A Cartesian-grid collocation method based on radial-basis-function networks for solving PDEs in irregular domains

N. Mai-Duy* and T. Tran-Cong

Faculty of Engineering and Surveying,

The University of Southern Queensland, Toowoomba, QLD 4350, Australia

Submitted to *Numerical Methods for Partial Differential Equations*,
24/4/2006; revised 14/9/2006

*Corresponding author: Telephone +61 7 4631 1324, Fax +61 7 4631 2526, E-mail maiduy@usq.edu.au

Abstract: This paper reports a new Cartesian-grid collocation method based on radial-basis-function networks (RBFNs) for numerically solving elliptic partial differential equations (PDEs) in irregular domains. The domain of interest is embedded in a Cartesian grid, and the governing equation is discretized by using a collocation approach. The new features here are (a) One-dimensional integrated RBFNs are employed to represent the variable along each line of the grid, resulting in a significant improvement of computational efficiency, (b) The present method does not require complicated interpolation techniques for the treatment of Dirichlet boundary conditions in order to achieve a high level of accuracy, and (c) Normal derivative boundary conditions are imposed by means of integration constants. The method is verified through the solution of second- and fourth-order PDEs; accurate results and fast convergence rates are obtained.

Key words: integrated radial-basis-function network, collocation method, Cartesian grid, irregular domain.

1 INTRODUCTION

Partial differential equations arise in the mathematical modelling of physical phenomena. Solutions to these equations can be obtained by means of numerical discretization methods. It is well known that the finite-element method (FEM) is the most popular discretization method in engineering computations. A salient feature of the FEM is that it requires a mesh to support the interpolation of a solution variable and the integration of a Galerkin weak form. For problems involving complex geometries, generating a mesh is typically the most costly and time-consuming part of the solution process. As a result, much effort has been devoted to the development of the so-called meshless methods and Cartesian-grid methods.

Meshless methods have attracted a great deal of attention in recent decades. The domain

of interest is simply represented by a set of unstructured discrete points. For a local, truly-meshless method, only a small region associated with a point, a node's region of influence, is activated to construct the approximations for that point, and the governing equation is solved by employing a variety of approaches, including point collocation, local symmetric weak form and local boundary-integral-equation formulation. The two most common shapes of an influence domain are circles and rectangles. Implementing local background overlapping cells is much easier than implementing a mesh (non-overlapping and fixed topology). Comprehensive discussions on meshless methods can be found in review articles and monographs, see for example [1-4].

Cartesian-grid methods have a long history. In recent years, there has been a renewed interest in the development of these methods, and their applications have become much more widespread. The irregular domain of interest is embedded in a Cartesian grid. Generating a Cartesian grid is a straightforward task, and hence the computational costs associated with mesh generation are greatly reduced. However, attention must be paid to the issue of how to handle irregular boundaries. The incorporation of boundary conditions on the immersed boundaries needs to be conducted in a way that does not adversely impact the accuracy of the method. There is a vast amount of literature on this subject, see for example [5-10] and references therein. For most Cartesian-grid methods reported, they are based on a finite-difference or a finite-volume discretization, which usually lead to methods that are second-order accurate.

Radial-basis-function networks (RBFNs) can be considered as a universal approximation scheme [11]. Madych and Nelson [12,13] showed that the RBF interpolation scheme using multiquadrics (MQ) exhibits exponential convergence/spectral accuracy. The application of MQ-RBFNs for the solution of PDEs has been an active research area over the past fifteen years. A great number of publications are available, see for example [14-19] and references therein. The MQ collocation method is truly meshless, and it is extremely easy to implement. The main drawback of the method is the lack of mathematical theories

for finding the appropriate values of network parameters. For example, the RBF width, which strongly affects the performance of RBFNs, has still been chosen either by empirical approaches or by optimization techniques, see for example [16,17]. On the other hand, in a computation, only a finite number of digits can be retained by the computer. As a result, it remains very difficult to achieve such exponential convergence in practice, even for the case of function approximation. As an alternative to the conventional direct/differentiated RBFN (DRBFN) method, Mai-Duy and Tran-Cong [18,20] proposed using integration to construct the RBFN expressions (the indirect/integrated RBFN (IRBFN) method) for the approximation of a function and its derivatives and for the solution of PDEs. Numerical results showed that the IRBFN method achieves superior accuracy. The improvement is attributable to the fact that integration is a smoothing operation and is more numerically stable.

In this study, a Cartesian-grid method based on IRBFNs for solving PDEs in irregular domains is proposed. One-dimensional IRBFNs are employed to represent the variable along each line of the grid. The construction of RBF approximations for a point $\mathbf{x}$ involves only points that lie on lines intersected at $\mathbf{x}$ and parallel to the $x-$ and $y-$axes, rather than the whole set of data points. The inversion is now conducted for a series of small matrices rather than for a large matrix. This use of 1D-IRBFNs thus leads to a considerable economy in forming the system matrix over the 2D-IRBFN method reported in [18,21-24].

The main challenge faced by Cartesian-grid methods is how to represent the boundary conditions, especially for normal derivatives, on non-rectangular boundaries accurately. For the classical finite-difference method (FDM), the point adjacent to the boundary requires forms with changing Δx and Δy in order to impose boundary conditions, and such changes deteriorate the order of truncation error [25]. The proposed method does not require an underlying mesh along a grid line, and one can impose Dirichlet boundary conditions in a straightforward manner. In addition, the use of integration to construct the

RBF approximations provides a good means for implementing normal derivative boundary conditions. These can be seen as advantages of the 1D-IRBFN method over the FDM.

Three types of problems, namely the Poisson equation with Dirichlet boundary conditions, the biharmonic equation with Dirichlet boundary conditions, and the Poisson equation with Dirichlet and Neumann boundary conditions, are considered. The obtained results are compared with those of the conventional DRBFN method where appropriate; the proposed method outperforms the conventional one with respect to the condition number of the system matrix, accuracy and convergence rate. For all test problems, the method yields accurate results and fast convergence rates. It is worth mentioning that its performance for fourth-order PDEs is far superior to that for second-order PDEs.

The remainder of the paper is organized as follows. The proposed method is presented and verified for the three types of problems in sections 2, 3 and 4, respectively. Section 5 gives some concluding remarks.

2 POISSON EQUATION WITH DIRICHLET BOUNDARY CONDITIONS

2.1 Formulation

Consider the Poisson equation

$$\nabla^2 u = b, \tag{1}$$

in a bounded two-dimensional domain with Dirichlet boundary conditions, where b is a driving function. The irregular domain of interest is embedded in a rectangular domain (Figure 1) and it is then discretized using a Cartesian grid, i.e. an array of straight lines that run parallel to the $x-$ and $y-$axes. Let N_x and N_y be the numbers of grid lines

in the $x-$ and $y-$directions, respectively. The interior points are defined as grid points inside the problem domain, while the boundary points are generated by the intersection of the grid lines with boundaries. Grid nodes outside the problem domain are removed from the computations. It can be seen that the task of generating a Cartesian grid is much easier than the task of generating a finite-element mesh. How to automatically choose the boundary and interior points from Cartesian grids is beyond the scope of this study.

Consider a grid point/regular point $\mathbf{x}$ ($\mathbf{x} = (x, y)^T$) (Figure 1). Along the horizontal line passing through this point, one can use IRBFNs to construct the expressions for the function u and its derivatives with respect to x. The construction process can be described as follows. The second-order derivative of u is first decomposed into RBFs; the RBF network is then integrated twice to obtain the expressions for the first-order derivative and the function itself

$$\frac{\partial^2 u(x)}{\partial x^2} = \sum_{i=1}^{N} w^{(i)} g^{(i)}(x) = \sum_{i=1}^{N} w^{(i)} H_{[2]}^{(i)}(x), \tag{2}$$

$$\frac{\partial u(x)}{\partial x} = \sum_{i=1}^{N} w^{(i)} H_{[1]}^{(i)}(x) + c_1, \tag{3}$$

$$u(x) = \sum_{i=1}^{N} w^{(i)} H_{[0]}^{(i)}(x) + c_1 x + c_2, \tag{4}$$

where N is the number of nodal points (interior and boundary points) on the line, $\{w^{(i)}\}_{i=1}^{N}$ are RBF weights to be determined, $\{g^{(i)}(x)\}_{i=1}^{N}$ are known RBFs, $H_{[1]}(x) = \int H_{[2]}(x)dx$, $H_{[0]}(x) = \int H_{[1]}(x)dx$, and c_1 and c_2 are integration constants. Here, it is referred to as a second-order 1D-IRBFN scheme, denoted by IRBFN-2. The present study employs multiquadrics (MQ) whose form is

$$g^{(i)}(x) = \sqrt{(x - c^{(i)})^2 + a^{(i)2}}, \tag{5}$$

where $c^{(i)}$ and $a^{(i)}$ are the centre and the RBF width/shape parameter of the ith RBF. The set of centres is chosen to be the same as the set of the collocation points.

It is more convenient to work in the physical space than in the network-weight space. The values of the variable u at the N nodal points can be expressed as

$$u(x^{(1)}) = \sum_{i=1}^{N} w^{(i)} H_{[0]}^{(i)}(x^{(1)}) + c_1 x^{(1)} + c_2, \tag{6}$$

$$u(x^{(2)}) = \sum_{i=1}^{N} w^{(i)} H_{[0]}^{(i)}(x^{(2)}) + c_1 x^{(2)} + c_2, \tag{7}$$

$$\cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots$$

$$u(x^{(N)}) = \sum_{i=1}^{N} w^{(i)} H_{[0]}^{(i)}(x^{(N)}) + c_1 x^{(N)} + c_2, \tag{8}$$

or in a matrix form

$$\widehat{u} = \mathcal{H} \begin{pmatrix} \widehat{w} \\ \widehat{c} \end{pmatrix}, \tag{9}$$

where $\widehat{u} = (u^{(1)}, u^{(2)}, \cdots, u^{(N)})^T$, $\widehat{w} = (w^{(1)}, w^{(2)}, \cdots, w^{(N)})^T$, $\widehat{c} = (c_1, c_2)^T$, and $\mathcal{H}$ is a known matrix of dimension $N \times (N+2)$ defined as

$$\mathcal{H} = \begin{bmatrix} H_{[0]}^{(1)}(x^{(1)}) & H_{[0]}^{(2)}(x^{(1)}) & \cdots & H_{[0]}^{(N)}(x^{(1)}) & x^{(1)} & 1 \\ H_{[0]}^{(1)}(x^{(2)}) & H_{[0]}^{(2)}(x^{(2)}) & \cdots & H_{[0]}^{(N)}(x^{(2)}) & x^{(2)} & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ H_{[0]}^{(1)}(x^{(N)}) & H_{[0]}^{(2)}(x^{(N)}) & \cdots & H_{[0]}^{(N)}(x^{(N)}) & x^{(N)} & 1 \end{bmatrix}.$$

Using the singular value decomposition (SVD) technique, one can write the RBF coefficients including two integration constants in terms of the meaningful nodal variable values

$$\begin{pmatrix} \widehat{w} \\ \widehat{c} \end{pmatrix} = \mathcal{H}^{-1} \widehat{u}. \tag{10}$$

It is noted that the purpose of using SVD here is to provide a solution whose norm is the smallest in the least-squares sense. By substituting (10) into (2)-(4), the values of u and

its derivatives with respective to x at point $\mathbf{x}$ can now be computed by

$$\frac{\partial^2 u(x)}{\partial x^2} = \left(H^{(1)}_{[2]}(x), H^{(2)}_{[2]}(x), \cdots, H^{(N)}_{[2]}(x), 0, 0\right) \mathcal{H}^{-1}\widehat{u}, \tag{11}$$

$$\frac{\partial u(x)}{\partial x} = \left(H^{(1)}_{[1]}(x), H^{(2)}_{[1]}(x), \cdots, H^{(N)}_{[1]}(x), 1, 0\right) \mathcal{H}^{-1}\widehat{u}, \tag{12}$$

$$u(x) = \left(H^{(1)}_{[0]}(x), H^{(2)}_{[0]}(x), \cdots, H^{(N)}_{[0]}(x), x, 1\right) \mathcal{H}^{-1}\widehat{u}. \tag{13}$$

It is noted that the above expressions are applicable to any point on the line through $\mathbf{x}$ parallel to the $x-$axis. Since $u^{(1)}$ and $u^{(N)}$ are given (Dirichlet boundary conditions), (11)-(13) can be rewritten in the form

$$\frac{\partial^2 u(x)}{\partial x^2} = \mathcal{D}_{2x}\widehat{u}_{ip} + k_{2x}, \tag{14}$$

$$\frac{\partial u(x)}{\partial x} = \mathcal{D}_{1x}\widehat{u}_{ip} + k_{1x}, \tag{15}$$

$$u(x) = \mathcal{D}_{0x}\widehat{u}_{ip} + k_{0x}, \tag{16}$$

where $\widehat{u}_{ip} = (u^{(2)}, u^{(3)}, \cdots, u^{(N-1)})^T$, $\mathcal{D}_{ix}$ $(i = \{0, 1, 2\})$ are known matrices of dimension $1 \times (N-2)$, and k_{ix} $(i = \{0, 1, 2\})$ are known constants whose values depend on the boundary conditions.

Similarly, along the vertical line passing through point $\mathbf{x}$, one can obtain the 1D-IRBFN expressions for u and its derivatives with respect y

$$\frac{\partial^2 u(y)}{\partial y^2} = \mathcal{D}_{2y}\widehat{u}_{ip} + k_{2y}, \tag{17}$$

$$\frac{\partial u(y)}{\partial y} = \mathcal{D}_{1y}\widehat{u}_{ip} + k_{1y}, \tag{18}$$

$$u(y) = \mathcal{D}_{0y}\widehat{u}_{ip} + k_{0y}. \tag{19}$$

It can be seen that the 1D-IRBFN approximations for u and its derivatives are written in terms of interior nodal values of u. Let N_{ip} be the total number of interior points, and N_{bpx} and N_{bpy} be the numbers of boundary points generated by the intersection of the

horizontal and vertical grid lines with the boundaries, respectively. Applying (14), (15), (17) and (18) to the nodal points over the whole domain, and then putting the obtained results together (like the assembly process in the FEM), one can obtain the following compact matrix-vector forms

$$\widetilde{\frac{\partial^2 u}{\partial x^2}} = \widetilde{\mathcal{D}}_{2x}\widetilde{u}_{ip} + \widetilde{k}_{2x}, \tag{20}$$

$$\widetilde{\frac{\partial u}{\partial x}} = \widetilde{\mathcal{D}}_{1x}\widetilde{u}_{ip} + \widetilde{k}_{1x}, \tag{21}$$

$$\widetilde{\frac{\partial^2 u}{\partial y^2}} = \widetilde{\mathcal{D}}_{2y}\widetilde{u}_{ip} + \widetilde{k}_{2y}, \tag{22}$$

$$\widetilde{\frac{\partial u}{\partial y}} = \widetilde{\mathcal{D}}_{1y}\widetilde{u}_{ip} + \widetilde{k}_{1y}, \tag{23}$$

where $\widetilde{\frac{\partial^2 u}{\partial x^2}} = \left(\frac{\partial^2 u^{(1)}}{\partial x^2}, \frac{\partial^2 u^{(2)}}{\partial x^2}, \cdots, \frac{\partial^2 u^{(N_{bpx}+N_{ip})}}{\partial x^2}\right)^T$, $\widetilde{u}_{ip}$ is a vector that consists of all interior nodal values of u, $\widetilde{\mathcal{D}}_{2x}$ is a known matrix of dimension $(N_{bpx} + N_{ip}) \times N_{ip}$, $\widetilde{k}_{2x}$ is a known vector, and so on.

Since the variable u is prescribed along the boundary, one only needs to find the values of u at the interior points. The objective here is to generate a number of algebraic equations equal to the number of unknowns. This can be achieved by collocating the governing equation (1) at the interior points (ip). Making use of (20) and (22), the discrete form of (1) can be written as

$$\left(\widetilde{\frac{\partial^2 u}{\partial x^2}}\right)_{ip} + \left(\widetilde{\frac{\partial^2 u}{\partial y^2}}\right)_{ip} = \left(\widetilde{b}\right)_{ip}, \tag{24}$$

or

$$\left[\left(\widetilde{\mathcal{D}}_{2x}\right)_{ip} + \left(\widetilde{\mathcal{D}}_{2y}\right)_{ip}\right]\widetilde{u}_{ip} = \left(\widetilde{b}\right)_{ip} - \left(\widetilde{k}_{2x}\right)_{ip} - \left(\widetilde{k}_{2y}\right)_{ip}, \tag{25}$$

or

$$\widetilde{\mathcal{A}}\widetilde{u}_{ip} = \widetilde{r}, \tag{26}$$

where $\widetilde{\mathcal{A}}$ is an $N_{ip} \times N_{ip}$ matrix, and $\widetilde{r}$ is a vector that is determined by the prescribed boundary values and the known driving function in the differential equation.

2.2 Numerical Results

For all numerical examples presented in this study, the width of the ith MQ-RBF, $a^{(i)}$, is simply chosen to be the minimum distance from the ith centre to its neighbours, and the interior points that fall very close to the boundary (within the distance of $h/8$, $h-$the spacing (grid size)) are removed from the set of nodal points.

The accuracy of an approximation scheme is measured by means of the discrete relative L_2 error defined as

$$N_e = \frac{\sqrt{\sum_{i=1}^{M}\left(u_e^{(i)} - u^{(i)}\right)^2}}{\sqrt{\sum_{i=1}^{M}\left(u_e^{(i)}\right)^2}}, \tag{27}$$

where M is the number of unknown nodal values of u, and u_e and u are the exact and computed solutions, respectively. Another important measure is the convergence rate of the solution with respect to the refinement of spatial discretization

$$N_e(h) \approx \gamma h^{\alpha} = O(h^{\alpha}) \tag{28}$$

in which α and γ are exponential model's parameters. Given a set of observations, these parameters can be found by the general linear least squares technique.

Consider the following Poisson equation

$$\nabla^2 u = -18\pi^2 \sin(3\pi x)\sin(3\pi y) \tag{29}$$

in a hollow domain (the region lying between a circle $R = 1/2$ and a square of $1/2 \times 1/2$ (Figure 2) with Dirichlet boundary conditions. The exact solution, which is plotted in (Figure 2), is given by

$$u = \sin(3\pi x)\sin(3\pi y). \tag{30}$$

To provide the basis for the assessment of the proposed method, the conventional DRBFN

method is also considered here. It uses the same sets of interior points and inner boundary points as the proposed method. However, the outer boundary points are replaced with $(N_x + N_y)$ points that are uniformly distributed along the circular boundary for a better performance. Since the DRBFN approximations are written in terms of network weights, it leads to the system matrix of dimension $(N_{ip} + N_{bp}) \times (N_{ip} + N_{bp})$.

A number of uniform $N_x \times N_y$ grids, namely $9 \times 9, 13 \times 13, \cdots, 101 \times 101$, are employed to study the convergence behaviour of the solution. Results concerning the condition number of the system matrix and the discrete relative L_2 error of the interior solution are given in Table 1. In terms of the condition number of the system, the proposed method yields condition numbers about four orders of magnitude lower than those associated with the conventional method. In terms of accuracy, more accurate results and faster convergence are achieved; for example, the convergence order and the L_2 error (N_e at the finest grid of 101×101) are $O(h^{3.23})$ and 9.93×10^{-6} for the 1D-IRBFN method, and $O(h^{1.52})$ and 2.02×10^{-3} for the DRBFN method. It can also be seen that the obtained convergence rate $O(h^{3.23})$ is faster than those of the standard Cartesian-grid methods reported in the literature (about $O(h^2)$). However, the present system matrix is not as sparse as those of finite- difference and finite-volume methods. A theoretical proof of the superior accuracy of the IRBFN method and the non-singularity of indirect RBFN matrices cannot be offered at this stage. Further studies are needed.

3 BIHARMONIC EQUATION WITH DIRICHLET BOUNDARY CONDITIONS

3.1 Formulation

The process of deriving the 1D-IRBFN formulation for biharmonic equations is similar to that for Poisson equations. However, the corresponding equations involve more terms, and one needs to pay attention to the following two issues: (a) the implementation of double boundary conditions (u and $\partial u/\partial n$) and (b) the treatment of mixed partial derivatives $\partial^4 u/\partial x^2 \partial y^2$. Notations used in this section and in the previous one have similar meanings.

Consider a grid point $\mathbf{x}$ (Figure 1). The nodal points along the horizontal line passing through point $\mathbf{x}$ are used to construct the approximations for u and $\partial^i u/\partial x^i$ ($i = \{1, 2, 3, 4\}$)

$$\frac{\partial^4 u(x)}{\partial x^4} = \sum_{i=1}^{N} w^{(i)} g^{(i)}(x) = \sum_{i=1}^{N} w^{(i)} H^{(i)}_{[4]}(x), \tag{31}$$

$$\frac{\partial^3 u(x)}{\partial x^3} = \sum_{i=1}^{N} w^{(i)} H^{(i)}_{[3]}(x) + c_1, \tag{32}$$

$$\frac{\partial^2 u(x)}{\partial x^2} = \sum_{i=1}^{N} w^{(i)} H^{(i)}_{[2]}(x) + c_1 x + c_2, \tag{33}$$

$$\frac{\partial u(x)}{\partial x} = \sum_{i=1}^{N} w^{(i)} H^{(i)}_{[1]}(x) + c_1 \frac{x^2}{2} + c_2 x + c_3, \tag{34}$$

$$u(x) = \sum_{i=1}^{N} w^{(i)} H^{(i)}_{[0]}(x) + c_1 \frac{x^3}{6} + c_2 \frac{x^2}{2} + c_3 x + c_4, \tag{35}$$

in which the fourth-order derivative of u is decomposed into RBFs. Here, it is referred to as a fourth-order 1D-IRBF scheme, denoted by IRBF-4.

Some relevant matrices and vectors to be used for the conversion process are given below

$$\mathcal{H} = \begin{bmatrix} H_{[0]}^{(1)}(x^{(1)}) & H_{[0]}^{(2)}(x^{(1)}) & \cdots & H_{[0]}^{(N)}(x^{(1)}) & x^{(1)3}/6 & x^{(1)2}/2 & x^{(1)} & 1 \\ H_{[0]}^{(1)}(x^{(2)}) & H_{[0]}^{(2)}(x^{(2)}) & \cdots & H_{[0]}^{(N)}(x^{(2)}) & x^{(2)3}/6 & x^{(2)2}/2 & x^{(2)} & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ H_{[0]}^{(1)}(x^{(N)}) & H_{[0]}^{(2)}(x^{(N)}) & \cdots & H_{[0]}^{(N)}(x^{(N)}) & x^{(N)3}/6 & x^{(N)2}/2 & x^{(N)} & 1 \end{bmatrix}$$

$$\widehat{w} = \begin{pmatrix} w^{(1)} \\ w^{(2)} \\ \cdots \\ w^{(N)} \end{pmatrix}, \qquad \widehat{c} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix}, \qquad \widehat{u} = \begin{pmatrix} u^{(1)} \\ u^{(2)} \\ \cdots \\ u^{(N)} \end{pmatrix}.$$

It can be seen that the presence of integration constants allows the addition of some extra equations to the conversion system. Given the double boundary conditions u and $\partial u/\partial n$, it is straightforward to obtain the values of $\partial u/\partial x$ and $\partial u/\partial y$ along the boundaries. The additional matrix and vector can be generated as follows

$$\mathcal{K} = \begin{bmatrix} H_{[1]}^{(1)}(x^{(1)}) & H_{[1]}^{(2)}(x^{(1)}) & \cdots & H_{[1]}^{(N)}(x^{(1)}) & x^{(1)2}/2 & x^{(1)} & 1 & 0 \\ H_{[1]}^{(1)}(x^{(N)}) & H_{[1]}^{(2)}(x^{(N)}) & \cdots & H_{[1]}^{(N)}(x^{(N)}) & x^{(N)2}/2 & x^{(N)} & 1 & 0 \end{bmatrix},$$

$$\widehat{f} = \begin{pmatrix} \frac{\partial u}{\partial x}(x^{(1)}) \\ \frac{\partial u}{\partial x}(x^{(N)}) \end{pmatrix}.$$

The conversion process thus becomes

$$\begin{pmatrix} \widehat{u} \\ \widehat{f} \end{pmatrix} = \begin{bmatrix} \mathcal{H} \\ \mathcal{K} \end{bmatrix} \begin{pmatrix} \widehat{w} \\ \widehat{c} \end{pmatrix} = \mathcal{C} \begin{pmatrix} \widehat{w} \\ \widehat{c} \end{pmatrix}, \tag{36}$$

$$\begin{pmatrix} \widehat{w} \\ \widehat{c} \end{pmatrix} = \mathcal{C}^{-1} \begin{pmatrix} \widehat{u} \\ \widehat{f} \end{pmatrix}. \tag{37}$$

Substitution of (37) into (31)-(34) yields

$$\frac{\partial^4 u(x)}{\partial x^4} = \left(H^{(1)}_{[4]}(x), H^{(2)}_{[4]}(x), \cdots, H^{(N)}_{[4]}(x), 0, 0, 0, 0\right) \mathcal{C}^{-1} \begin{pmatrix} \widehat{u} \\ \widehat{f} \end{pmatrix}, \tag{38}$$

$$\frac{\partial^3 u(x)}{\partial x^3} = \left(H^{(1)}_{[3]}(x), H^{(2)}_{[3]}(x), \cdots, H^{(N)}_{[3]}(x), 1, 0, 0, 0\right) \mathcal{C}^{-1} \begin{pmatrix} \widehat{u} \\ \widehat{f} \end{pmatrix}, \tag{39}$$

$$\frac{\partial^2 u(x)}{\partial x^2} = \left(H^{(1)}_{[2]}(x), H^{(2)}_{[2]}(x), \cdots, H^{(N)}_{[2]}(x), x, 1, 0, 0\right) \mathcal{C}^{-1} \begin{pmatrix} \widehat{u} \\ \widehat{f} \end{pmatrix}, \tag{40}$$

$$\frac{\partial u(x)}{\partial x} = \left(H^{(1)}_{[1]}(x), H^{(2)}_{[1]}(x), \cdots, H^{(N)}_{[1]}(x), \frac{x^2}{2}, x, 1, 0\right) \mathcal{C}^{-1} \begin{pmatrix} \widehat{u} \\ \widehat{f} \end{pmatrix}. \tag{41}$$

Since $u^{(1)}, u^{(N)}$ and $\widehat{f}$ are known, the above expressions can be rewritten as

$$\frac{\partial^i u(x)}{\partial x^i} = \mathcal{D}^{IV}_{ix} \widehat{u}_{ip} + k^{IV}_{ix}, \qquad i = \{1, 2, 3, 4\}, \tag{42}$$

where the superscript IV is used to indicate that $\mathcal{D}_{ix}$ and k_{ix} are obtained using the IRBF-4 scheme, $\mathcal{D}^{IV}_{ix}$ are known matrices of dimension $1 \times (N-2)$, and k_{ix} are known constants.

Similarly, the approximations for $\partial^i u/\partial y^i$ $(i = \{1, 2, 3, 4\})$ at point $\mathbf{x}$ are constructed using the nodal points along the vertical line passing through that point

$$\frac{\partial^i u(y)}{\partial y^i} = \mathcal{D}^{IV}_{iy} \widehat{u}_{ip} + k^{IV}_{iy}, \qquad i = \{1, 2, 3, 4\}. \tag{43}$$

Applying (42) and (43) to the nodal points over the whole domain leads to

$$\widetilde{\frac{\partial^i u}{\partial x^i}} = \widetilde{\mathcal{D}}^{IV}_{ix} \widetilde{u}_{ip} + \widetilde{k}^{IV}_{ix}, \tag{44}$$

$$\widetilde{\frac{\partial^i u}{\partial y^i}} = \widetilde{\mathcal{D}}^{IV}_{iy} \widetilde{u}_{ip} + \widetilde{k}^{IV}_{iy}, \tag{45}$$

where $\widetilde{\mathcal{D}}_{ix}^{IV}$ and $\widetilde{\mathcal{D}}_{iy}^{IV}$ are known matrices of dimensions $(N_{ip} + N_{bpx}) \times N_{ip}$ and $(N_{ip} + N_{bpy}) \times N_{ip}$, respectively.

The mixed fourth-order partial derivative $\partial^4 u/\partial x^2 \partial y^2$ can be computed by means of relevant second-order derivatives according to the following relation

$$\frac{\partial^4 u}{\partial x^2 \partial y^2} = \frac{1}{2}\left(\frac{\partial^2}{\partial x^2}\left(\frac{\partial^2 u}{\partial y^2}\right) + \frac{\partial^2}{\partial y^2}\left(\frac{\partial^2 u}{\partial x^2}\right)\right). \tag{46}$$

Due to the fact that the cross derivative needs information from both $x-$ and $y-$ directions, it is straightforward to obtain the values of this derivative only at the grid points (i.e interior points). Fortunately, the governing equation is required to be discretized at the interior points only. Expression (46) can be computed by

$$2\left(\widetilde{\frac{\partial^4 u}{\partial x^2 \partial y^2}}\right)_{ip} = \widetilde{\mathcal{D}}_{2x}^{*}\left(\widetilde{\frac{\partial^2 u}{\partial y^2}}\right)_{ip} + \widetilde{\mathcal{D}}_{2y}^{*}\left(\widetilde{\frac{\partial^2 u}{\partial x^2}}\right)_{ip}, \tag{47}$$

where the construction of $\widetilde{\mathcal{D}}_{2x}^{*}$ and $\widetilde{\mathcal{D}}_{2y}^{*}$ is similar to that of $\widetilde{\mathcal{D}}_{2x}$ (20) and $\widetilde{\mathcal{D}}_{2y}$ (22), except that the present training points do not include the boundary points ($\widetilde{k}_{2x}^{*} = \widetilde{k}_{2y}^{*} = []$). Substitution of (44) and (45) with $i = 2$ into (47) yields

$$2\left(\widetilde{\frac{\partial^4 u}{\partial x^2 \partial y^2}}\right)_{ip} = \widetilde{\mathcal{D}}_{2x}^{*}\left(\widetilde{\mathcal{D}}_{2y}^{IV}\right)_{ip}\widetilde{u}_{ip} + \widetilde{\mathcal{D}}_{2y}^{*}\left(\widetilde{\mathcal{D}}_{2x}^{IV}\right)_{ip}\widetilde{u}_{ip} + \widetilde{k}_{4xy}^{*}, \tag{48}$$

$$= \widetilde{\mathcal{D}}_{4xy}^{*}\widetilde{u}_{ip} + \widetilde{k}_{4xy}^{*}, \tag{49}$$

where $\widetilde{\mathcal{D}}_{4xy}^{*}$ is a known matrix of dimension $N_{ip} \times N_{ip}$ and $\widetilde{k}_{4xy}^{*}$ is a known vector.

Using (44), (45) with $i = 4$ and (49), the biharmonic equation $\nabla^4 u = b$ can be reduced to the following square system of algebraic equations

$$\left(\widetilde{\frac{\partial^4 u}{\partial x^4}}\right)_{ip} + 2\left(\widetilde{\frac{\partial^4 u}{\partial x^2 \partial y^2}}\right)_{ip} + \left(\widetilde{\frac{\partial^4 u}{\partial y^4}}\right)_{ip} = (b)_{ip}, \tag{50}$$

or

$$\left(\left(\widetilde{\mathcal{D}}_{4x}^{IV}\right)_{ip} + \widetilde{\mathcal{D}}_{4xy}^{*} + \left(\widetilde{\mathcal{D}}_{4y}^{IV}\right)_{ip}\right)\widetilde{u}_{ip} = (b)_{ip} - \left(\left(\widetilde{k}_{4x}\right)_{ip} + \widetilde{k}_{4xy}^{*} + \left(\widetilde{k}_{4y}\right)_{ip}\right), \tag{51}$$

or

$$\widetilde{\mathcal{A}}\widetilde{u}_{ip} = \widetilde{r}, \tag{52}$$

where $\widetilde{A}$ is an $N_{ip} \times N_{ip}$ matrix.

3.2 Numerical Results

The problem here is to find a function $u(x, y)$ satisfying the following biharmonic equation

$$\nabla^4 u = 256(\pi^2 - 1)^2 \left[\sin(4\pi x)\cosh(4y) - \cos(4\pi x)\sinh(4y)\right] \tag{53}$$

defined on an annulus domain of radii $R1 = 1/4$ and $R2 = 1/2$ (Figure 3) and subject to Dirichlet boundary conditions (u and $\partial u/\partial n$). The exact solution (Figure 3) is given by

$$u = \left[\sin(4\pi x)\cosh(4y) - \cos(4\pi x)\sinh(4y)\right] \tag{54}$$

from which the boundary data can be easily obtained.

The convergence behaviour of the method is numerically investigated using a number of uniform Cartesian grids, $11 \times 11, 17 \times 17, \cdots, 67 \times 67$. Table 2 shows that the proposed method produces a very high convergence rate, $O(h^{5.39})$. The condition numbers of the system matrix are relatively low, only up to $O(10^5)$. When compared to the case of Poisson equation (Table 1), it can be seen that the accuracy of the proposed method is enhanced with increasing order of the PDE. It appears that the method is particularly well suited to the solution of high-order PDEs. This observation is similar to those reported in [22,24].

4 POISSON EQUATION WITH DIRICHLET AND NEUMANN BOUNDARY CONDITIONS

4.1 Formulation

Special treatment is required for the imposition of Neumann boundary conditions at immersed boundaries. Viswanathan [26] proposed constructing a FD approximation at grid nodes that lie adjacent to the curved boundary by taking into account the rate of change of the normal gradient of u along the boundary. In the work of Thuraisamy [6], the normal derivative at a boundary point was approximated using two lines that intersect at that point and make angles of $\pi/4$ on either side of the local normal direction. Recently, Sanmigue-Rojas et al [27] reported a technique for generating a non-uniform Cartesian grid in which all the boundary nodes are regular nodes of the grid. As a result, the values of derivatives with respect to x and y at the boundary points can be computed in a straightforward manner. The technique of Sanmigue-Rojas and his co-workers will be applied here to discretize sub-regions involving the Neumann boundary condition.

Consider a domain as shown in Figure 4. A Neumann boundary condition is specified on the segment CD, while Dirichlet boundary conditions are applied along AB, BC, DE, EF and FA (AB=BC=DE=EF=1/2). The requirement here is that the boundary points on the segment CD are also grid nodes, thereby avoiding the usual complicated interpolations at the boundaries. Let $bp*$ denote the boundary points on CD. Since ip and $bp*$ are grid nodes, one can easily obtain their 1D-IRBFN expressions for derivatives with respect to x and y. To obtain the values of u at ip and $bp*$, one needs to generate a set of $(N_{ip} + N_{bp*})$ algebraic equations.

4.1.1 Approach 1

The system of equations is generated here by applying the governing equation to the interior points ip and by collocating the Neumann boundary condition at the boundary points $bp*$.

The construction of the 1D-IRBFN approximations for this case is similar to the previous case of Poisson equation with Dirichlet boundary conditions. One only needs to pay a little attention to lines that cross the segment CD. For the vector $\widehat{u}$ in (11)-(12), only the first component $u^{(1)}$ is given, while the remaining components $\{u^{(2)}, u^{(3)}, \cdots, u^{(N)}\}$ are unknowns to be found. Expressions (14) and (16) thus become

$$\frac{\partial^2 u(x)}{\partial x^2} = \mathcal{D}_{2x} \begin{pmatrix} \widehat{u}_{ip} \\ u(x^{(N)}) \end{pmatrix} + k_{2x}, \tag{55}$$

$$\frac{\partial u(x)}{\partial x} = \mathcal{D}_{1x} \begin{pmatrix} \widehat{u}_{ip} \\ u(x^{(N)}) \end{pmatrix} + k_{1x}, \tag{56}$$

$$u(x) = \mathcal{D}_{0x} \begin{pmatrix} \widehat{u}_{ip} \\ u(x^{(N)}) \end{pmatrix} + k_{1x}, \tag{57}$$

where k_{ix} $(i = \{0, 1, 2\})$ are known constants whose values depend on $u^{(1)}$.

The assembly process leads to

$$\widetilde{\frac{\partial^2 u}{\partial x^2}} = \widetilde{\mathcal{D}}_{2x} \widetilde{u}_{ip+bp*} + \widetilde{k}_{2x}, \tag{58}$$

$$\widetilde{\frac{\partial u}{\partial x}} = \widetilde{\mathcal{D}}_{1x} \widetilde{u}_{ip+bp*} + \widetilde{k}_{1x}, \tag{59}$$

$$\widetilde{\frac{\partial^2 u}{\partial y^2}} = \widetilde{\mathcal{D}}_{2y} \widetilde{u}_{ip+bp*} + \widetilde{k}_{2y}, \tag{60}$$

$$\widetilde{\frac{\partial u}{\partial y}} = \widetilde{\mathcal{D}}_{1y} \widetilde{u}_{ip+bp*} + \widetilde{k}_{1y}, \tag{61}$$

For this approach, the system of equations can be written as

$$\left(\widetilde{\frac{\partial^2 u}{\partial x^2}}\right)_{ip} + \left(\widetilde{\frac{\partial^2 u}{\partial y^2}}\right)_{ip} = \left(\widetilde{b}\right)_{ip} \tag{62}$$

$$n_x \left(\widetilde{\frac{\partial u}{\partial x}}\right)_{bp*} + n_y \left(\widetilde{\frac{\partial u}{\partial y}}\right)_{bp*} = \left(\widetilde{\frac{\partial u}{\partial n}}\right)_{bp*}, \tag{63}$$

or

$$\left[\left(\widetilde{\mathcal{D}}_{2x}\right)_{ip} + \left(\widetilde{\mathcal{D}}_{2y}\right)_{ip}\right] \widetilde{u}_{ip+bp*} = \left(\widetilde{b}\right)_{ip} - \left(\widetilde{k}_{2x}\right)_{ip} - \left(\widetilde{k}_{2y}\right)_{ip}, \tag{64}$$

$$\left[n_x \left(\widetilde{\mathcal{D}}_{1x}\right)_{bp*} + n_y \left(\widetilde{\mathcal{D}}_{1y}\right)_{bp*}\right] \widetilde{u}_{ip+bp*} = \left(\widetilde{\frac{\partial u}{\partial n}}\right)_{bp*} - n_x \left(\widetilde{k}_{1x}\right)_{bp*} - n_y \left(\widetilde{k}_{1y}\right)_{bp*}, \tag{65}$$

or

$$\widetilde{\mathcal{A}}\widetilde{u}_{ip+bp*} = \widetilde{r}, \tag{66}$$

where n_x and n_y are the components of the unit vector normal to the boundary CD, and $\widetilde{\mathcal{A}}$ is an $(N_{ip} + N_{bp*}) \times (N_{ip} + N_{bp*})$ matrix.

4.1.2 Approach 2

Unlike Approach 1, the nodal values of the Neumann boundary condition on the boundary CD are imposed through the conversion process. Consequently, it allows the exact satisfaction of the governing equation not only at the interior points *ip* but also at the boundary points *bp*∗.

Consider a horizontal line that crosses the line CD. The conversion process is based on

the following equations

$$u(x^{(1)}) = \sum_{i=1}^{N} w^{(i)} H_{[0]}^{(i)}(x^{(1)}) + c_1 x^{(1)} + c_2, \tag{67}$$

$$u(x^{(2)}) = \sum_{i=1}^{N} w^{(i)} H_{[0]}^{(i)}(x^{(2)}) + c_1 x^{(2)} + c_2, \tag{68}$$

$$\cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots$$

$$u(x^{(N)}) = \sum_{i=1}^{N} w^{(i)} H_{[0]}^{(i)}(x^{(N)}) + c_1 x^{(N)} + c_2, \tag{69}$$

$$n_x \frac{\partial u}{\partial x}(x^{(N)}) = \sum_{i=1}^{N} w^{(i)} H_{[1]}^{(i)}(x^{(N)}) + c_1, \tag{70}$$

or in a matrix form

$$\begin{pmatrix} \widehat{u} \\ n_x \frac{\partial u}{\partial x}(x^{(N)}) \end{pmatrix} = \mathcal{C} \begin{pmatrix} \widehat{w} \\ \widehat{c} \end{pmatrix}, \tag{71}$$

or

$$\begin{pmatrix} \widehat{w} \\ \widehat{c} \end{pmatrix} = \mathcal{C}^{-1} \begin{pmatrix} \widehat{u} \\ n_x \frac{\partial u}{\partial x}(x^{(N)}) \end{pmatrix} = \mathcal{C}^{-1} \begin{pmatrix} \widehat{u} \\ \frac{\partial u}{\partial n}(x^{(N)}) - n_y \frac{\partial u}{\partial y}(x^{(N)}) \end{pmatrix}. \tag{72}$$

Since $\partial u(x^{(N)})/\partial n$ is given and the value of $\partial u(x^{(N)})/\partial y$ can be replaced with a linear combination of nodal values of u using (61) (the expression of $\partial u/\partial y$ from Approach 1), one can express the RBF coefficients in terms of nodal variable values. The remainder of the construction process is similar to those of the previous sections.

For this approach, the system of equations can be written as

$$\left(\widetilde{\frac{\partial^2 u}{\partial x^2}}\right)_{ip+bp*} + \left(\widetilde{\frac{\partial^2 u}{\partial y^2}}\right)_{ip+bp*} = \left(\widetilde{b}\right)_{ip+bp*}, \tag{73}$$

or

$$\left[\left(\widetilde{\mathcal{D}}_{2x}\right)_{ip+bp*} + \left(\widetilde{\mathcal{D}}_{2y}\right)_{ip+bp*}\right] \widetilde{u}_{ip+bp*} = \left(\widetilde{b}\right)_{ip+bp*} - \left(\widetilde{k}_{2x}\right)_{ip+bp*} - \left(\widetilde{k}_{2y}\right)_{ip+bp*}, \tag{74}$$

or

$$\widetilde{\mathcal{A}}\widetilde{u}_{ip+bp*} = \widetilde{r}, \tag{75}$$

where $\widetilde{\mathcal{A}}$ is an $(N_{ip} + N_{bp*}) \times (N_{ip} + N_{bp*})$ matrix.

4.2 Numerical Results

The proposed method is applied to solve the Poisson equation of the form

$$\nabla^2 u = 4\left[\cos(2x)\cosh(2y) + \sin(2x)\sinh(2y)\right] \tag{76}$$

with the exact solution being

$$u = x\left[\sin(2x)\cosh(2y) - \cos(2x)\sinh(2y)\right]. \tag{77}$$

The variation of (77) over the square that covers the problem domain is plotted in Figure 4. A number of uniform grids, $7\times7, 11\times11, \cdots, 81\times81$, are employed. Table 3 reveals that Approach 2 is superior to Approach 1 with respect to the condition number of the system matrix, accuracy and convergence rate. The proposed method converges apparently as $O(h^{1.93})$ for Approach 1 and $O(h^{2.40})$ for Approach 2. At the finest grid, N_es are 6.88×10^{-5} and 3.34×10^{-6} for Approach 1 and Approach 2, respectively. Imposing normal derivatives by means of integration constants is recommended for use.

5 CONCLUDING REMARKS

This paper reports a new Cartesian-grid RBF collocation method for numerically solving elliptic PDEs in irregular domains. The proposed method combines the efficiency of a Cartesian-grid method and the property of high-order convergence of a 1D-integrated RBF

interpolation scheme. Three types of problems, namely Poisson equation with Dirichlet boundary condition, biharmonic equation with Dirichlet boundary condition, Poisson equation with Dirichlet and Neumann boundary conditions, are investigated. The nodal values of the variable along the boundary can be accommodated straightforwardly, while the boundary values of its normal derivative can be imposed effectively through the process of converting network weights into nodal variable values. The proposed method requires much less computational effort than the 2D-IRBFN method. Numerical tests show that the method yields fast convergence, i.e third- and fifth-order accuracy with respect to grid size for the the solution of second- and fourth-order PDEs, respectively.

ACKNOWLEDGEMENTS

This research was supported by the Australian Research Council. Dr. N. Mai-Duy is the recipient of an Australian Research Council Post-Doctoral Fellowship. The authors would like to thank the referees for their helpful comments.

REFERENCES

1. T. Belytschko, Y. Krongauz, D. Organ, M. Fleming and P. Krysl, "Meshless methods: an overview and recent developments," Computer Methods in Applied Mechanics and Engineering, 139, 3 (1996).

2. S.N. Atluri and S. Shen, The Meshless Local Petrov-Galerkin Method, Tech Science Press, Encino, 2002.

3. S. Li and W.K. Liu, "Meshfree and particle methods and their applications," Applied Mechanics Reviews, 55(1), 1 (2002).

4. G.R. Liu, Mesh Free Methods: Moving beyond the Finite Element Method, CRC Press, Boca Raton, 2003.

5. V. Thuraisamy, "Approximate solutions for mixed boundary value problems by finite-difference methods," Mathematics of Computation, 23(106), 373 (1969).

6. V. Thuraisamy, "Monotone type discrete analogue for the mixed boundary value problem," Mathematics of Computation, 23(106), 387 (1969).

7. H. Johansen and P. Colella, "A Cartesian grid embedded boundary method for Poisson's equation on irregular domains," Journal of Computational Physics, 147(1), 60 (1998).

8. T. Ye, R. Mittal, H.S. Udaykumar and W. Shyy, "An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries," Journal of Computational Physics, 156(2), 209 (1999).

9. F. Gibou, R.P. Fedkiw, L.-T. Cheng and M. Kang, "A second-order-accurate symmetric discretization of the Poisson equation on irregular domains," Journal of Computational Physics, 176(1), 205 (2002).

10. Z. Jomaa and C. Macaskill, "The embedded finite difference method for the Poisson equation in a domain with an irregular boundary and Dirichlet boundary conditions," Journal of Computational Physics, 202(2), 488 (2005).

11. S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice-Hall, New Jersey, 1999.

12. W.R. Madych and S.A. Nelson, "Multivariate interpolation and conditionally positive definite functions," Approximation Theory and its Applications, 4, 77 (1988).

13. W.R. Madych and S.A. Nelson, "Multivariate interpolation and conditionally positive definite functions, II," Mathematics of Computation, 54(189), 211 (1990).

14. E.J. Kansa, "Multiquadrics- A scattered data approximation scheme with applications to computational fluid-dynamics-II. Solutions to parabolic, hyperbolic and

elliptic partial differential equations," Computers and Mathematics with Applications, 19(8/9), 147 (1990).

15. G.E. Fasshauer, "Solving partial differential equations by collocation with radial basis functions," in Surface Fitting and Multiresolution Methods, A. LeMhaut, C. Rabut and L.L. Schumaker (Editors), Vanderbilt University Press, Nashville, TN, 1997, p. 131.

16. M. Zerroukat, H. Power and C.S. Chen, "A numerical method for heat transfer problems using collocation and radial basis functions," International Journal for Numerical Methods in Engineering, 42, 1263 (1998).

17. E.J. Kansa and Y.C. Hon, "Circumventing the ill-conditioning problem with multiquadric radial basis functions: applications to elliptic partial differential equations," Computers and Mathematics with Applications, 39, 123 (2000).

18. N. Mai-Duy and T. Tran-Cong, "Numerical solution of differential equations using multiquadric radial basis function networks," Neural Networks, 14(2), 185 (2001).

19. C. Shu, H. Ding and K.S. Yeo, "Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier-Stokes equations," Computer Methods in Applied Mechanics and Engineering, 192, 941 (2003).

20. N. Mai-Duy and T. Tran-Cong, "Approximation of function and its derivatives using radial basis function networks," Applied Mathematical Modelling, 27, 197 (2003).

21. N. Mai-Duy, "Indirect RBFN method with scattered points for numerical solution of PDEs," Computer Modeling in Engineering and Sciences, 6(2), 209 (2004).

22. N. Mai-Duy and R.I. Tanner, "Solving high order partial differential equations with radial basis function networks," International Journal for Numerical Methods in Engineering, 63, 1636 (2005).

23. N. Mai-Duy and T. Tran-Cong, "An efficient indirect RBFN-based method for numerical solution of PDEs," Numerical Methods for Partial Differential Equations, 21, 770 (2005).

24. N. Mai-Duy and T. Tran-Cong, "Solving biharmonic problems with scattered-point discretisation using indirect radial-basis-function networks," Engineering Analysis with Boundary Elements, 30(2), 77 (2006).

25. P.J. Roache, Computational Fluid Dynamics, Hermosa Publishers, Albuquerque, 1980.

26. R.V. Viswanathan, "Solution of Poisson's equation by relaxation method–normal gradient specified on curved boundaries," Mathematical Tables and Other Aids to Computation, 11(58), 67 (1957).

27. E. Sanmiguel-Rojas, J. Ortega-Casanova, C. del Pino and R. Fernandez-Feria, "A Cartesian grid finite-difference method for 2D incompressible viscous flows in irregular geometries," Journal of Computational Physics, 204(1), 302 (2005).

Table 1: Example 1 (Poisson equation, Dirichlet boundary conditions): Condition number and accuracy obtained by the conventional DRBFN and the proposed 1D-IRBF methods. Notice that h is the spacing (grid size) and $a(-b)$ means $a \times 10^{-b}$.

Grid	$cond(\widetilde{\mathcal{A}})$		$N_e(u)$	
	DRBFN	1D-IRBFN	DRBFN	1D-IRBFN
9×9	1.67(4)	3.81(0)	1.27(-1)	1.08(-1)
13×13	6.96(4)	1.98(1)	4.24(-2)	8.30(-3)
17×17	1.68(5)	3.52(1)	3.14(-2)	1.78(-3)
21×21	3.34(5)	6.03(1)	2.56(-2)	1.06(-3)
25×25	5.88(5)	4.55(1)	1.71(-2)	5.57(-4)
29×29	9.39(5)	6.19(1)	1.25(-2)	4.26(-4)
33×33	1.44(6)	1.31(2)	9.26(-3)	2.46(-4)
37×37	2.05(6)	1.62(2)	7.72(-3)	1.91(-4)
41×41	2.83(6)	2.62(2)	6.96(-3)	1.39(-4)
45×45	3.76(6)	2.59(2)	7.57(-3)	1.11(-4)
49×49	4.94(6)	3.77(2)	5.94(-3)	8.27(-5)
53×53	6.26(6)	3.37(2)	4.86(-3)	6.71(-5)
57×57	7.88(6)	4.90(2)	4.29(-3)	5.27(-5)
61×61	9.69(6)	5.24(2)	3.91(-3)	4.36(-5)
65×65	1.17(7)	5.94(2)	4.16(-3)	3.71(-5)
69×69	1.41(7)	6.84(2)	3.87(-3)	3.13(-5)
73×73	1.67(7)	8.00(2)	3.32(-3)	2.59(-5)
77×77	1.97(7)	8.13(2)	2.91(-3)	2.21(-5)
81×81	2.30(7)	9.21(2)	2.68(-3)	1.89(-5)
85×85	2.66(7)	1.31(3)	2.51(-3)	1.64(-5)
89×89	3.06(7)	1.39(3)	2.48(-3)	1.45(-5)
93×93	3.50(7)	1.03(3)	2.45(-3)	1.30(-5)
97×97	3.98(7)	1.45(3)	2.22(-3)	1.12(-5)
101×101	4.50(7)	1.34(3)	2.02(-3)	9.93(-6)
			$O(h^{1.52})$	$O(h^{3.23})$

Table 2: Example 2 (biharmonic equation, Dirichlet boundary conditions): Condition number and accuracy. Notice that h is the spacing (grid size) and $a(-b)$ means $a \times 10^{-b}$.

Grid	$cond(\widetilde{\mathcal{A}})$	$N_e(u)$
11×11	2.93(1)	1.15(-2)
17×17	5.40(2)	1.05(-3)
21×21	2.12(3)	5.56(-4)
27×27	3.53(3)	4.81(-5)
31×31	1.41(4)	2.47(-5)
37×37	1.24(4)	1.46(-5)
41×41	3.41(4)	8.37(-6)
47×47	5.80(4)	1.97(-6)
51×51	8.37(4)	1.77(-6)
57×57	1.50(5)	1.37(-6)
61×61	2.29(5)	8.85(-7)
67×67	2.70(5)	5.87(-7)
		$O(h^{5.39})$

Table 3: Example 3 (Poisson equation, Dirichlet and Neumann boundary conditions): Condition number and accuracy obtained by the proposed method. The Neumann boundary conditions are imposed by adding addition equations to the system for Approach 1, and by means of integration constants for Approach 2. Notice that h is the spacing (grid size) and $a(-b)$ means $a \times 10^{-b}$.

Grid	$cond(\widetilde{\mathcal{A}})$		$N_e(u)$	
	Approach 1	Approach 2	Approach 1	Approach 2
7×7	2.01(2)	9.21(1)	1.00(-2)	2.20(-3)
11×11	5.47(2)	3.35(2)	3.91(-3)	4.13(-4)
17×17	2.06(3)	9.18(2)	1.62(-3)	1.17(-4)
21×21	3.99(3)	1.45(3)	1.05(-3)	6.85(-5)
27×27	9.34(3)	2.47(3)	6.33(-4)	3.70(-5)
31×31	1.51(4)	3.30(3)	4.79(-4)	2.69(-5)
37×37	1.67(4)	4.76(3)	3.34(-4)	1.79(-5)
41×41	2.25(4)	5.89(3)	2.72(-4)	1.42(-5)
47×47	3.40(4)	7.79(3)	2.06(-4)	1.05(-5)
51×51	4.43(4)	9.21(3)	1.75(-4)	8.84(-6)
57×57	6.85(4)	1.15(4)	1.39(-4)	7.01(-6)
61×61	7.95(4)	1.32(4)	1.21(-4)	6.04(-6)
67×67	1.08(5)	1.60(4)	1.00(-4)	4.96(-6)
71×71	1.32(5)	1.80(4)	8.98(-5)	4.40(-6)
77×77	1.48(5)	2.12(4)	7.62(-5)	3.71(-6)
81×81	1.68(5)	2.35(4)	6.88(-5)	3.34(-6)
			$O(h^{1.93})$	$O(h^{2.40})$

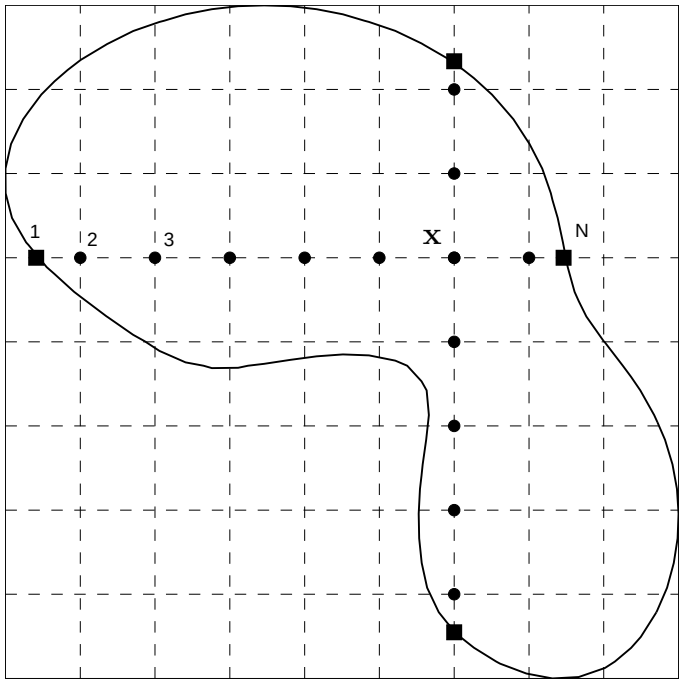

Figure 1: Domain discretization. The boundary and interior points used for constructing the IRBFN approximations at point $\mathbf{x}$ are highlighted.

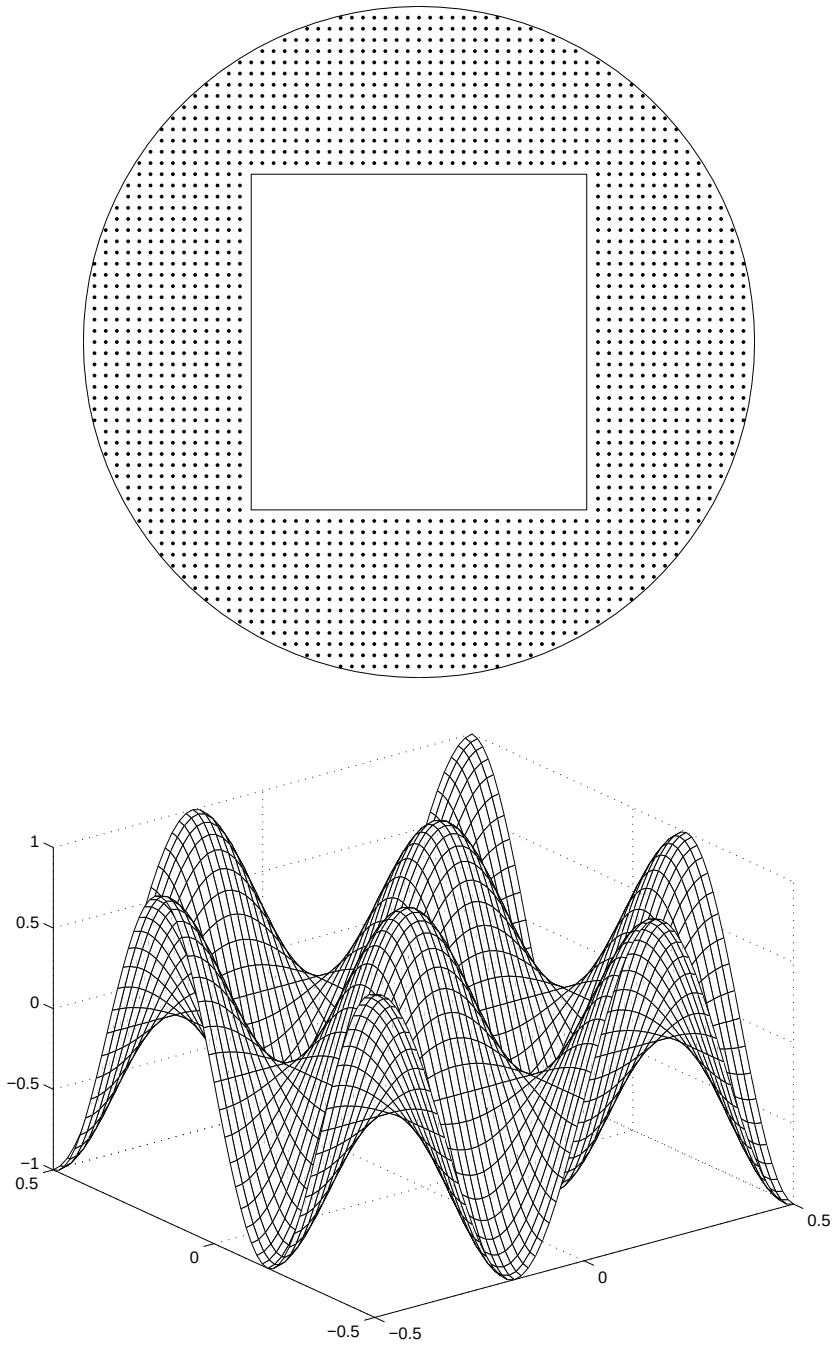

Figure 2: Example 1 (Poisson equation, Dirichlet boundary conditions): discretization and exact solution. Notice that the exact solution is plotted over the square covering the problem domain.

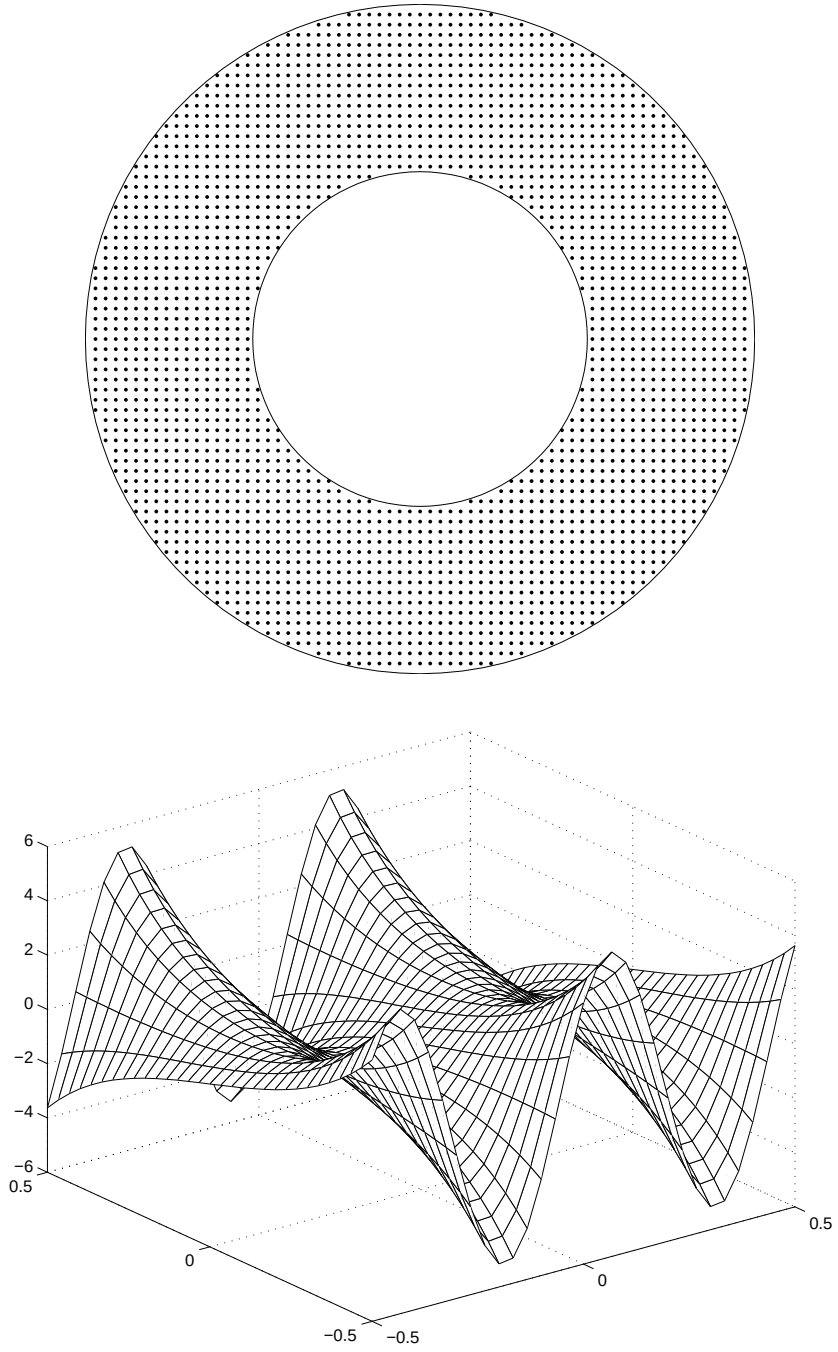

Figure 3: Example 2 (biharmonic equation, Dirichlet boundary conditions): discretization and exact solution. Notice that the exact solution is plotted over the square covering the problem domain.

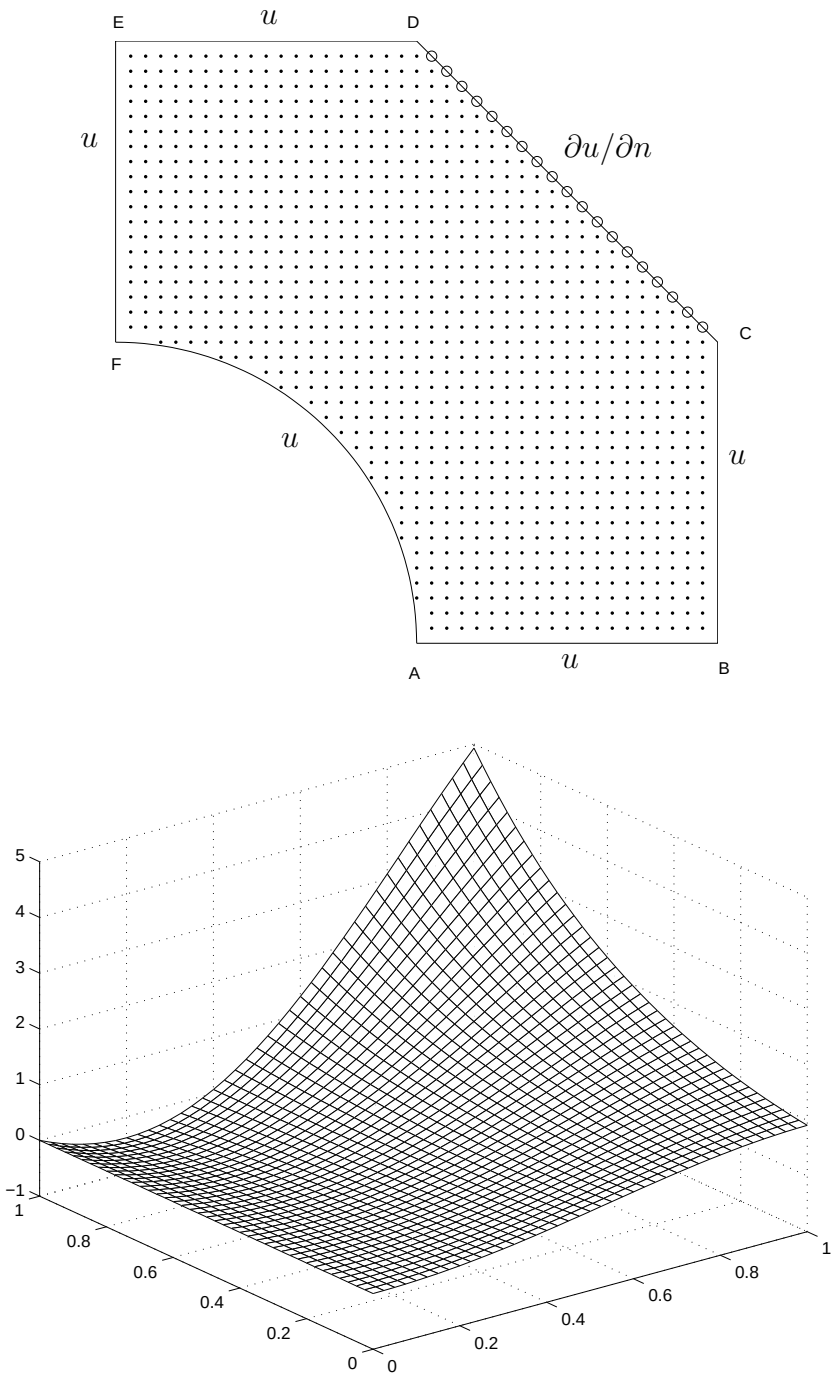

Figure 4: Example 3 (Poisson equation, Dirichlet and Neumann boundary conditions): discretization and exact solution. Notice that the exact solution is plotted over the square covering the problem domain.