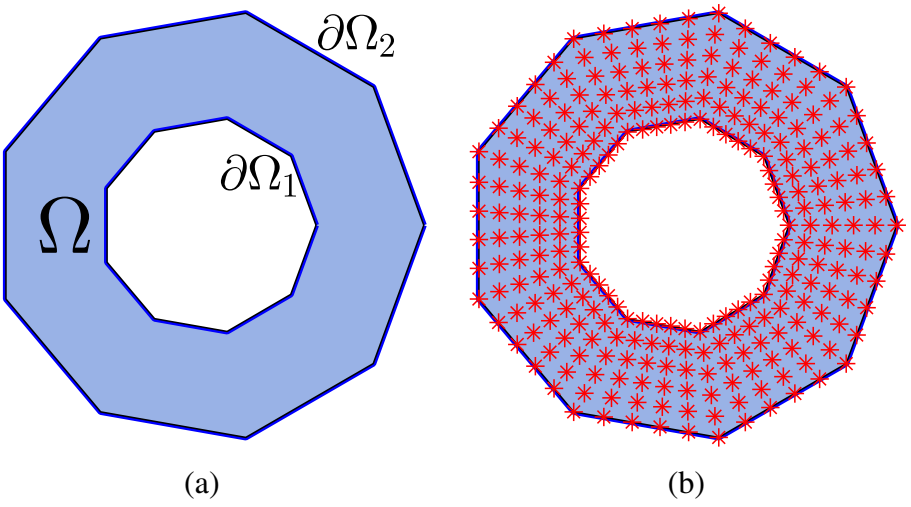


Fig. 1 **a** Typical geometry considered. **b** Typical Kansa-RBF discretization

$\partial\Omega_2$ is ϱ_2. We shall define K similar concentric N-gons in $\overline{\Omega}$, and on each of the N sides of each of these polygons, we shall place M collocation points. This will result in a total of MNK collocation points.

We first define the M angles

$$\vartheta_m = \frac{2\pi(m-1)}{MN}, \quad m = 1, \ldots, M, \tag{2.2}$$

and the K radii

$$\mathsf{r}_k = \varrho_1 + (\varrho_2 - \varrho_1)\,\frac{k-1}{K-1}, \quad k = 1, \ldots, K. \tag{2.3}$$

The collocation points $\{(x_{nmk}, y_{nmk})\}_{n=1,m=1,k=1}^{N,M,K}$ are then defined as follows:

$$x_{nmk} = \mathsf{r}_k \frac{\cos\left(\frac{\pi}{N}\right)\cos\left(\vartheta_m + \frac{2\pi}{N}(n-1)\right)}{\cos\left(\frac{\pi}{N} - \vartheta_m\right)}, \quad y_{nmk} = \mathsf{r}_k \frac{\cos\left(\frac{\pi}{N}\right)\sin\left(\vartheta_m + \frac{2\pi}{N}(n-1)\right)}{\cos\left(\frac{\pi}{N} - \vartheta_m\right)}, \tag{2.4}$$

for $n = 1, \ldots, N$, $m = 1, \ldots, M$, $k = 1, \ldots, K$. A typical distribution of collocation points is depicted in Fig. 1b.

We approximate the solution of boundary value problem (2.1) by

$$u(x, y) \simeq \hat{u}(x, y) = \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K} a_{nmk}\phi_{nmk}(x, y), \qquad (x, y) \in \overline{\Omega}, \tag{2.5}$$

where the ϕ_{nmk} are RBFs defined by

$$\phi_{nmk}(x, y) = \Phi(r_{nmk}), \quad \text{where} \quad r_{nmk}^2 = (x - x_{nmk})^2 + (y - y_{nmk})^2. \tag{2.6}$$

The unknown coefficients $\{(a_{nmk})\}_{n=1,m=1,k=1}^{N,M,K}$ are determined by collocating the differential equation (2.1a) and the boundary conditions (2.1b)–(2.1c) or (2.1d)–(2.1e) as follows:

$$\begin{aligned}
\Delta\hat{u}(x_{nmk}, y_{nmk}) &= f(x_{nmk}, y_{nmk}),\ n = 1, \ldots, N,\ m = 1, \ldots, M,\ k = 2, \ldots, K-1,\\
\hat{u}(x_{nm1}, y_{nm1}) &= g_1(x_{nm1}, y_{nm1}),\ n = 1, \ldots, N,\ m = 1, \ldots, M,\\
\text{or}\quad \frac{\partial\hat{u}}{\partial \mathrm{n}}(x_{nm1}, y_{nm1}) &= g_1(x_{nm1}, y_{nm1}),\ n = 1, \ldots, N,\ m = 1, \ldots, M,\\
\text{and}\quad \hat{u}(x_{nmK}, y_{nmK}) &= g_2(x_{nmK}, y_{nmK}),\ n = 1, \ldots, N,\ m = 1, \ldots, M.
\end{aligned} \tag{2.7}$$

By vectorizing the arrays of unknown coefficients and collocation points from

$$\mathsf{a}_{(k-1)MN+(m-1)N+n} = a_{nmk},\quad \mathsf{x}_{(k-1)MN+(m-1)N+n} = x_{nmk},\quad \mathsf{y}_{(k-1)MN+(m-1)N+n} = y_{nmk}, \tag{2.8}$$

for $n = 1, \ldots, N,\ m = 1, \ldots, M,\ k = 1, \ldots, K$, (2.7) yield a system of the form

$$A\,\boldsymbol{a} = \begin{pmatrix} A_{1,1} & A_{1,2} & \ldots & A_{1,L} \\ A_{2,1} & A_{2,2} & \ldots & A_{2,L} \\ \vdots & \vdots & \ddots & \vdots \\ A_{L,1} & A_{L,2} & \ldots & A_{L,L} \end{pmatrix} \begin{pmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \\ \vdots \\ \boldsymbol{a}_L \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_L \end{pmatrix} = \boldsymbol{b}\,, \tag{2.9}$$

where $L = MK$.

The $N \times N$ submatrices $A_{\ell_1,\ell_2},\ \ell_1, \ell_2 = 1, \ldots, L$, are defined as follows:

$$\left(A_{m_1,(k_2-1)M+m_2}\right)_{n_1,n_2} = \phi_{n_2,m_2,k_2}(x_{n_1,m_1,1}, y_{n_1,m_1,1}), \tag{2.10a}$$

or

$$\left(A_{m_1,(k_2-1)M+m_2}\right)_{n_1,n_2} = \frac{\partial\phi_{n_2,m_2,k_2}}{\partial \mathrm{n}}(x_{n_1,m_1,1}, y_{n_1,m_1,1}), \tag{2.10b}$$

and

$$\left(A_{(K-1)M+m_1,(k_2-1)M+m_2}\right)_{n_1,n_2} = \phi_{n_2,m_2,k_2}(x_{n_1,m_1,K}, y_{n_1,m_1,K}), \tag{2.10c}$$

for $n_1, n_2 = 1, \ldots, N,\ m_1, m_2 = 1, \ldots, M,\ k_2 = 1, \ldots, K$, and

$$\left(A_{(k_1-1)M+m_1,(k_2-1)M+m_2}\right)_{n_1,n_2} = \Delta\phi_{n_2,m_2,k_2}(x_{n_1,m_1,k_1}, y_{n_1,m_1,k_1}), \tag{2.10d}$$

for $n_1, n_2 = 1, \ldots, N,\ m_1, m_2 = 1, \ldots, M,\ k_1 = 2, \ldots, K-1,\ k_2 = 1, \ldots, K$, while the $N \times 1$ vectors $\boldsymbol{a}_\ell, \boldsymbol{b}_\ell, \ell = 1, \ldots, L$, are defined as

$$\left(\boldsymbol{a}_{(k-1)M+m}\right)_n = a_{nmk},\ n = 1, \ldots, N,\ m = 1, \ldots, M,\ k = 1, \ldots, K,$$

$$\left(\boldsymbol{b}_{(k-1)M+m}\right)_n = f(x_{nmk}, y_{nmk}),\ n = 1, \ldots, N,\ m = 1, \ldots, M,\ k = 2, \ldots, K-1,$$

$$(\boldsymbol{b}_m)_n = g_1(x_{nm1}, y_{nm1}),\quad \left(\boldsymbol{b}_{(K-1)M+m}\right)_n = g_2(x_{nmK}, y_{nmK}),\quad n = 1, \ldots, N,\ m = 1, \ldots, M.$$

Each of the $N \times N$ submatrices $A_{\ell_1,\ell_2},\ \ell_1, \ell_2 = 1, \ldots, L$, in the coefficient matrix in (2.9) is circulant [5]. Hence, matrix A in system (2.9) is *block circulant*.

2.2 Matrix decomposition algorithm

We shall be using an MDA similar to the one proposed in [28, Section2.3] as well as properties of circulant matrices [5].

First, we define the unitary $N \times N$ Fourier matrix

$$U_N = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \bar{\omega} & \bar{\omega}^2 & \cdots & \bar{\omega}^{N-1} \\ 1 & \bar{\omega}^2 & \bar{\omega}^4 & \cdots & \bar{\omega}^{2(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \bar{\omega}^{N-1} & \bar{\omega}^{2(N-1)} & \cdots & \bar{\omega}^{(N-1)(N-1)} \end{pmatrix}, \quad \text{where} \quad \omega = \mathrm{e}^{2\pi \mathrm{i}/N},\ \mathrm{i}^2 = -1. \tag{2.11}$$

If I_L is the $L \times L$ identity matrix, pre-multiplication of system (2.9) by $I_L \otimes U_N$ yields

$$(I_L \otimes U_N)\, A \left(I_L \otimes U_N^*\right) (I_L \otimes U_N)\, \boldsymbol{a} = (I_L \otimes U_N)\, \boldsymbol{b} \quad \text{or} \quad \tilde{A}\tilde{\boldsymbol{a}} = \tilde{\boldsymbol{b}}, \tag{2.12}$$

where

$$\tilde{A} = (I_L \otimes U_N)\, A \left(I_L \otimes U_N^*\right) = \begin{pmatrix} D_{1,1} & D_{1,2} & \cdots & D_{1,L} \\ D_{2,1} & D_{2,2} & \cdots & D_{2,L} \\ \vdots & \vdots & & \vdots \\ D_{L,1} & D_{L,2} & \cdots & D_{L,L} \end{pmatrix}, \tag{2.13}$$

with $D_{\ell_1,\ell_2} = U_N A_{\ell_1,\ell_2} U_N^*,\ \ell_1, \ell_2 = 1, \ldots, L$, and

$$\tilde{\boldsymbol{a}} = (I_L \otimes U_N)\, \boldsymbol{a} = \begin{pmatrix} \tilde{\boldsymbol{a}}_1 \\ \tilde{\boldsymbol{a}}_2 \\ \vdots \\ \tilde{\boldsymbol{a}}_L \end{pmatrix}, \quad \tilde{\boldsymbol{b}} = (I_L \otimes U_N)\, \boldsymbol{b} = \begin{pmatrix} \tilde{\boldsymbol{b}}_1 \\ \tilde{\boldsymbol{b}}_2 \\ \vdots \\ \tilde{\boldsymbol{b}}_L \end{pmatrix}, \tag{2.14}$$

with $\tilde{\boldsymbol{a}}_\ell = U_N \boldsymbol{a}_\ell,\ \tilde{\boldsymbol{b}}_\ell = U_N \boldsymbol{b}_\ell,\ \ell = 1, \ldots, L$. From the properties of circulant matrices [5], each of the $N \times N$ matrices $D_{\ell_1,\ell_2},\ \ell_1, \ell_2 = 1, \ldots, L$, is diagonal. If, in particular

$$D_{\ell_1,\ell_2} = \operatorname{diag}\left(D_{\ell_1,\ell_{21}}, D_{\ell_1,\ell_{22}}, \ldots, D_{\ell_1,\ell_{2N}}\right) \quad \text{and} \quad A_{\ell_1,\ell_2} = \operatorname{circ}\left(A_{\ell_1,\ell_{21}}, A_{\ell_1,\ell_{22}} \ldots, A_{\ell_1,\ell_{2N}}\right),$$

we have, for $\ell_1, \ell_2 = 1, \ldots, L$,

$$D_{\ell_1,\ell_{2n}} = \sum_{k=1}^{N} A_{\ell_1,\ell_{2k}} \omega^{(k-1)(n-1)},\ n = 1, \ldots, N. \tag{2.15}$$

Since the matrix $\tilde{A}$ consists of L^2 blocks of order N, each of which is diagonal, the solution of system (2.12) can be decomposed into solving the N independent systems of order L

$$E_n \boldsymbol{x}_n = \boldsymbol{y}_n, \quad n = 1, \ldots, N, \tag{2.16}$$

where

$$(E_n)_{\ell_1,\ell_2} = D_{\ell_1,\ell_{2_n}},\ \ell_1,\ell_2 = 1,\ldots,L,\ \text{and}\ (\boldsymbol{x}_n)_\ell = (\tilde{\boldsymbol{a}}_\ell)_n,\ \left(\boldsymbol{y}_n\right)_\ell = \left(\tilde{\boldsymbol{b}}_\ell\right)_n,\ \ell = 1,\ldots,L. \tag{2.17}$$

Having obtained the vectors $\boldsymbol{x}_n,\ n = 1,\ldots,L$, we can recover the vectors $\tilde{\boldsymbol{a}}_\ell,\ \ell = 1,\ldots,L$, and, subsequently, the vector $\boldsymbol{a}$ from (2.14), i.e.,

$$\boldsymbol{a} = \begin{pmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \\ \vdots \\ \boldsymbol{a}_L \end{pmatrix} = \left(I_L \otimes U_N^*\right)\tilde{\boldsymbol{a}} = \begin{pmatrix} U_N^*\tilde{\boldsymbol{a}}_1 \\ U_N^*\tilde{\boldsymbol{a}}_2 \\ \vdots \\ U_N^*\tilde{\boldsymbol{a}}_L \end{pmatrix}. \tag{2.18}$$

In conclusion, the MDA can be summarized as follows:

Algorithm 1

Step 1: `Compute` $\tilde{\boldsymbol{b}}_\ell = U_N\boldsymbol{b}_\ell, \ell = 1,\ldots,L$.
Step 2: `Construct the diagonal matrices` D_{ℓ_1,ℓ_2} `from (2.15).`
Step 3: `Solve the` N`,` $L\times L$ `systems (2.16) to obtain the` $\{\boldsymbol{x}_n\}_{n=1}^N$`, and subsequently the` $\{\tilde{\boldsymbol{a}}_\ell\}_{\ell=1}^L$ `from (2.17).`
Step 4: `Recover the vector of coefficients` $\boldsymbol{a}$ `from (2.18).`

In Steps 1, 2, and 4, FFTs are used while the most expensive part of the algorithm is the solution of N linear systems, each of order L. The FFTs are carried out using the MATLAB© [29] commands `fft` and `ifft`. The dominant cost of the algorithm is thus $O(NL^3)$. Note that we have also implemented the slightly different approach proposed in [14, Appendix] where, in order to solve system (2.12), the sparse matrix $\tilde{A}$ in (2.13) is reordered as a block diagonal matrix.

Moreover, from the description of the algorithm, it is obvious that the larger N, i.e., the more sides the regular polygonal domain has, the more the savings that can be achieved. This is because for larger N, we can take a smaller M in (2.2) and thus L will be smaller.

3 The biharmonic equation

3.1 The problem

We next consider the biharmonic equation in $\mathbb{R}^2$

$$\Delta^2 u = \frac{\partial^4 u}{\partial x^4} + 2\frac{\partial^4 u}{\partial x^2\partial y^2} + \frac{\partial^4 u}{\partial y^4} = f(x,y) \quad \text{in} \quad \Omega, \tag{3.1a}$$

subject to the boundary conditions

$$u = g_1(x, y) \quad \text{and} \quad \frac{\partial u}{\partial \mathrm{n}} = h_1(x, y) \quad \text{on} \quad \partial\Omega_1, \tag{3.1b}$$

and

$$u = g_2(x, y) \quad \text{and} \quad \frac{\partial u}{\partial \mathrm{n}} = h_2(x, y) \quad \text{on} \quad \partial\Omega_2, \tag{3.1c}$$

or the boundary conditions

$$u = g_1(x, y) \quad \text{and} \quad \Delta u = h_1(x, y) \quad \text{on} \quad \partial\Omega_1, \tag{3.1d}$$

and

$$u = g_2(x, y) \quad \text{and} \quad \Delta u = h_2(x, y) \quad \text{on} \quad \partial\Omega_2, \tag{3.1e}$$

where the domain Ω is the region between two concentric regular N-gons as in (2.1). Boundary problem (3.1a), (3.1b)–(3.1c) is known as the *first biharmonic problem* whereas boundary problem (3.1a), (3.1d)–(3.1e) is known as the *second biharmonic problem*.

3.2 Kansa-RBF method

The collocation points are defined from (2.4) as in Section 2.1. The approximation of the solution of boundary value problem (3.1) is given by (2.5).

The unknown coefficients $\{(a_{nmk})\}_{n=1,m=1,k=1}^{N,M,K}$ are determined by collocating the differential equation (3.1a) and the boundary conditions (3.1b)–(3.1c) or (3.1d)–(3.1e) as follows:

$$\Delta^2 \hat{u}(x_{nmk}, y_{nmk}) = f(x_{nmk}, y_{nmk}),\ n = 1, \ldots, N,\ m = 1, \ldots, M,\ k = 3, \ldots, K-2,$$

$$\hat{u}(x_{nm1}, y_{nm1}) = g_1(x_{nm1}, y_{nm1}),\ \frac{\partial \hat{u}}{\partial \mathrm{n}}(x_{nm1}, y_{nm1}) = h_1(x_{nm1}, y_{nm1}),$$

and

$$\hat{u}(x_{nmK}, y_{nmK}) = g_2(x_{nmK}, y_{nmK}),\ \frac{\partial \hat{u}}{\partial \mathrm{n}}(x_{nmK}, y_{nmK}) = h_2(x_{nmK}, y_{nmK}),$$

or

$$\hat{u}(x_{nm1}, y_{nm1}) = g_1(x_{nm1}, y_{nm1}),\ \Delta\hat{u}(x_{nm1}, y_{nm1}) = h_1(x_{nm1}, y_{nm1}),$$

and

$$\hat{u}(x_{nmK}, y_{nmK}) = g_2(x_{nmK}, y_{nmK}),\ \Delta\hat{u}(x_{nmK}, y_{nmK}) = h_2(x_{nmK}, y_{nmK}), \tag{3.2}$$

for $n = 1, \ldots, N,\ m = 1, \ldots, M$.

By vectorizing the arrays of unknown coefficients and collocation points as in (2.8), we obtain a system of the form (2.9) where the $N \times N$ submatrices $A_{\ell_1,\ell_2},\ \ell_1, \ell_2 = 1, \ldots, L$, are defined as follows:

$$\left(A_{m_1,(k_2-1)M+m_2}\right)_{n_1,n_2} = \phi_{n_2,m_2,k_2}(x_{n_1,m_1,1}, y_{n_1,m_1,1}), \tag{3.3a}$$

and

$$\left(A_{M+m_1,(k_2-1)M+m_2}\right)_{n_1,n_2} = \frac{\partial \phi_{n_2,m_2,k_2}}{\partial \mathrm{n}}(x_{n_1,m_1,1}, y_{n_1,m_1,1}), \tag{3.3b}$$

or

$$\left(A_{M+m_1,(k_2-1)M+m_2}\right)_{n_1,n_2} = \Delta\phi_{n_2,m_2,k_2}(x_{n_1,m_1,1}, y_{n_1,m_1,1}), \tag{3.3c}$$

and

$$\left(A_{(K-1)M+m_1,(k_2-1)M+m_2}\right)_{n_1,n_2} = \phi_{n_2,m_2,k_2}(x_{n_1,m_1,K}, y_{n_1,m_1,K}), \tag{3.3d}$$

and

$$\left(A_{(K-2)M+m_1,(k_2-1)M+m_2}\right)_{n_1,n_2} = \frac{\partial\phi_{n_2,m_2,k_2}}{\partial \mathrm{n}}(x_{n_1,m_1,K}, y_{n_1,m_1,K}), \tag{3.3e}$$

or

$$\left(A_{(K-2)M+m_1,(k_2-1)M+m_2}\right)_{n_1,n_2} = \Delta\phi_{n_2,m_2,k_2}(x_{n_1,m_1,K}, y_{n_1,m_1,K}), \tag{3.3f}$$

$$\text{for} \quad n_1, n_2 = 1, \ldots, N,\ m_1, m_2 = 1, \ldots, M,\ k_2 = 1, \ldots, K,$$

and

$$\left(A_{(k_1-1)M+m_1,(k_2-1)M+m_2}\right)_{n_1,n_2} = \Delta^2\phi_{n_2,m_2,k_2}(x_{n_1,m_1,k_1}, y_{n_1,m_1,k_1}), \tag{3.3g}$$

for $n_1, n_2 = 1, \ldots, N,\ m_1, m_2 = 1, \ldots, M,\ k_1 = 2, \ldots, K-1,\ k_2 = 1, \ldots, K,$ while the $N \times 1$ vectors $\boldsymbol{a}_\ell, \boldsymbol{b}_\ell, \ell = 1, \ldots, L$ are defined as

$$\left(\boldsymbol{a}_{(k-1)M+m}\right)_n = a_{nmk},\ n = 1, \ldots, N,\ m = 1, \ldots, M,\ k = 1, \ldots, K,$$

$$\left(\boldsymbol{b}_{(k-1)M+m}\right)_n = f(x_{nmk}, y_{nmk}),\ n = 1, \ldots, N,\ m = 1, \ldots, M,\ k = 3, \ldots, K-2,$$

$$(\boldsymbol{b}_m)_n = g_1(x_{nm1}, y_{nm1}), \quad (\boldsymbol{b}_{M+m})_n = h_1(x_{nm1}, y_{nm1}), \quad n = 1, \ldots, N,\ m = 1, \ldots, M,$$

$$\left(\boldsymbol{b}_{(K-1)M+m}\right)_n = g_2(x_{nmK}, y_{nmK}), \quad \left(\boldsymbol{b}_{(K-2)M+m}\right)_n = h_2(x_{nmK}, y_{nmK}), \quad n = 1, \ldots, N,\ m = 1, \ldots, M.$$

As in the case of the Poisson equation in Section 2.1, each of the $L \times L$ submatrices A_{n_1,n_2}, $n_1, n_2 = 1, \ldots, L$, is circulant, and hence, the matrix A is again block circulant. The resulting system can therefore be solved efficiently using the MDA (Algorithm 1) described in Section 2.2.

4 The Cauchy-Navier equations of elasticity

We finally consider the Cauchy-Navier system in $\mathbb{R}^2$ for the displacements (u_1, u_2) in the form (see, e.g. [13])

$$\begin{cases} \mathcal{L}_1(u_1, u_2) \equiv \mathcal{L}_{11}u_1 + \mathcal{L}_{12}u_2 \equiv \mu\Delta u_1 + \dfrac{\mu}{1-2\nu}\left(\dfrac{\partial^2 u_1}{\partial x^2} + \dfrac{\partial^2 u_2}{\partial x \partial y}\right) = f_1, \\ \mathcal{L}_2(u_1, u_2) \equiv \mathcal{L}_{21}u_1 + \mathcal{L}_{22}u_2 \equiv \dfrac{\mu}{1-2\nu}\left(\dfrac{\partial^2 u_1}{\partial x \partial y} + \dfrac{\partial^2 u_2}{\partial y^2}\right) + \mu\Delta u_2 = f_2, \end{cases} \quad \text{in} \quad \Omega, \tag{4.1a}$$

subject to the Dirichlet boundary conditions

$$u_1 = g_1 \quad \text{and} \quad u_2 = h_1 \quad \text{on} \quad \partial\Omega_1, \tag{4.1b}$$

and

$$u_1 = g_2 \quad \text{and} \quad u_2 = h_2 \quad \text{on} \quad \partial\Omega_2, \tag{4.1c}$$

or the mixed Neumann/Dirichlet boundary conditions

$$t_1 = g_1 \quad \text{and} \quad t_2 = h_1 \quad \text{on} \quad \partial\Omega_1, \tag{4.1d}$$

and

$$u_1 = g_2 \quad \text{and} \quad u_2 = h_2 \quad \text{on} \quad \partial\Omega_2, \tag{4.1e}$$

where the domain Ω is the region between two concentric regular N-gons as in (2.1). In (4.1a), the constant $\nu \in [0, 1/2)$ is Poisson's ratio and $\mu > 0$ is the shear modulus. Also, in (4.1a), the operators $\mathcal{L}_{11}$, $\mathcal{L}_{12}$, $\mathcal{L}_{21}$, and $\mathcal{L}_{22}$ are defined by

$$\mathcal{L}_{11} \equiv \mu\Delta + \frac{\mu}{1-2\nu}\frac{\partial^2}{\partial x^2}, \quad \mathcal{L}_{12} \equiv \frac{\mu}{1-2\nu}\frac{\partial^2}{\partial x \partial y}, \quad \mathcal{L}_{21} \equiv \mathcal{L}_{12}, \quad \mathcal{L}_{22} \equiv \mu\Delta + \frac{\mu}{1-2\nu}\frac{\partial^2}{\partial y^2}.$$

In (4.1d), t_1 and t_2 are the tractions [13] defined by

$$t_1 = 2\mu\left[\left(\frac{1-\nu}{1-2\nu}\right)\frac{\partial u_1}{\partial x} + \left(\frac{\nu}{1-2\nu}\right)\frac{\partial u_2}{\partial y}\right]\mathrm{n}_x + \mu\left[\frac{\partial u_1}{\partial y} + \frac{\partial u_2}{\partial x}\right]\mathrm{n}_y,$$

$$t_2 = \mu\left[\frac{\partial u_1}{\partial y} + \frac{\partial u_2}{\partial x}\right]\mathrm{n}_x + 2\mu\left[\left(\frac{\nu}{1-2\nu}\right)\frac{\partial u_1}{\partial x} + \left(\frac{1-\nu}{1-2\nu}\right)\frac{\partial u_2}{\partial y}\right]\mathrm{n}_y.$$

Problem (4.1a), (4.1b)–(4.1c) is a *Dirichlet boundary value problem* whereas problem (4.1a), (4.1d)–(4.1e) is a *mixed Neumann/Dirichlet boundary value problem.*

4.1 Kansa-RBF method

The collocation points are defined from (2.4) as in Section 2.1.

We approximate the solution (u_1, u_2) of boundary value problem (4.1) by

$$\hat{u}^{(\ell)}(x, y) = \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K} a_{nmk}^{(\ell)}\phi_{nmk}(x, y), \quad \ell = 1, 2, \qquad (x, y) \in \overline{\Omega}. \tag{4.2}$$

The unknown coefficients $\left\{\left(a_{nmk}^{(\ell)}\right)\right\}_{n=1,m=1,k=1}^{N,M,K}$, $\ell = 1, 2$, are determined by collocating the differential equations (4.1a) and the boundary conditions (4.1b)–(4.1c) or (4.1d)–(4.1e) as follows:

$$\mathcal{L}_\ell(\hat{u}^{(1)}, \hat{u}^{(2)})(x_{nmk}, y_{nmk}) = f_\ell(x_{nmk}, y_{nmk}), \ \ell = 1, 2,$$

$$n = 1, \ldots, N, \ m = 1, \ldots, M, \ k = 2, \ldots, K-1,$$

$$\hat{u}^{(1)}(x_{nm1}, y_{nm1}) = g_1(x_{nm1}, y_{nm1}), \ \hat{u}^{(2)}(x_{nm1}, y_{nm1}) = h_1(x_{nm1}, y_{nm1}),$$

or

$$t_{NMK}^{(1)}(x_{nm1}, y_{nm1}) = g_1(x_{nm1}, y_{nm1}), \ t_{NMK}^{(2)}(x_{nm1}, y_{nm1}) = h_1(x_{nm1}, y_{nm1}),$$

and

$$\hat{u}^{(1)}(x_{nmK}, y_{nmK}) = g_2(x_{nmK}, y_{nmK}), \ \hat{u}^{(2)}(x_{nmK}, y_{nmK}) = h_2(x_{nmK}, y_{nmK}),$$

$$n = 1, \ldots, N,\ m = 1, \ldots, M. \tag{4.3}$$

By vectorizing the arrays of unknown coefficients and collocation points from (2.8), (4.3) yield a $2MNK \times 2MNK$ system of the form (2.9). The $2N \times 2N$ submatrices $A_{\ell_1,\ell_2},\ \ell_1, \ell_2 = 1, \ldots, L$,are now defined as follows: (Note that we are now defining the matrix and vector elements in (2.9) as 2×2 and 2×1 *arrays*, respectively.)

$$\left(A_{m_1,(k_2-1)M+m_2}\right)_{n_1,n_2} = \begin{pmatrix} \phi_{n_2,m_2,k_2}(x_{n_1,m_1,1},\, y_{n_1,m_1,1}) & 0 \\ 0 & \phi_{n_2,m_2,k_2}(x_{n_1,m_1,1},\, y_{n_1,m_1,1}) \end{pmatrix}, \tag{4.4a}$$

or

$$\left(A_{m_1,(k_2-1)M+m_2}\right)_{n_1,n_2} \tag{4.4b}$$
$$= \mu \begin{pmatrix} 2\left(\frac{1-\nu}{1-2\nu}\right)\frac{\partial \phi_{n_2,m_2,k_2}}{\partial x}\mathrm{n}_x + \frac{\partial \phi_{n_2,m_2,k_2}}{\partial y}\mathrm{n}_y & \left(\frac{2\nu}{1-2\nu}\right)\frac{\partial \phi_{n_2,m_2,k_2}}{\partial y}\mathrm{n}_x + \frac{\partial \phi_{n_2,m_2,k_2}}{\partial x}\mathrm{n}_y \\ \frac{\partial \phi_{n_2,m_2,k_2}}{\partial y}\mathrm{n}_x + \left(\frac{2\nu}{1-2\nu}\right)\frac{\partial \phi_{n_2,m_2,k_2}}{\partial x}\mathrm{n}_y & \frac{\partial \phi_{n_2,m_2,k_2}}{\partial x}\mathrm{n}_x + 2\left(\frac{1-\nu}{1-2\nu}\right)\frac{\partial \phi_{n_2,m_2,k_2}}{\partial y}\mathrm{n}_y \end{pmatrix},$$

evaluated at the point $(x_{n_1,m_1,1},\, y_{n_1,m_1,1})$, and

$$\left(A_{(K-1)M+m_1,(k_2-1)M+m_2}\right)_{n_1,n_2} = \begin{pmatrix} \phi_{n_2,m_2,k_2}(x_{n_1,m_1,K},\, y_{n_1,m_1,K}) & 0 \\ 0 & \phi_{n_2,m_2,k_2}(x_{n_1,m_1,K},\, y_{n_1,m_1,K}) \end{pmatrix}, \tag{4.4c}$$

for $n_1, n_2 = 1, \ldots, N,\ m_1, m_2 = 1, \ldots, M,\ k_2 = 1, \ldots, K$, and

$$\left(A_{(k_1-1)M+m_1,(k_2-1)M+m_2}\right)_{n_1,n_2}$$
$$= \begin{pmatrix} \mathcal{L}_{11}\phi_{n_2,m_2,k_2}(x_{n_1,m_1,k_1},\, y_{n_1,m_1,k_1}) & \mathcal{L}_{12}\phi_{n_2,m_2,k_2}(x_{n_1,m_1,k_1},\, y_{n_1,m_1,k_1}) \\ \mathcal{L}_{21}\phi_{n_2,m_2,k_2}(x_{n_1,m_1,k_1},\, y_{n_1,m_1,k_1}) & \mathcal{L}_{22}\phi_{n_2,m_2,k_2}(x_{n_1,m_1,k_1},\, y_{n_1,m_1,k_1}) \end{pmatrix} \tag{4.4d}$$

for $n_1, n_2 = 1, \ldots, N,\ m_1, m_2 = 1, \ldots, M,\ k_1 = 2, \ldots, K-1,\ k_2 = 1, \ldots, K$.

The $2N \times 1$ vectors $\boldsymbol{a}_\ell, \boldsymbol{b}_\ell, \ell = 1, \ldots, L$, are defined as

$$\left(\boldsymbol{a}_{(k-1)M+m}\right)_n = \begin{pmatrix} a^{(1)}_{nmk} \\ a^{(2)}_{nmk} \end{pmatrix},\ n = 1, \ldots, N,\ m = 1, \ldots, M,\ k = 1, \ldots, K,$$

$$\left(\boldsymbol{b}_{(k-1)M+m}\right)_n = \begin{pmatrix} f_1(x_{nmk},\, y_{nmk}) \\ f_2(x_{nmk},\, y_{nmk}) \end{pmatrix},\ n = 1, \ldots, N,\ m = 1, \ldots, M,\ k = 2, \ldots, K-1,$$

$$(\boldsymbol{b}_m)_n = \begin{pmatrix} g_1(x_{nm1},\, y_{nm1}) \\ h_1(x_{nm1},\, y_{nm1}) \end{pmatrix},\ \left(\boldsymbol{b}_{(K-1)M+m}\right)_n = \begin{pmatrix} g_2(x_{nmK},\, y_{nmK}) \\ h_2(x_{nmK},\, y_{nmK}) \end{pmatrix},\ n = 1, \ldots, N,\ m = 1, \ldots, M.$$

Unlike the Poisson and biharmonic problems, the global matrix A does not possess a block circulant structure which, however, can be obtained by means of a simple transformation; see, e.g. [28, Section 4.3].

4.2 Matrix decomposition algorithm

Following the corresponding MDA in [28, Section 4.3], we first introduce the $2N \times 2N$ matrix

$$R = \left(\begin{array}{c|c|c|c|c|c} R_{\vartheta_1} & 0 & 0 & \cdots & 0 & 0 \\ \hline 0 & R_{\vartheta_2} & 0 & \cdots & 0 & 0 \\ \hline \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \hline 0 & 0 & 0 & \cdots & R_{\vartheta_{N-1}} & 0 \\ \hline 0 & 0 & 0 & \cdots & 0 & R_{\vartheta_N} \end{array}\right), \text{ where } R_{\vartheta_n} = \begin{pmatrix} \cos\vartheta_n & \sin\vartheta_n \\ \sin\vartheta_n & -\cos\vartheta_n \end{pmatrix}, \quad (4.5)$$

and $\vartheta_n = \frac{2\pi(n-1)}{N}$. Since clearly $R_{\vartheta_n}^2 = I_2$ then $R^2 = I_{2N}$.

Keeping in mind that $L = MK$, we premultiply the $2NL \times 2NL$ system $A\boldsymbol{a} = \boldsymbol{b}$ by the $2NL \times 2NL$ matrix $I_L \otimes R$ to get

$$(I_L \otimes R)\, A\boldsymbol{a} = (I_L \otimes R)\, \boldsymbol{b} \quad \text{or} \quad \tilde{A}\tilde{\boldsymbol{a}} = \tilde{\boldsymbol{b}}, \quad (4.6)$$

where

$$\tilde{A} = (I_L \otimes R)\, A\, (I_L \otimes R)\,, \quad \tilde{\boldsymbol{a}} = (I_L \otimes R)\, \boldsymbol{a}, \quad \tilde{\boldsymbol{b}} = (I_L \otimes R)\, \boldsymbol{b}.$$

The $2NL \times 2NL$ matrix $\tilde{A}$ can be written as

$$\tilde{A} = \begin{pmatrix} \tilde{A}_{1,1} & \tilde{A}_{1,2} & \dots & \tilde{A}_{1,L} \\ \tilde{A}_{2,1} & \tilde{A}_{2,2} & \dots & \tilde{A}_{2,L} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{A}_{L,1} & \tilde{A}_{L,2} & \dots & \tilde{A}_{L,L} \end{pmatrix}, \quad (4.7)$$

where each of the $2N \times 2N$ submatrices $\tilde{A}_{\ell_1,\ell_2} = RA_{\ell_1,\ell_2}R$.

Moreover, keeping in mind that the elements $\left(\tilde{A}_{\ell_1,\ell_2}\right)_{n_1,n_2} = \left(\left(\tilde{A}_{\ell_1,\ell_2}\right)_{n_1,n_2}\right)_{i,j=1}^{2}$ are 2×2 arrays, we have

$$\left(\tilde{A}_{\ell_1,\ell_2}\right)_{n_1,n_2} = R_{\vartheta_{n_1}} \left(A_{\ell_1,\ell_2}\right)_{n_1,n_2} R_{\vartheta_{n_2}}, \quad n_1, n_2 = 1, \dots, N,\ \ell_1, \ell_2 = 1, \dots, L. \quad (4.8)$$

Each submatrix $\tilde{A}_{\ell_1,\ell_2}$, $\ell_1, \ell_2 = 1, \dots, L$, has a block 2×2 block circulant structure. The $2NL \times 1$ vectors $\tilde{\boldsymbol{a}}$, $\tilde{\boldsymbol{b}}$ are written as

$$\tilde{\boldsymbol{a}} = \begin{pmatrix} \tilde{\boldsymbol{a}}_1 \\ \tilde{\boldsymbol{a}}_2 \\ \vdots \\ \tilde{\boldsymbol{a}}_L \end{pmatrix}, \quad \tilde{\boldsymbol{b}} = \begin{pmatrix} \tilde{\boldsymbol{b}}_1 \\ \tilde{\boldsymbol{b}}_2 \\ \vdots \\ \tilde{\boldsymbol{b}}_L \end{pmatrix},$$

where the $2N \times 1$ subvectors $\tilde{\boldsymbol{a}}_\ell, \tilde{\boldsymbol{b}}_\ell$, $\ell = 1, \ldots L$, are defined by $\tilde{\boldsymbol{a}}_\ell = R\boldsymbol{a}_\ell, \tilde{\boldsymbol{b}}_\ell = R\boldsymbol{b}_\ell$ and the 2×1 subvectors $\left((\tilde{\boldsymbol{a}}_\ell)_n\right)_{i=1}^2$, $\left(\left(\tilde{\boldsymbol{b}}_\ell\right)_n\right)_{i=1}^2$, $n = 1, \ldots, N$, are defined by

$$(\tilde{\boldsymbol{a}}_\ell)_n = R_{\vartheta_n} (\boldsymbol{a}_\ell)_n , \quad \left(\tilde{\boldsymbol{b}}_\ell\right)_n = R_{\vartheta_n} (\boldsymbol{b}_\ell)_n .$$

We next rewrite system (4.6) in the form

$$\left(\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array}\right) \begin{pmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \end{pmatrix} = \begin{pmatrix} \boldsymbol{d}_1 \\ \boldsymbol{d}_2 \end{pmatrix}, \tag{4.9}$$

where the $NL \times NL$ matrices $B_{ij}, i, j = 1, 2$, are expressed in the form

$$B_{ij} = \begin{pmatrix} \tilde{B}_{1,1}^{ij} & \tilde{B}_{1,2}^{ij} & \ldots & \tilde{B}_{1,L}^{ij} \\ \tilde{B}_{2,1}^{ij} & \tilde{B}_{2,2}^{ij} & \ldots & \tilde{B}_{2,L}^{ij} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{B}_{L,1}^{ij} & \tilde{B}_{L,2}^{ij} & \ldots & \tilde{B}_{L,L}^{ij} \end{pmatrix}.$$

Each $N \times N$ submatrix $\tilde{B}_{\ell_1,\ell_2}^{ij}, i, j = 1, 2, \ \ell_1, \ell_2 = 1, \ldots, L$, is circulant and defined from

$$\left(\tilde{B}_{\ell_1,\ell_2}^{ij}\right)_{n_1,n_2} = \left(\left(\tilde{A}_{\ell_1,\ell_2}\right)_{n_1,n_2}\right)_{i,j}, \quad n_1, n_2 = 1, \ldots, N. \tag{4.10}$$

Also, the $NL \times 1$ vectors $\boldsymbol{c}_i, \boldsymbol{d}_i, i = 1, 2$, are defined from

$$\boldsymbol{c}_i = \begin{pmatrix} \tilde{\boldsymbol{c}}_1^i \\ \tilde{\boldsymbol{c}}_2^i \\ \vdots \\ \tilde{\boldsymbol{c}}_L^i \end{pmatrix}, \ \boldsymbol{d}_i = \begin{pmatrix} \tilde{\boldsymbol{d}}_1^i \\ \tilde{\boldsymbol{d}}_2^i \\ \vdots \\ \tilde{\boldsymbol{d}}_L^i \end{pmatrix}, \text{ where } \left(\tilde{c}_\ell^i\right)_n = \left((\tilde{a}_\ell)_n\right)_i , \ \left(\tilde{d}_\ell^i\right)_n = \left(\left(\tilde{b}_\ell\right)_n\right)_i , \ n = 1, \ldots, N. \tag{4.11}$$

We premultiply system (4.9) by the matrix $I_2 \otimes I_L \otimes U_N$ to get

$$\begin{aligned} &(I_2 \otimes I_L \otimes U_N) \left(\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array}\right) \left(I_2 \otimes I_L \otimes U_N^*\right) (I_2 \otimes I_L \otimes U_N) \begin{pmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \end{pmatrix} \\ &= (I_2 \otimes I_L \otimes U_N) \begin{pmatrix} \boldsymbol{d}_1 \\ \boldsymbol{d}_2 \end{pmatrix}, \end{aligned} \tag{4.12}$$

or

$$\left(\begin{array}{c|c} \tilde{B}_{11} & \tilde{B}_{12} \\ \hline \tilde{B}_{21} & \tilde{B}_{22} \end{array}\right) \begin{pmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \end{pmatrix} = \begin{pmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \end{pmatrix}, \tag{4.13}$$

where

$$\begin{pmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \end{pmatrix} = (I_2 \otimes I_L \otimes U_N) \begin{pmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \end{pmatrix}, \quad \begin{pmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \end{pmatrix} = (I_2 \otimes I_L \otimes U_N) \begin{pmatrix} \boldsymbol{d}_1 \\ \boldsymbol{d}_2 \end{pmatrix},$$

where for $i = 1, 2$,

$$\boldsymbol{p}_i = \begin{pmatrix} \tilde{\boldsymbol{p}}_1^i \\ \tilde{\boldsymbol{p}}_2^i \\ \vdots \\ \tilde{\boldsymbol{p}}_L^i \end{pmatrix}, \quad \boldsymbol{q}_i = \begin{pmatrix} \tilde{\boldsymbol{q}}_1^i \\ \tilde{\boldsymbol{q}}_2^i \\ \vdots \\ \tilde{\boldsymbol{q}}_L^i \end{pmatrix}, \quad \text{with} \quad \tilde{\boldsymbol{p}}_\ell^i = U_N \tilde{\boldsymbol{c}}_\ell^i,\ \tilde{\boldsymbol{q}}_\ell^i = U_N \tilde{\boldsymbol{d}}_\ell^i,\ \ell = 1, \ldots, L.$$

The matrices $\tilde{B}_{ij}, i, j = 1, 2$, are given from

$$\tilde{B}_{ij} = (I_L \otimes U_N)\, B_{ij} \left(I_L \otimes U_N^*\right),$$

and since each of the matrices $B_{ij}, i, j = 1, 2$ is block circulant, from (2.13) it follows that

$$\tilde{B}_{ij} = \begin{pmatrix} D_{1,1}^{ij} & D_{1,2}^{ij} & \cdots & D_{1,L}^{ij} \\ D_{2,1} & D_{2,2}^{ij} & \cdots & D_{2,L}^{ij} \\ \vdots & \vdots & & \vdots \\ D_{L,1}^{ij} & D_{L,2}^{ij} & \cdots & D_{L,L}^{ij} \end{pmatrix}, \tag{4.14}$$

where each $N \times N$ matrix $D_{\ell_1,\ell_2}^{ij}, \ell_1, \ell_2 = 1, \ldots, L$, is diagonal. More specifically, if

$$D_{\ell_1,\ell_2}^{ij} = \operatorname{diag}\left(D_{\ell_1,\ell_{21}}^{ij}, D_{\ell_1,\ell_{22}}^{ij}, \ldots, D_{\ell_1,\ell_{2N}}^{ij}\right) \quad \text{and} \quad \tilde{B}_{\ell_1,\ell_2}^{ij} = \operatorname{circ}\left(\tilde{B}_{\ell_1,\ell_{21}}^{ij}, \tilde{B}_{\ell_1,\ell_{22}}^{ij} \ldots, \tilde{B}_{\ell_1,\ell_{2N}}^{ij}\right),$$

we have, for $\ell_1, \ell_2 = 1, \ldots, L$,

$$D_{\ell_1,\ell_{2n}}^{ij} = \sum_{k=1}^{N} \tilde{B}_{\ell_1,\ell_{2k}}^{ij} \omega^{(k-1)(n-1)},\ n = 1, \ldots, N. \tag{4.15}$$

Since each matrix $\tilde{B}_{ij}, i, j = 1, 2$, consists of L^2 blocks of order N each of which is diagonal, the solution of system (4.13) can be decomposed into solving the N systems of order $4L$

$$\left(\begin{array}{c|c} E_{11}^n & E_{12}^n \\ \hline E_{21}^n & E_{22}^n \end{array}\right) \left(\begin{array}{c} \boldsymbol{x}_1^n \\ \hline \boldsymbol{x}_2^n \end{array}\right) = \left(\begin{array}{c} \boldsymbol{y}_1^n \\ \hline \boldsymbol{y}_2^n \end{array}\right), \quad n = 1, \ldots, N, \tag{4.16}$$

where

$$\left(E_{ij}^n\right)_{\ell_1,\ell_2} = D_{\ell_1,\ell_{2n}}^{ij},\ \ell_1, \ell_2 = 1, \ldots, L, \text{ and } \left(\boldsymbol{x}_i^n\right)_\ell = \left(\tilde{\boldsymbol{p}}_\ell^i\right)_n,\ \left(\boldsymbol{y}_i^n\right)_\ell = \left(\tilde{\boldsymbol{q}}_\ell^i\right)_n,\ \ell = 1, \ldots, L. \tag{4.17}$$

From the vectors $\boldsymbol{x}_i^n, i = 1, 2, n = 1, \ldots, L$ we can obtain the vectors p_1, p_2 and the vectors c_1, c_2, and subsequently the vector $\tilde{\boldsymbol{a}}$, before finally obtaining the vector $\boldsymbol{a}$.

The MDA, in this case, can be summarized as follows:

Algorithm 2

Step 1: Compute $\tilde{\boldsymbol{b}} = (I_L \otimes R)\boldsymbol{b}$.
Step 2: Calculate the 2×2 arrays $\left(\tilde{A}_{\ell_1,\ell_2}\right)_{1,n_2}$.
Step 3: Compute $\tilde{\boldsymbol{q}}_\ell^i = U_N \tilde{\boldsymbol{d}}_\ell^i$, $\ell = 1, \ldots, L$ and hence $\boldsymbol{y}_i^n$, $n = 1, \ldots, N$ from (4.17).
Step 4: Construct the diagonal matrices D_{ℓ_1,ℓ_2}^{ij} from (4.15) and hence matrices E_{ij}^n in (4.16).
Step 5: Solve the N, $4L \times 4L$ systems (4.16) to obtain the $\boldsymbol{x}_i^n, i = 1, 2; n = 1, \ldots, N$, and subsequently the vectors $\boldsymbol{p}_i, i = 1, 2$.
Step 6: Recover the vectors $\boldsymbol{c}_i, i = 1, 2$ from $\tilde{\boldsymbol{c}}_\ell^i = U_N^* \tilde{\boldsymbol{p}}_\ell^i, \ell = 1, \ldots, L$.
Step 7: Reorder vectors $\boldsymbol{c}_i, i = 1, 2$ to obtain vector $\tilde{\boldsymbol{a}}$.
Step 8: Compute $\boldsymbol{a} = (I_L \otimes R)\tilde{\boldsymbol{a}}$.

In Steps 3, 4, and 6, FFTs are used while the most expensive part of the algorithm is the solution of N linear systems, each of order $4L$.

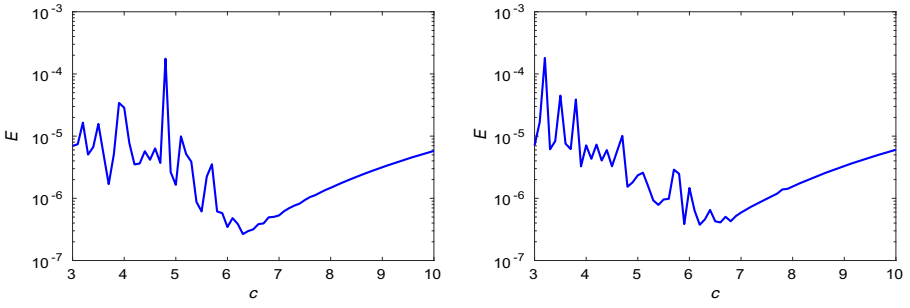
5 Numerical examples

The maximum relative error was calculated on a set of test points in $\overline{\Omega}$ defined from the $\mathcal{M}$ angles

$$\theta_m = \frac{2\pi(m-1)}{\mathcal{M}N}, \quad m = 1, \ldots, \mathcal{M}, \tag{5.1}$$

and the $\mathcal{K}$ radii

$$r_k = \varrho_1 + (\varrho_2 - \varrho_1)\frac{k-1}{\mathcal{K}-1}, \quad k = 1, \ldots, \mathcal{K}. \tag{5.2}$$



(a) Dirichlet problem (b) Mixed Neumann/Dirichlet problem

Fig. 2 Example 1: error versus c for the Poisson problem with $N = 9$, $M = K = 35$. **a** Dirichlet problem. **b** Neumann/Dirichlet problem

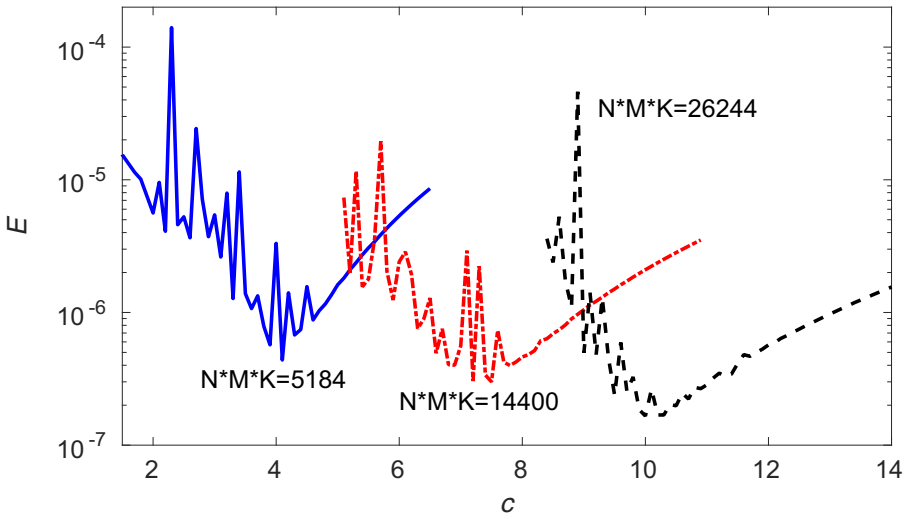


Fig. 3 Example 1: the relation of the number of collocation points and c

The test points $\{(\mathrm{x}_{nmk}, \mathrm{y}_{nmk})\}_{n=1,m=1,k=1}^{N,\mathcal{M},\mathcal{K}}$ were taken as follows:

$$\mathrm{x}_{nmk} = r_k \frac{\cos\left(\frac{\pi}{N}\right)\cos\left(\theta_m + \frac{2\pi}{N}(n-1)\right)}{\cos\left(\frac{\pi}{N} - \theta\right)}, \quad \mathrm{y}_{nmk} = r_k \frac{\cos\left(\frac{\pi}{N}\right)\sin\left(\theta_m + \frac{2\pi}{N}(n-1)\right)}{\cos\left(\frac{\pi}{N} - \theta\right)}, \tag{5.3}$$

for $n = 1, \ldots, N,\ m = 1, \ldots, \mathcal{M},\ k = 1, \ldots, \mathcal{K}$. Unless otherwise stated, we took $\mathcal{M} = \mathcal{K} = 7$, so that the test points are different from the collocation points. The

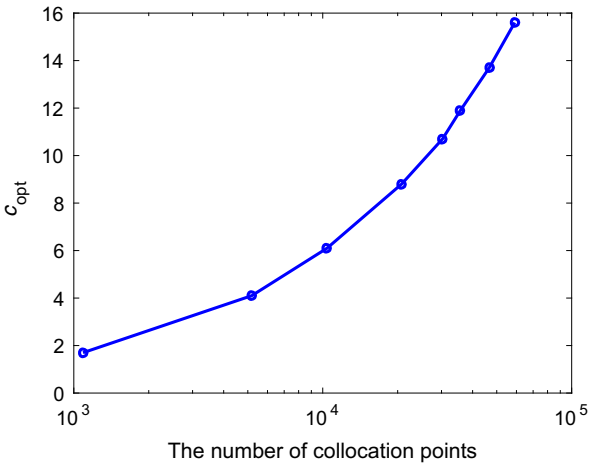


Fig. 4 Example 1: number of collocation points $\mathcal{N}$ versus c_{opt}

Table 1 Example 1: Dirichlet Poisson boundary value problem. Optimal and approximate shape parameters and their corresponding accuracy for various numbers of collocation points

$\mathcal{N}$	c_{opt}	E_{opt}	c_{app}	E
2916	2.9	5.169(−7)	2.850	1.062(−6)
15129	7.8	2.443(−7)	7.414	2.726(−7)
25281	9.8	1.408(−7)	9.735	1.675(−7)
40401	13.0	1.275(−7)	12.935	1.374(−7)
47961	14.1	1.012(−7)	13.737	2.902(−7)

maximum relative error E is defined as

$$E = \frac{||u - \hat{u}||_{\infty,\overline{\Omega}}}{||u||_{\infty,\overline{\Omega}}}. \tag{5.4}$$

In all examples, we shall denote by $\mathcal{N} = N \times M \times K$ and use the normalized MQ basis functions

$$\phi_j(x, y) = \sqrt{r_j^2 c^2 + 1}, \quad r_j^2 = (x - x_j)^2 + (y - y_j)^2, \quad \text{where } c \text{ is the shape parameter.}$$

The computations in this section were carried out using MATLAB© R2015a on the cluster BEM in the Wroclaw Centre for Networking and Supercomputing (Wroclaw, Poland) with Intel(R) Xeon(R) CPU E5-2670 v3 @ 2.30GHz and 24 cores in the computing node (or Intel(R) Xeon(R) CPU E5-2697 v3 @ 2.60GHz and 28 cores in the computing node), 64 to 512 GB memory, in Linux CentOS 6.

Example 1 We first considered the Poisson Dirichlet boundary value problem (2.1a), (2.1b)–(2.1c) corresponding to the exact solution $u(x, y) = \mathrm{e}^{2x+y}$ in a regular polygonal domain defined by $N = 9$ (nonagon) and $\varrho_1 = 0.5$ and $\varrho_2 = 1$. In Fig. 2a, we present the error E in the case $M = 35$, $K = 35$ versus the shape parameter c. The corresponding results for the Poisson mixed Neumann/Dirichlet boundary value problem (2.1a), (2.1d)–(2.1e) are presented in Fig. 2b.

As is well known, when using RBF approximations, the determination of an appropriate shape parameter remains a challenging issue. From Fig. 2, we observe that

Table 2 Example 1: Neumann/Dirichlet Poisson boundary value problem. Optimal and approximate shape parameters and their corresponding accuracy for various numbers of collocation points

$\mathcal{N}$	c_{opt}	E_{opt}	c_{app}	E
2916	2.9	5.092(−7)	2.850	1.619(−6)
15129	7.3	2.444(−7)	7.414	3.705(−7)
25281	10.1	2.099(−7)	9.735	5.138(−7)
40401	12.8	1.414(−7)	12.935	1.845(−7)
47961	13.8	1.142(−7)	13.737	3.170(−7)

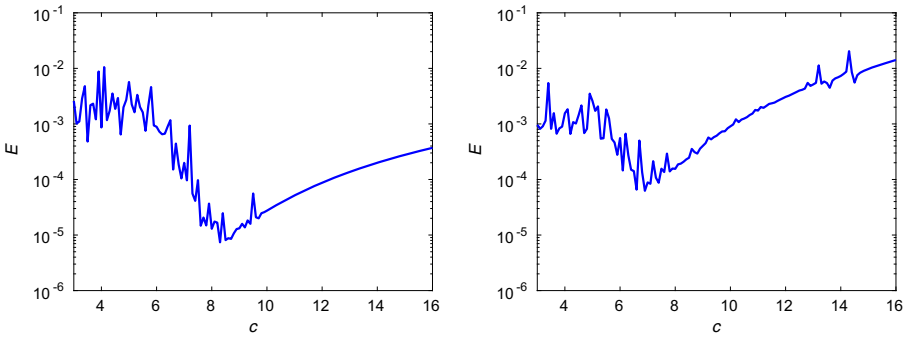


Fig. 5 Example 2: error versus c for the case $N = 6, M = K = 44$. **a** First biharmonic problem. **b** Second biharmonic problem

if we can find a way of estimating a value of the shape parameter c_{app} close to its optimal value c_{opt}, we can still obtain satisfactory accuracy. For the normalized MQ, we observe that the higher the density of the collocation points, the larger the value of the optimal shape parameter (see Fig. 3). In our problems, the collocation points are all confined in polygons inscribed in the unit circle and the value of the optimal shape parameter depends on the density of the collocation points. Based on this observation, we solved the Dirichlet boundary value problem with eight different sets of collocation points (each yielding a different number of collocation points $\mathcal{N}$). For each set of collocation points, we recorded the optimal value of the shape parameter (see Fig. 4) and corresponding error obtained by brute force. These eight sets of data were interpolated using the RBF $\phi(r) = r^3$ in order to predict the optimal value of the shape parameter for different numbers of collocation points.

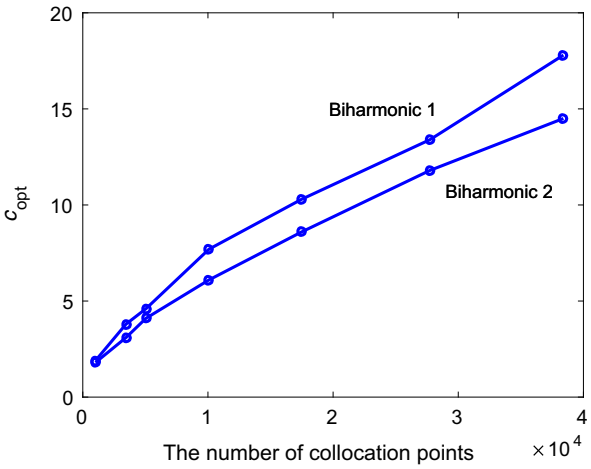


Fig. 6 Example 2: number of collocation points $\mathcal{N}$ versus c_{opt} for biharmonic problems

Table 3 Example 2: first biharmonic problem. Optimal and approximate shape parameters and their corresponding accuracy for various numbers of collocation points

$\mathcal{N}$	c_{opt}	E_{opt}	c_{app}	E
2400	2.9	2.578(−6)	3.109	3.000(−6)
13254	9.0	6.675(−6)	9.121	1.667(−5)
21600	11.5	7.761(−6)	11.276	9.394(−6)
30246	14.9	1.008(−5)	14.691	1.312(−5)
34656	16.3	1.078(−5)	16.916	2.457(−5)

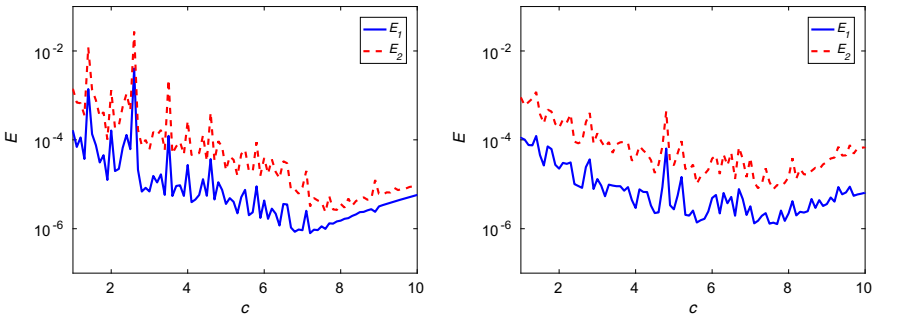
In Tables 1 and 2, we let c_{opt} be the optimal shape parameter and E_{opt} the corresponding maximum error (obtained by brute force), and c_{app} be the approximate shape parameter. From these tables, we observe that for various numbers of collocation points and for both the Dirichlet and the mixed Neumann/Dirichlet boundary value problem, the calculated c_{app} is close to c_{opt} and the maximum error E obtained with c_{app} is very close to E_{opt}.

We subsequently used the same eight data sets to predict an appropriate value of the shape parameter for Poisson problems in different regular polygons with different exact solutions. The results obtained were consistent with the optimal values of the shape parameter obtained by brute force, as was the case in the current example. This observation implies that we may choose a sample regular polygonal problem with a known exact solution and use the procedure mentioned above to predict appropriate values of the shape parameter for different regular polygonal problems without knowing their exact solution. This approach proved effective not only for the Dirichlet and mixed Neumann/Dirichlet Poisson boundary problems as shown in this example but also, as we shall see in Example 3, for the Cauchy-Navier equations.

Example 2 We next considered the first biharmonic problem (3.1a), (3.1b)–(3.1c) corresponding to the exact solution $u(x, y) = \mathrm{e}^{2x+y}$ in a regular polygonal domain defined by $N = 6$ (hexagon) and $\varrho_1 = 0.5$ and $\varrho_2 = 1$. In Fig. 5a, we present the error E in the case $M = K = 44$ versus the shape parameter c. The corresponding results for the second biharmonic problem (3.1a), (3.1d)–(3.1e) are presented in Fig. 5b.

Table 4 Example 2: second biharmonic problem. Optimal and approximate shape parameters and their corresponding accuracy for various numbers of collocation points

$\mathcal{N}$	c_{opt}	E_{opt}	c_{app}	E
2400	2.7	2.383(−5)	2.533	1.549(−5)
13254	7.6	5.693(−5)	7.159	6.606(−5)
21600	10.4	1.289(−4)	9.839	2.366(−4)
30246	12.6	1.746(−4)	12.764	2.103(−4)
34656	13.1	1.405(−4)	14.213	1.413(−4)



(a) Dirichlet Cauchy-Navier problem (b) Neumann/Dirichlet Cauchy-Navier problem

Fig. 7 Example 3: error versus c for $N = 12, M = K = 35$. **a** Dirichlet Cauchy-Navier problem. **b** Neumann/Dirichlet Cauchy-Navier problem

In general, the numerical solution of fourth-order partial equations by RBF collocation methods is less stable than the numerical solution of second-order problems. In particular, the second biharmonic problem which includes a Laplacian boundary condition is very sensitive to the value of the shape parameter. In this example, as in Fig. 4 of Example 1, we chose seven data sets each for the both the first and second biharmonic problems using brute force to identify the values of the optimal shape parameters (see Fig. 6). We then used these data sets to predict the optimal value of the shape parameter c_{opt} for various numbers of collocation points as shown in Tables 3 and 4. From these tables, we observe that there is good agreement between E and E_{opt} for various values of $\mathcal{N}$. We also observe that the results obtained in this example are two to three orders of magnitude less accurate than results obtained for the second-order problems examined in Example 1.

Example 3 We next considered the Dirichlet Cauchy-Navier problem (4.1a), (4.1b)–(4.1c) corresponding to the exact solution $u_1 = e^{2x+y}$, $u_2 = \sin(x+3y)$ in a regular polygonal domain defined by $N = 12$ (dodecagon) and $\varrho_1 = 0.5$ and $\varrho_2 = 1$. We took Poisson's ratio and the shear modulus to be $\nu = 0.3$ and $\mu = 1$, respectively. In Fig. 7a, we present the maximum relative errors E_1 and E_2 in u_1 and u_2, respectively (defined as in (5.4)), in the case $M = K = 35$ versus the shape parameter c.

Table 5 Example 3: Dirichlet Cauchy-Navier problem. Optimal and approximate shape parameters and their corresponding accuracy for various $\mathcal{N}$

$\mathcal{N}$	c_{opt}	$E_{1(opt)}$	$E_{2(opt)}$	c_{app}	E_1	E_2
3072	2.82	1.088(−6)	4.907(−6)	2.944	2.066(−6)	1.003(−5)
7500	4.70	6.892(−7)	3.162(−6)	5.118	1.131(−6)	5.090(−6)
12288	6.82	1.181(−6)	2.288(−6)	6.645	9.805(−7)	5.750(−6)
17328	7.63	5.505(−7)	2.066(−6)	7.983	7.020(−7)	8.184(−6)
22188	9.11	6.364(−7)	1.643(−6)	9.116	6.234(−7)	4.032(−6)
27648	10.63	7.143(−7)	1.434(−6)	10.185	6.048(−7)	4.966(−6)

Table 6 Example 3: Neumann/Dirichlet Cauchy-Navier problem. Optimal and approximate shape parameters and their corresponding accuracy for various $\mathcal{N}$

$\mathcal{N}$	c_{opt}	$E_{1(opt)}$	$E_{2(opt)}$	c_{app}	E_1	E_2
3072	2.61	1.173(−6)	8.470(−6)	2.944	3.569(−6)	2.885(−5)
7500	4.74	1.304(−6)	8.615(−6)	5.118	1.563(−6)	1.060−5)
12288	6.76	1.129(−6)	6.358(−6)	6.645	1.722(−6)	1.668(−5)
17328	8.44	1.738(−6)	6.383(−6)	7.983	1.764(−6)	1.442(−5)
22188	8.98	8.420(−7)	6.804(−6)	9.116	3.262(−6)	3.272(−5)
27648	9.45	6.483(−7)	7.556(−6)	10.185	2.434(−6)	1.800(−5)

The corresponding results for the mixed Neumann/Dirichlet Cauchy-Navier problem (4.1a), (4.1d)–(4.1e) are presented in Fig. 7b.

Since Cauchy-Navier problems are governed by second-order partial differential equations, we use the same method as in Example 1 to predict the optimal value of the shape parameter. In other words, we use the same data sets as shown in Fig. 4 to approximate the shape parameters for various values of $\mathcal{N}$. The results shown in Tables 5 and 6 are quite satisfactory, especially in view of the fact that no additional cost is needed to produce the eight data sets presented in Fig. 4, for the prediction of the shape parameter.

6 Conclusions

A Kansa-RBF method has been applied for the solution of three types of elliptic boundary value problems in regular polygonal domains. In the case of Poisson and inhomogeneous biharmonic problems, by appropriately choosing the collocation points, the RBF discretization leads to linear systems possessing block circulant structures. In the case of problems governed by the inhomogeneous Cauchy-Navier equations of elasticity, an additional step is required before, again, obtaining block circulant linear systems. In all three cases, the block circulant linear systems are solved efficiently using MDAs with FFTs. This leads to substantial savings in both computational time and storage. The more sides the regular polygon possesses the higher the savings.

The determination of an appropriate value for shape parameter in RBFs has always been a challenging issue. In this paper, a novel method for obtaining such a value for the shape parameter is proposed which is based on the observation that there is a strong correlation between the density of the collocation points and the value of the optimal shape parameter. The proposed approach is more effective for the second-order partial differential equations than for fourth-order ones.

Although all the regular polygonal domains considered in this work had a regular polygonal hole, the application of the proposed MDAs to regular polygonal domains without a hole is trivial. Moreover, the proposed MDAs are applicable for any choice of RBF which does not involve polynomial terms.

Possible areas of future research include the application of the proposed method using compactly supported RBFs, see, e.g. [2] and localized RBF collocation, see, e.g. [25], as well as certain three-dimensional polyhedra possessing a form of axial symmetry, such as regular polygonal prisms.

Acknowledgments This research was supported in part by the PLGrid Infrastructure. The computations were performed on the cluster BEM in the Wroclaw Centre for Networking and Supercomputing (Wroclaw, Poland).

References

1. Bialecki, B., Fairweather, G., Karageorghis, A.: Matrix decomposition algorithms for elliptic boundary value problems: a survey. Numer. Algorithms **56**, 253–295 (2011)
2. Buhmann, M.D.: A new class of radial basis functions with compact support. Math. Comp. **70**, 307–318 (2001)
3. Chen, W., Fu, Z.-J., Chen, C.S.: Recent advances in radial basis function collocation methods, Springer Briefs in Applied Sciences and Technology. Springer, Heidelberg (2014)
4. Cheng, A.H.-D.: Multiquadric and its shape parameter—a numerical investigation of error estimate, condition number, and round-off error by arbitrary precision computation. Eng. Anal. Bound. Elem. **36**, 220–239 (2012)
5. Davis, P.J.: Circulant Matrices, Second Edition. AMS Chelsea Publishing, Providence (1994)
6. Fasshauer, G.E.: Newton iteration with multiquadrics for the solution of nonlinear PDEs. Comput. Math. Appl. **43**, 423–438 (2002)
7. Fasshauer, G.E.: Meshfree approximation methods with MATLAB. In: Interdisciplinary Mathematical Sciences, vol. 6. World Scientific Publishing Co. Pte. Ltd., Hackensack (2007)
8. Fasshauer, G.E., Zhang, J.G.: On choosing optimal shape parameters for RBF approximation. Numer. Algorithms **45**, 345–368 (2007)
9. Franke, R.: Scattered data interpolation: tests of some methods. Math. Comp. **48**, 181–200 (1982)
10. González-Casanova, P., Muñoz-Gómez, J.A., Rodríguez-Gómez, G.: Node adaptive domain decomposition method by radial basis functions. Numer Methods Partial Differential Equations **25**, 1482–1501 (2009)
11. Greengard, L., Rokhlin, V.: A fast algorithm for particle simulations. J. Comput. Phys. **73**, 325–348 (1987)
12. Hardy, R.L.: Multiquadric equations of topography and other irregular surfaces. J. Geophys. Res. **176**, 1905–1915 (1971)
13. Hartmann, F.: Elastostatics. In: Brebbia, C.A. (ed.) Progress in boundary element methods, vol. 1, pp. 84–167. Pentech Press, London (1981)
14. Heryudono, A.R.H., Driscoll, T.A.: Radial basis function interpolation on irregular domain through conformal transplantation. J. Sci. Comput. **44**, 286–300 (2010)
15. Ingber, M.S., Chen, C.S., Tanski, J.A.: A mesh free approach using radial basis functions and parallel domain decomposition for solving three dimensional diffusion equations. Internat. J. Numer. Methods Engrg. **60**, 2183–2201 (2004)
16. Kansa, E.J.: Multiquadrics–a scattered data approximation scheme with applications to computational fluid-dynamics. II. Solutions to parabolic, hyperbolic and elliptic partial differential equations. Comput. Math. Appl. **19**, 147–161 (1990)
17. Kansa, E.J.: Radial basis functions: achievements and challenges. In: Boundary elements and other mesh reduction methods XXXVIII. WIT Trans. Model Simul, vol. 61, pp. 3–22. WIT Press, Southampton (2015)
18. Kansa, E.J., Hon, Y.C.: Circumventing the ill-conditioning problem with multiquadric radial basis functions: applications to elliptic partial differential equations. Comput. Math. Appl. **39**(7-8), 123–137 (2000)
19. Karageorghis, A.: Efficient MFS algorithms in regular polygonal domains. Numer. Algorithms **50**, 215–240 (2009)

20. Karageorghis, A., Chen, C.S., Liu, X.Y.: Kansa-RBF algorithms for elliptic problems in axisymmetric domains. SIAM J. Sci. Comput. **38**, A435–A470 (2016)
21. Karageorghis, A., Chen, C.S., Smyrlis, Y.-S.: A matrix decomposition RBF algorithm: approximation of functions and their derivatives. Appl. Numer. Math. **57**, 304–319 (2007)
22. Karageorghis, A., Chen, C.S., Smyrlis, Y.-S.: Matrix decomposition RBF algorithm for solving 3D elliptic problems. Eng. Anal. Bound. Elem. **33**, 1368–1373 (2009)
23. Lee, C.K., Liu, X., Fan, S.C.: Local multiquadric approximation for solving boundary value problems. Comput. Mech. **30**, 396–409 (2003)
24. Li, J., Hon, Y.C.: Domain decomposition for radial basis meshless methods. Numer. Methods Partial Differential Equations **20**, 450–462 (2004)
25. Li, M., Chen, W., Chen, C.S.: The localized RBFs collocation methods for solving high dimensional PDEs. Eng. Anal. Bound Elem. **37**, 1300–1304 (2013)
26. Ling, L., Kansa, E.J.: Preconditioning for radial basis functions with domain decomposition methods. Math. Comput. Modelling **40**, 1413–1427 (2004)
27. Liu, X.Y., Chen, C.S., Karageorghis, A.: Conformal mapping for the efficient solution of Poisson problems with the Kansa-RBF method. J. Sci. Comput. **71**, 1035–1061 (2017)
28. Liu, X.Y., Karageorghis, A., Chen, C.S.: A Kansa-radial basis function method for elliptic boundary value problems in annular domains. J. Sci. Comput. **65**, 1240–1269 (2015)
29. The MathWorks, Inc., 3 Apple Hill Dr., Natick, MA, Matlab
30. Rippa, S.: An algorithm for selecting a good value for the parameter c in radial basis function interpolation. Adv. Comput. Math. **11**, 193–210 (1999)