A collocation method based on one-dimensional RBF interpolation scheme for solving PDEs

N. Mai-Duy and R.I. Tanner
School of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney, Sydney, Australia

Abstract

Purpose – To present a new collocation method for numerically solving partial differential equations (PDEs) in rectangular domains.

Design/methodology/approach – The proposed method is based on a Cartesian grid and a 1D integrated-radial-basis-function scheme. The employment of integration to construct the RBF approximations representing the field variables facilitates a fast convergence rate, while the use of a 1D interpolation scheme leads to considerable economy in forming the system matrix and improvement in the condition number of RBF matrices over a 2D interpolation scheme.

Findings – The proposed method is verified by considering several test problems governed by second- and fourth-order PDEs; very accurate solutions are achieved using relatively coarse grids.

Research limitations/implications – Only 1D and 2D formulations are presented, but we believe that extension to 3D problems can be carried out straightforwardly. Further, development is needed for the case of non-rectangular domains.

Originality/value – The contribution of this paper is a new effective collocation formulation based on RBFs for solving PDEs.

Keywords Differential equations, Numerical analysis, Boundary-elements methods

Paper type Research paper

1. Introduction

Radial-basis-function networks (RBFNs) have become one of the main fields of research in numerical analysis (Haykin, 1999). It has been proved that RBFNs have the property of universal approximation, i.e. an arbitrary continuous function can be approximated to a prescribed degree of accuracy by increasing the number of hidden neurons (Park and Sandberg, 1991). Madych and Nelson (1988, 1990) showed that the RBF interpolation scheme using multiquadrics (MQ) can offer exponential convergence rates/spectral accuracy. The application of MQ-RBFNs for the numerical solution of differential equations has received a great deal of attention over the past 15 years (Kansa, 1990; Fasshauer, 1997; Zerroukat *et al.*, 1998; Mai-Duy and Tran-Cong, 2001; Fedoseyev *et al.*, 2002; Power and Barraco, 2002; Cheng *et al.*, 2003; Larsson and Fornberg, 2003; Sarler *et al.*, 2004). These global methods had considerable success in

N. Mai-Duy wishes to thank the University of Sydney for a Sesquicentennial Post-doctoral Research Fellowship. Authors would like to thank the referees for their helpful comments and suggestions to improve the paper.

International Journal of Numerical Methods for Heat & Fluid Flow
Vol. 17 No. 2, 2007
pp. 165-186

0961-5539
DOI 10.1108/09615530710723948

solving a variety of scientific and engineering problems governed by differential equations.

However, it should be noted that the resultant RBF matrices are dense and their condition numbers grow rapidly as the number of nodes is increased. To resolve this problem, several attempts to use local RBF approximations (Shu *et al.*, 2003; Lee *et al.*, 2003) or to combine RBF and domain decomposition (Dubal, 1994; Kansa and Hon, 2000; Li and Hon, 2004) have been made. For a local-approximation-based approach, only a small region associated with a point, a node's region of influence, is activated to construct the RBF approximations for that point. The two most common shapes of an influence domain are circles and rectangles. Using a local procedure, the cost of computation can be modest; for example, the inversion involved in the construction process is conducted for a series of smaller matrices rather than for a large matrix. For a domain-decomposition-based approach, the given analysis domain is divided into a finite number of subdomains. The original problem can be then reformulated for each subdomain, leading to a series of coupled smaller subproblems. The solution is required to be continuous and smooth across the subdomain interfaces. This can be achieved either by overlapping regions (Li and Hon, 2004) or by common data points along the interfaces (Dubal, 1994). Using local approximations or domain decompositions have the following advantages:

- the resultant coefficient matrices are sparse/block-banded and hence their solutions are more efficient; and
- it can help prevent the rapid growth of the condition number of the system.

It should also be noted that the performance of the RBF scheme is strongly affected by the RBF width. To date, there is a lack of mathematical theory for finding appropriate values of the RBF width. In practice, the RBF width is chosen either by empirical approaches or by optimization techniques. The latter are expensive, especially for non-linear problems. Generally, the RBF scheme is more accurate, but less stable with increasing RBF-width.

Recently, an alternative approach based on integration to construct the RBF expressions for the interpolation of functions and the solution of differential equations was proposed (Mai-Duy and Tran-Cong, 2001, 2003). It was found that the indirect/integration-based RBFN approach (IRBFN) outperforms the direct/differentiation-based RBFN approach (DRBFN) regarding accuracy and convergence rate over a wide range of the RBF width. The improvement is attributable to the fact that integration is a smoothing operation and is more numerically stable.

In this study, a collocation method based on a 1D global IRBF interpolation scheme for the solution of 2D-partial differential equations (PDEs) is proposed. A rectangular domain of computation is discretized using a Cartesian grid (for the case of a non-rectangular domain, prior coordinate transformation can be conducted to produce a rectangular domain in the computational space). With the use of IRBFNs, one can make the approximating functions smoother, and generate additional coefficients (integration constants) that can be used to impose the governing equation on boundaries and/or to incorporate normal derivative boundary conditions more efficiently. On the other hand, with the use of a 1D global interpolation scheme, the construction of RBF approximations for a given point x involve only points that lie on lines intersected at x and parallel to x- and y-axes, rather than the whole set of grid points. This improves the conditioning of the system and requires less computational work than the case of using

a higher-dimensional scheme. One important feature of the present scheme is that it still maintains the advantages of a global high-order method such as the capability to achieve a high degree of accuracy using relatively low numbers of nodes. It was reported that the rapid growth of the conditioning of the system matrix limits the use of a global 2D-RBF collocation method to a few hundred interpolation points. With the present approach, much larger numbers of nodes (e.g. up to 10,201 nodes in this study) can be employed. Numerical results show that the proposed method achieves a high degree of accuracy.

The remainder of the paper is organized as follows. The proposed 1D integrated-radial-basis-function (1D-IRBF) collocation method for the solution of second- and fourth-order PDEs is presented and verified in sections 2 and 3, respectively. The method is then applied to simulate the thermally-driven cavity flow in section 4. Section 5 gives some concluding remarks.

2. Second-order PDEs

2.1 The IRBF formulation

The domain of interest is 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 nodes in the x- and y-directions, respectively. The dependent variable u and its derivatives are approximated using a 1D-IRBF interpolation scheme. It should be indicated that the 1D interpolation scheme uses only N_x or N_y nodes (instead of N_xN_y nodes) to construct the approximations for a given point, resulting in considerable economy when compared with an earlier 2D-IRBF interpolation scheme reported in Mai-Duy and Tran-Cong (2005). The construction process involves two steps, to use :

(1) IRBFNs to approximate the variable u and its derivatives along a straight line; and

(2) Kronecker tensor products to construct the approximations for derivatives over a 2D-domain.

2.1.1 One-dimensional formulation. Consider a line in a Cartesian grid, e.g. the line runs parallel to the x-axis. The dependent variable u along this line is sought in the IRBF form. The second-order derivative of u is decomposed into RBFs; the RBF network is then integrated twice to obtain the expressions for the first-order derivative of u and the solution u itself:

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

$$\frac{\partial u(x)}{\partial x} = \sum_{i=1}^{N_x} w^{(i)} H_{[1]}^{(i)}(x) + c_1, \tag{2}$$

$$u(x) = \sum_{i=1}^{N_x} w^{(i)} H_{[0]}^{(i)}(x) + c_1 x + c_2, \tag{3}$$

where:

$\{w^{(i)}\}_{i=1}^{N_x}$ are RBF weights to be determined;

$\{g^{(i)}(x)\}_{i=1}^{N_x}$ are known RBFs;

$H_{[1]}(x) = \int H_{[2]}(x)\mathrm{d}x$, $H_{[0]}(x) = \int H_{[1]}(x)\mathrm{d}x$; and c_1 and c_2 are integration constants. Here, it is referred to as a second-order 1D-IRBF scheme, denoted by IRBF-2. It is more convenient to work in physical space than in network-weight space. The RBF coefficients including two integration constants can be transformed into the meaningful nodal variable values, based on the following equations:

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

$$u(x^{(2)}) = \sum_{i=1}^{N_x} w^{(i)} H_{[0]}^{(i)}(x^{(2)}) + c_1 x^{(2)} + c_2, \tag{5}$$

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

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

The above system can be written in a matrix form:

$$\hat{u} = \mathscr{H} \begin{pmatrix} \hat{w} \\ \hat{c} \end{pmatrix}, \tag{7}$$

where $\mathscr{H}$ is an $N_x \times (N_x + 2)$ matrix: $\hat{u} = (u^{(1)}, u^{(2)}, \ldots, u^{(N_x)})^{\mathrm{T}}$, $\hat{w} = (w^{(1)}, w^{(2)}, \ldots, w^{(N_x)})^{\mathrm{T}}$, and $\hat{c} = (c_1, c_2)^{\mathrm{T}}$.

One difference between 1D integrated- and differentiated-RBF interpolation schemes is that the former possesses a larger set of expansion coefficients owing to the presence of two integration constants c_1 and c_2. The present conversion schemes can be thus developed into two directions.

2.1.2 Non-square conversion matrix (NSCM). The direct use of equation (7) leads to an underdetermined system of equations. The associated matrix is referred to as the conversion matrix, denoted by $\mathscr{C}$ ($\mathscr{C} = \mathscr{H}$). The pseudo inverse of $\mathscr{C}$ can be found using the SVD technique. 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:

$$\hat{u} = \mathscr{H} \begin{pmatrix} \hat{w} \\ \hat{c} \end{pmatrix} = \mathscr{C} \begin{pmatrix} \hat{w} \\ \hat{c} \end{pmatrix}, \tag{8}$$

$$\begin{pmatrix} \hat{w} \\ \hat{c} \end{pmatrix} = \mathscr{C}^{-1} \hat{u}. \tag{9}$$

2.1.3 Square conversion matrix (SCM). One can add two additional equations of the form:

$$f_1 = \sum_{i=1}^{N_x} w^{(i)} H^{(i)}_{[\dagger]} + c_1\alpha_1 + c_2\beta_1, \tag{10}$$

$$f_2 = \sum_{i=1}^{N_x} w^{(i)} H^{(i)}_{[\ddagger]} + c_1\alpha_2 + c_2\beta_2, \tag{11}$$

or:

$$\hat{f} = \mathscr{K}\begin{pmatrix} \hat{w} \\ \hat{c} \end{pmatrix} \tag{12}$$

to equation (7). The conversion system can be written as:

$$\begin{pmatrix} \hat{u} \\ \hat{f} \end{pmatrix} = \begin{bmatrix} \mathscr{H} \\ \mathscr{K} \end{bmatrix} \begin{pmatrix} \hat{w} \\ \hat{c} \end{pmatrix} = \mathscr{C}\begin{pmatrix} \hat{w} \\ \hat{c} \end{pmatrix}. \tag{13}$$

The conversion matrix $\mathscr{C}$ becomes square and its inverse can be computed by using the LU decomposition:

$$\begin{pmatrix} \hat{w} \\ \hat{c} \end{pmatrix} = \mathscr{C}^{-1}\begin{pmatrix} \hat{u} \\ \hat{f} \end{pmatrix}. \tag{14}$$

It is widely observed that the RBF approximations have a tendency to produce larger errors near boundaries. Fedoseyev *et al.* (2002) developed an MQ-RBF collocation method with PDE collocation on the boundaries; numerical results indicated a considerable improvement in accuracy. Motivated by these results, the two extra equations (10) and (11) can be utilized here to satisfy the governing equation at both ends of the line: $x^{(1)}$ and $x^{(N_x)}$. If the Neumann boundary condition rather than the Dirichlet condition is given, these equations can be used to represent normal derivative boundary conditions; imposition of the governing equation on the boundaries is carried out at a later stage of constructing a system matrix. All f_i, $H_{[\cdot]}$, α_i and β_i with $i = (1, 2)$ in equations (10) and (11) are known quantities. However, their explicit forms/values depend on the problem to be solved; they will be presented in some detail in section "Numerical results". A distinct difference between the differentiation- and integration-based formulations is that the governing equation is forced to be satisfied exactly on the boundaries by means of fictitious points inside/outside domain for the former and by integration constants for the latter.

In the following discussion, only the SCM version is considered since the NSCM system (8) can be obtained from the SCM system (13) by simply setting matrix $\mathscr{K}$ and vector $\hat{f}$ to null.

By substituting equation (14) into equations (1) and (2), the second- and first-order derivatives of the variable u will be expressed in terms of nodal variable values:

$$\frac{\partial^2 u(x)}{\partial x^2} = \left(H^{(1)}_{[2]}(x), H^{(2)}_{[2]}(x), \ldots, H^{(N_x)}_{[2]}(x), 0, 0\right)\mathscr{C}^{-1}\begin{pmatrix} \hat{u} \\ \hat{f} \end{pmatrix}, \tag{15}$$

$$\frac{\partial u(x)}{\partial x} = \left(H_{[1]}^{(1)}(x), H_{[1]}^{(2)}(x), \ldots, H_{[1]}^{(N_x)}(x), 1, 0\right)\mathscr{C}^{-1}\begin{pmatrix}\hat{u}\\ \hat{f}\end{pmatrix}, \tag{16}$$

or:

$$\frac{\partial^2 u(x)}{\partial x^2} = \mathscr{D}_{2x}\hat{u} + k_{2x}, \tag{17}$$

$$\frac{\partial u(x)}{\partial x} = \mathscr{D}_{1x}\hat{u} + k_{1x}, \tag{18}$$

where k_{2x} and k_{1x} are scalars whose values depend on f_1 and f_2.

Application of equations (17) and (18) to every collocation point on the line yields:

$$\widehat{\frac{\partial^2 u}{\partial x^2}} = \hat{\mathscr{D}}_{2x}\hat{u} + \hat{k}_{2x}, \tag{19}$$

$$\widehat{\frac{\partial u}{\partial x}} = \hat{\mathscr{D}}_{1x}\hat{u} + \hat{k}_{1x}, \tag{20}$$

where $\hat{\mathscr{D}}_{2x}$ and $\hat{\mathscr{D}}_{1x}$ are known matrices of dimension $N_x \times N_x$, and $\hat{k}_{2x}$ and $\hat{k}_{1x}$ are known vectors of length N_x.

Similarly, along a vertical line, the values of the second- and first-order derivatives of u with respect to y at the collocation points can be given by:

$$\widehat{\frac{\partial^2 u}{\partial y^2}} = \hat{\mathscr{D}}_{2y}\hat{u} + \hat{k}_{2y}, \tag{21}$$

$$\widehat{\frac{\partial u}{\partial y}} + \hat{\mathscr{D}}_{1y}\hat{u} + \hat{k}_{1y}. \tag{22}$$

2.1.4 Two-dimensional formulation. Assuming that the grid points are numbered from bottom to top and from left to right, one can write the values of the derivatives of u over the whole domain by using Kronecker tensor products as follows:

$$\widetilde{\frac{\partial^2 u}{\partial x^2}} = (\hat{\mathscr{D}}_{2x}\otimes\mathscr{I}_y)\tilde{u} + \tilde{k}_{2x} = \tilde{\mathscr{D}}_{2x}\tilde{u} + \tilde{k}_{2x}, \tag{23}$$

$$\widetilde{\frac{\partial u}{\partial x}} = (\hat{\mathscr{D}}_{1x}\otimes\mathscr{I}_y)\tilde{u} + \tilde{k}_{1x} = \tilde{\mathscr{D}}_{1x}\tilde{u} + \tilde{k}_{1x}, \tag{24}$$

$$\widetilde{\frac{\partial^2 u}{\partial y^2}} = (\mathscr{I}_x\otimes\hat{\mathscr{D}}_{2y})\tilde{u} + \tilde{k}_{2y} = \tilde{\mathscr{D}}_{2y}\tilde{u} + \tilde{k}_{2y}, \tag{25}$$

$$\widetilde{\frac{\partial u}{\partial y}} = (\mathscr{I}_x\otimes\hat{\mathscr{D}}_{1y})\tilde{u} + \tilde{k}_{1y} = \tilde{\mathscr{D}}_{1y}\tilde{u} + \tilde{k}_{1y}, \tag{26}$$

where $\mathscr{I}_x$ and $\mathscr{I}_y$ are the identity matrices of dimension $N_x \times N_x$ and $N_y \times N_y$, respectively; $\tilde{k}_{2x}, \tilde{k}_{1x}, \tilde{k}_{2y}$ and $\tilde{k}_{1y}$ are known vectors of length $N_x N_y$; $\tilde{\mathscr{D}}_{2x}, \tilde{\mathscr{D}}_{1x}, \tilde{\mathscr{D}}_{2y}$ and $\tilde{\mathscr{D}}_{1y}$ are known matrices of dimension $N_x N_y \times N_x N_y$; and $\tilde{u} = (u^{(1)}, u^{(2)}, \ldots, u^{(N_x N_y)})^{\mathrm{T}}$.

2.2 Numerical results

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

$$N_{\mathrm{e}} = \frac{\sqrt{\sum_{i=1}^{N} (u_{\mathrm{e}}^{(i)} - u^{(i)})^2}}{\sqrt{\sum_{i=1}^{N} (u_{\mathrm{e}}^{(i)})^2}}, \tag{27}$$

where N is the number of collocation points; and u_{e} and u are the exact and computed solutions, respectively. The present study employs MQ whose form is:

$$g^{(i)}(x) = \sqrt{\|\boldsymbol{x} - \boldsymbol{c}^{(i)}\|^2 + a^{(i)2}}, \tag{28}$$

where $\mathbf{c}$ is the centre, a is the RBF width and $\|\cdot\|$ denotes a Euclidean norm. The width of the ith MQ-RBF, $a^{(i)}$, is simply chosen to be the minimum distance from the ith centre to its neighbours.

2.2.1 Problem 1. Consider the following Poisson equation:

$$\nabla^2 u = b = -8\pi^2 \sin(2\pi x)\sin(2\pi y). \tag{29}$$

An approximate solution is sought in the unit square domain, $0 \leq x, y \leq 1$. The exact solution is given by:

$$u_{\mathrm{e}} = \sin(2\pi x)\sin(2\pi y). \tag{30}$$

Ten uniform grids, $11 \times 11, 21 \times 21, \ldots, 101 \times 101$, are employed. A comparative study of the accuracy of the present 1D-IRBF method between the two versions, NSCM and SCM, is carried out for two different types of the boundary condition, namely $\{u\}$ and $\{u, \partial u/\partial n\}$.

2.2.2 Dirichlet boundary condition. Along the two vertical sides, the extra information f_i, which is used for computing the derivatives of u with respect to x, is taken to be:

$$f_1(x = 0, y) = \frac{\partial^2 u}{\partial x^2}(0, y) = b(0, y) - \frac{\partial^2 u}{\partial y^2}(0, y) = 0, \tag{31}$$

$$f_2(x = 1, y) = \frac{\partial^2 u}{\partial x^2}(1, y) = b(1, y) - \frac{\partial^2 u}{\partial y^2}(1, y) = 0, \tag{32}$$

or:

$$\sum_{i=1}^{N_x} w^{(i)} H_{[2]}^{(i)}(0) = 0, \tag{33}$$

$$\sum_{i=1}^{N_x} w^{(i)} H_{[2]}^{(i)}(1) = 0. \tag{34}$$

Similarly, along the two horizontal sides, the extra information f_i, which is used for computing the derivatives of u with respect to y, is defined as:

$$f_1(x, y = 0) = \frac{\partial^2 u}{\partial y^2}(x, 0) = b(x, 0) - \frac{\partial^2 u}{\partial x^2}(x, 0) = 0, \tag{35}$$

$$f_2(x, y = 1) = \frac{\partial^2 u}{\partial y^2}(x, 1) = b(x, 1) - \frac{\partial^2 u}{\partial x^2}(x, 1) = 0. \tag{36}$$

Applying the governing equation (29) at the interior points (ip) yields:

$$(\tilde{\mathscr{D}}_{2x} + \tilde{\mathscr{D}}_{2y})_{\text{ip}}\tilde{u} = (b)_{\text{ip}} - (\tilde{k}_{2x})_{\text{ip}} - (\tilde{k}_{2y})_{\text{ip}}. \tag{37}$$

Making use of the Dirichlet boundary condition of the problem, a determinate system of equations is obtained which can be solved by Gaussian elimination.

Results concerning the relative L_2 error are shown in Figure 1, which indicates the rapid improvement in accuracy with increasing density for both versions. The SCM version is more accurate, but converges slightly slower than the NSCM version.

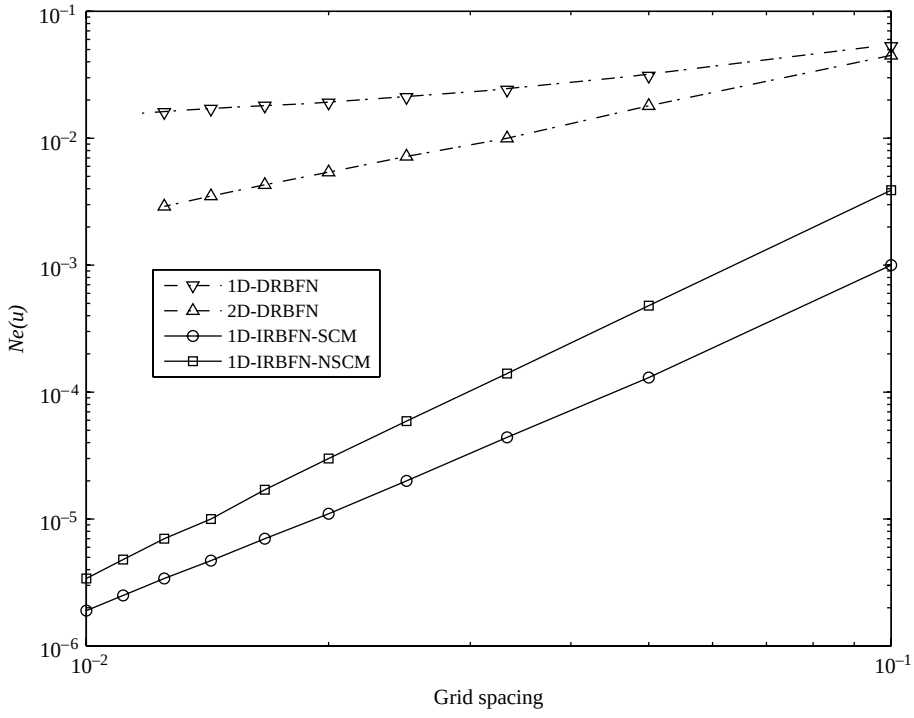

Figure 1.
Second-order PDE, Dirichlet boundary conditions: relative L_2 errors obtained by the proposed 1D-IRBF method and the classical DRBFN method

Notes: The former outperforms the latter regarding accuracy and convergence. It is noted that not only 2D- but also 1D-interpolation schemes are employed for the DRBFN method

The present results are also compared with those obtained by the classical DRBFN method (Figure 1). The DRBFN method that is based on a 1D interpolation scheme is also implemented here. The 1D- and 2D-DRBF methods use the same network parameters (e.g. the number of collocation points, their locations and the RBF widths) as the proposed 1D-IRBF method. It can be seen that the proposed method outperforms the 1D- and 2D-DRBF methods regarding accuracy and convergence.

In comparison with the 2D-IRBF method, Table I shows that for each version (SCM and NSCM), the 2D-IRBF method is more accurate than the 1D-IRBF method. It is interesting to note that the accuracy of the 1D-IRBF method with SCM is superior to that of the 2D-IRBF method with NSCM. The 2D method requires much more computational effort than the 1D method. For example, the inversion is conducted for matrices of dimensions about $N_x \times N_x$ or $N_y \times N_y$ for the 1D method, but of dimensions $N_xN_y \times N_xN_y$ for the 2D method.

2.2.3 Dirichlet and Neumann boundary conditions. The Neumann boundary conditions are imposed on the two vertical sides, while the Dirichlet conditions are specified along the two horizontal sides. Special attention here is given to the implementation of the Neumann boundary condition. The two additional equations (10) and (11) can take the form:

$$f_1(x=0,y) = \frac{\partial u}{\partial x}(0,y) = 2\pi \sin(2\pi y), \tag{38}$$

$$f_2(x=1,y) = \frac{\partial u}{\partial x}(1,y) = 2\pi \sin(2\pi y), \tag{39}$$

$$\sum_{i=1}^{N_x} w^{(i)} H^{(i)}_{[1]}(0) + c_1 = 2\pi \sin(2\pi y), \tag{40}$$

$$\sum_{i=1}^{N_x} w^{(i)} H^{(i)}_{[1]}(1) + c_1 = 2\pi \sin(2\pi y). \tag{41}$$

For the SCM version, the governing equation (29) is forced to be satisfied not only at the ip but also at the $2(N_y - 2)$ boundary points on the two vertical sides. For the NSCM

Grid	2D-IRBFN NSCM	2D-IRBFN SCM	1D-IRBFN NSCM	1D-IRBFN SCM
11 × 11	3.6 (−3)	8.8 (−4)	3.9 (−3)	1.0 (−3)
21 × 21	3.6 (−4)	8.6 (−5)	4.8 (−4)	1.3 (−4)
31 × 31	9.3 (−5)	2.1 (−5)	1.4 (−4)	4.4 (−5)
41 × 41	3.5 (−5)	8.0 (−6)	5.9 (−5)	2.0 (−5)
51 × 51	1.6 (−5)	3.7 (−6)	3.0 (−5)	1.1 (−5)
61 × 61	8.7 (−6)	2.1 (−6)	1.7 (−5)	7.0 (−6)
71 × 71	5.1 (−6)	1.5(−6)	1.0 (−5)	4.7 (−6)

Notes: For each version (SCM and NSCM), the 2D-IRBF method is more accurate than 1D-IRBF method. It is interesting to note that the accuracy of the 1D-IRBF method with SCM is superior to that of the 2D-IRBF method with NSCM. The 2D-IRBF method requires much more computational effort than the 1D-IRBF method. a $(-b)$ means $a \times 10^{-b}$

Table I. Second-order PDE, Dirichlet boundary conditions: relative L_2 errors obtained by the 2D- and 1D-IRBF methods

version, the governing equation (29) is applied at the ip and the Neumann boundary conditions are enforced explicitly by adding some additional equations to the system. After introducing the Dirichlet boundary condition of the problem, for both versions, a square algebraic system of dimension $(N_x N_y \times 2N_x) \times (N_x N_y \times 2N_x)$ is obtained for the unknown nodal values of u inside the domain and on the boundaries of the Neumann boundary condition.

Figure 2 shows the convergence behaviour of the present method using NSCM and SCM. It is clear that both versions have essentially the same convergence rates, but the latter produces a higher degree of accuracy. Imposition of the Neumann boundary condition through the conversion process is recommended for use because of its superior accuracy and its straightforward implementation.

3. Fourth-order PDEs

One distinct feature of problems governed by fourth-order PDEs is that there are two required values of the variable at each boundary point. For the conventional implementation of multiple boundary conditions, fictitious points are introduced outside/inside the domain (the fictitious-point technique) or the differential equations are collocated at a smaller number of ip (the node-reduction technique). The interested reader is referred to references Shu (2000), Shu *et al.* (2000) and Chen *et al.* (2000) for more detailed discussion of this topic. In the present work, the imposition of multiple boundary conditions is simply conducted by means of integration constants; its performance is compared with that of the node-reduction technique. The details of

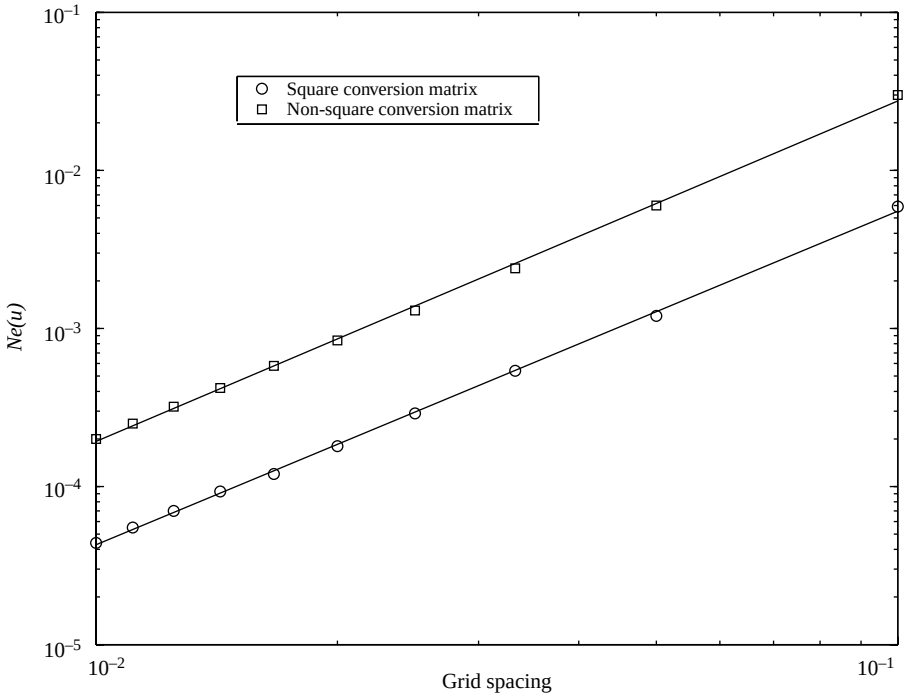

Figure 2.
Second-order PDE, Dirichlet and Neumann boundary conditions: comparison of the accuracy of the 1D-IRBF method between the two versions

implementing multiple boundary conditions using integration constants were reported in Mai-Duy (2005) and Mai-Duy and Tanner (2005).

For the sake of simplicity, the following discussion is given to the case of biharmonic equations of the form:

$$\frac{\partial^4 u}{\partial x^4} + 2\frac{\partial^4 u}{\partial x^2 \partial y^2} + \frac{\partial^4 u}{\partial y^4} = b(x,y) \tag{42}$$

in the rectangular domain, subject to the boundary conditions u and $\partial u/\partial n$ along the boundaries.

The process of deriving the 1D-IRBF formulation for fourth-order PDEs is similar to that for second-order PDEs. However, the corresponding equations involve more terms. Furthermore, special attention needs to be paid to the treatment of mixed partial derivatives, e.g. $\partial^4 u/\partial x^2 \partial y^2$. Notations used in this section and in the previous one have similar meanings.

3.1 The IRBF formulation

3.1.1 One-dimensional formulation. The variable u and its derivatives along a grid line that runs parallel to the $x$axis can be approximated by:

$$\frac{\partial^4 u(x)}{\partial x^4} = \sum_{i=1}^{N_x} w^{(i)} g^{(i)}(x) = \sum_{i=1}^{N_x} w^{(i)} H^{(i)}_{[4]}(x), \tag{43}$$

$$\frac{\partial^3 u(x)}{\partial x^3} = \sum_{i=1}^{N_x} w^{(i)} H^{(i)}_{[3]}(x) + c_1, \tag{44}$$

$$\frac{\partial^2 u(x)}{\partial x^2} = \sum_{i=1}^{N_x} w^{(i)} H^{(i)}_{[2]}(x) + c_1 x + c_2, \tag{45}$$

$$\frac{\partial u(x)}{\partial x} = \sum_{i=1}^{N_x} w^{(i)} H^{(i)}_{[1]}(x) + c_1 \frac{x^2}{2} + c_2 x + c_3, \tag{46}$$

$$u(x) = \sum_{i=1}^{N_x} w^{(i)} H^{(i)}_{[0]}(x) + c_1 \frac{x^3}{6} + c_2 \frac{x^2}{2} + c_3 x + c_4, \tag{47}$$

in which the fourth-order derivative of the variable 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:

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

$$\hat{w} = \begin{pmatrix} w^{(1)} \\ w^{(2)} \\ \cdots \\ w^{(N_x)} \end{pmatrix}, \quad \hat{c} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix}, \quad \hat{u} = \begin{pmatrix} u^{(1)} \\ u^{(2)} \\ \cdots \\ u^{(N_x)} \end{pmatrix}$$

3.1.2 Non-square conversion matrix. The unknown expansion coefficients can be converted into the nodal variable values according to the following relation:

$$\hat{u} = \mathscr{H} \begin{pmatrix} \hat{w} \\ \hat{c} \end{pmatrix} = \mathscr{C} \begin{pmatrix} \hat{w} \\ \hat{c} \end{pmatrix}, \tag{48}$$

$$\begin{pmatrix} \hat{w} \\ \hat{c} \end{pmatrix} = \mathscr{C}^{-1} \hat{u}. \tag{49}$$

3.1.3 Square conversion matrix. The presence of four integration constants allows the addition of four extra equations to the conversion system. Using information on the governing equation and normal derivative boundary conditions at both ends of the line, the additional matrix and vector can be generated as follows:

$$\mathscr{K} = \begin{bmatrix} H_{[1]}^{(1)}(x^{(1)}) & H_{[1]}^{(2)}(x^{(1)}) & \cdots & H_{[1]}^{(N_x)}(x^{(1)}) & x^{(1)2}/2 & x^{(1)} & 1 & 0 \\ H_{[1]}^{(1)}(x^{(N_x)}) & H_{[1]}^{(2)}(x^{(N_x)}) & \cdots & H_{[1]}^{(N_x)}(x^{(N_x)}) & x^{(N_x)2}/2 & x^{(N_x)} & 1 & 0 \\ H_{[4]}^{(1)}(x^{(1)}) & H_{[4]}^{(2)}(x^{(1)}) & \cdots & H_{[4]}^{(N_x)}(x^{(1)}) & 0 & 0 & 0 & 0 \\ H_{[4]}^{(1)}(x^{(N_x)}) & H_{[4]}^{(2)}(x^{(N_x)}) & \cdots & H_{[4]}^{(N_x)}(x^{(N_x)}) & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\hat{f} = \begin{pmatrix} \frac{\partial u}{\partial x}(x^{(1)}) \\ \frac{\partial u}{\partial x}(x^{(N_x)}) \\ b(x^{(1)}) - 2\frac{\partial^4 u}{\partial x^2 y^2}(x^{(1)}) - \frac{\partial^4 u}{\partial y^4}(x^{(1)}) \\ b(x^{(N_x)}) - 2\frac{\partial^4 u}{\partial x^2 y^2}(x^{(N_x)}) - \frac{\partial^4 u}{\partial y^4}(x^{(N_x)}) \end{pmatrix}.$$

The conversion process thus becomes:

$$\begin{pmatrix} \hat{u} \\ \hat{f} \end{pmatrix} = \begin{bmatrix} \mathscr{H} \\ \mathscr{K} \end{bmatrix} \begin{pmatrix} \hat{w} \\ \hat{c} \end{pmatrix} = \mathscr{C} \begin{pmatrix} \hat{w} \\ \hat{c} \end{pmatrix}, \tag{50}$$

$$\begin{pmatrix} \hat{w} \\ \hat{c} \end{pmatrix} = \mathscr{C}^{-1} \begin{pmatrix} \hat{u} \\ \hat{f} \end{pmatrix}. \tag{51}$$

Substitution of equation (51) into equations (43)-(46) yields:

$$\frac{\partial^4 u(x)}{\partial x^4} = \left(H_{[4]}^{(1)}(x), H_{[4]}^{(2)}(x), \ldots, H_{[4]}^{(N_x)}(x), 0, 0, 0, 0 \right) \mathscr{C}^{-1} \begin{pmatrix} \hat{u} \\ \hat{f} \end{pmatrix}, \tag{52}$$

$$\frac{\partial^3 u(x)}{\partial x^3} = \left(H_{[3]}^{(1)}(x), H_{[3]}^{(2)}(x), \ldots, H_{[3]}^{(N_x)}(x), 1, 0, 0, 0 \right) \mathscr{C}^{-1} \begin{pmatrix} \hat{u} \\ \hat{f} \end{pmatrix}, \tag{53}$$

$$\frac{\partial^2 u(x)}{\partial x^2} = \left(H_{[2]}^{(1)}(x), H_{[2]}^{(2)}(x), \ldots, H_{[2]}^{(N_x)}(x), x, 1, 0, 0 \right) \mathscr{C}^{-1} \begin{pmatrix} \hat{u} \\ \hat{f} \end{pmatrix}, \tag{54}$$

$$\frac{\partial u(x)}{\partial x} = \left(H_{[1]}^{(1)}(x), H_{[1]}^{(2)}(x), \ldots, H_{[1]}^{(N_x)}(x), \frac{x^2}{2}, x, 1, 0 \right) \mathscr{C}^{-1} \begin{pmatrix} \hat{u} \\ \hat{f} \end{pmatrix}. \tag{55}$$

The values of the ith-order derivative of u ($i = \{1, 2, 3, 4\}$) at the grid points along a horizontal line can be computed by:

$$\widehat{\frac{\partial^i u}{\partial x^i}} = \hat{\mathscr{D}}_{ix}^{\mathrm{IV}} \begin{pmatrix} \hat{u} \\ \hat{f} \end{pmatrix}, \quad i = \{1, 2, 3, 4\}, \tag{56}$$

where the superscript IV is used to indicate that $\hat{\mathscr{D}}_{ix}$ is obtained using the IRBF-4 scheme; and $\hat{\mathscr{D}}_{ix}^{\mathrm{IV}}$ is a known matrix of dimension $N_x \times (N_x + 4)$; and:

$$\widehat{\frac{\partial^i u}{\partial x^i}} = \left(\frac{\partial^i u^{(1)}}{\partial x^i}, \frac{\partial^i u^{(2)}}{\partial x^i}, \ldots, \frac{\partial^i u^{(N_x)}}{\partial x^i} \right)^{\mathrm{T}}.$$

Expression (56) can be rewritten as:

$$\widehat{\frac{\partial^i u}{\partial x^i}} = \hat{\mathscr{D}} 1_{ix}^{\mathrm{IV}} \hat{u} + \hat{\mathscr{D}} 2_{ix}^{\mathrm{IV}} \hat{f}, \tag{57}$$

where $\hat{\mathscr{D}} 1_{ix}^{\mathrm{IV}}$ and $\hat{\mathscr{D}} 2_{ix}^{\mathrm{IV}}$ are matrices that are formed by the first N_x columns and the last four columns of the matrix $\hat{\mathscr{D}}_{ix}^{\mathrm{IV}}$, respectively. Unlike the case of second-order PDEs, the extra information vector $\hat{f}$ (components f_3 and f_4) contains some unknown values – the mixed partial derivative $\partial^4 u / \partial x^2 \partial y^2$ at the two boundary points. The treatment is as follows. These unknown values will be replaced by linear combinations of nodal values

of the variable u (the detailed expression of $\partial^4 u/\partial x^2 \partial y^2$ will be given in section 3.1.4). In the same way, one can obtain the vector of the values of the ith-order derivative of u with respect to y along a vertical line.

3.1.4 Two-dimensional formulation. Making use of the following expression:

$$\frac{\partial^4 u}{\partial^2 x \partial^2 y} = \frac{1}{2}\left[\frac{\partial^2}{\partial x^2}\left(\frac{\partial^2 u}{\partial y^2}\right) + \frac{\partial^2}{\partial y^2}\left(\frac{\partial^2}{\partial x^2}\right)\right], \tag{58}$$

the mixed fourth-order partial derivative can be computed by the 1D-IRBF-2 scheme with the extra information f_i being the values of the first-order derivatives at the boundary points which are given or can be computed easily. Expression (58) can be rewritten as:

$$\frac{\widetilde{\partial^4 u}}{\partial x^2 \partial y^2} = \frac{1}{2}\left(\tilde{\mathscr{D}}_{2x}\frac{\widetilde{\partial^2 u}}{\partial y^2} + \tilde{k}_{2x} + \tilde{\mathscr{D}}_{2y}\frac{\widetilde{\partial^2 u}}{\partial x^2} + \tilde{k}_{2y}\right), \tag{59}$$

$$= \frac{1}{2}(\tilde{\mathscr{D}}_{2x}\tilde{\mathscr{D}}_{2y} + \tilde{\mathscr{D}}_{2y}\tilde{\mathscr{D}}_{2x})\tilde{u} + \tilde{k}_{4xy}, \tag{60}$$

$$= \tilde{\mathscr{D}}_{4xy}\tilde{u} + \tilde{k}_{4xy}, \tag{61}$$

where $\tilde{\mathscr{D}}_{2x}$ and $\tilde{\mathscr{D}}_{2y}$ are known matrices obtained from equations (23) and (25); $\tilde{k}_{4xy}$ is a known vector of length $N_x N_y$; $\tilde{\mathscr{D}}_{4xy}$ is a known matrix of dimension $N_x N_y \times N_x N_y$; and:

$$\frac{\widetilde{\partial^4 u}}{\partial x^2 \partial y^2} = \left(\frac{\partial^4 u}{\partial x^2 \partial y^2}^{(1)}, \frac{\partial^4 u}{\partial x^2 \partial y^2}^{(2)}, \ldots, \frac{\partial^4 u}{\partial x^2 \partial y^2}^{(N_x N_y)}\right)^{\mathrm{T}}.$$

It can be seen that the mixed fourth-order partial derivative of the variable u is now expressed in terms of nodal variable values. Using these results to represent the components f_3 and f_4, one can now extend the approximations for $\partial^i u/\partial x^i$ (equation (57)) to a 2D domain by means of Kronecker tensor products. Their final forms can be written as:

$$\frac{\widetilde{\partial^i u}}{\partial x^i} = \tilde{\mathscr{D}}_{ix}^{\mathrm{IV}}\tilde{u} + \tilde{k}_{ix}^{\mathrm{IV}} \tag{62}$$

$$\frac{\widetilde{\partial^i u}}{\partial y^i} = \tilde{\mathscr{D}}_{iy}^{\mathrm{IV}}\tilde{u} + \tilde{k}_{iy}^{\mathrm{IV}} \tag{63}$$

where $i = \{1, 2, 3, 4\}$; $\tilde{k}_{ix}^{\mathrm{IV}}$ and $\tilde{k}_{iy}^{\mathrm{IV}}$ are known vectors of length $N_x N_y$; and $\tilde{\mathscr{D}}_{ix}^{\mathrm{IV}}$ and $\tilde{\mathscr{D}}_{iy}^{\mathrm{IV}}$ are known matrices of dimension $N_x N_y \times N_x N_y$.

3.2 Numerical results

A biharmonic Dirichlet problem is considered here to verify the 1D-IRBF formulation.

3.2.1 Problem 2. Consider the biharmonic equation:

$$\nabla^4 u = 64\pi^4 \sin(2\pi x)\sin(2\pi y) \tag{64}$$

in the domain $0 \leq x,y \leq 1$, subject to the boundary conditions:

$$u = 0 \quad \text{along the boundaries,}$$

$$\frac{\partial u}{\partial x} = 2\pi \sin(2\pi y) \quad x = 0, \;\; x = 1$$

$$\frac{\partial u}{\partial y} = 2\pi \sin(2\pi x) \quad y = 0, \;\; y = 1$$

The exact solution is given by:

$$v_e(x, y) = \sin(2\pi x)\sin(2\pi y). \tag{65}$$

A number of uniform grids, $6 \times 6, 11 \times 11, \ldots,$ and 71×71, are employed. In forming the system matrix, the SCM version allows the biharmonic equation (64) to be collocated at every interior point. For the NSCM version, it is necessary to reduce the number of ip used for equation (64) in order to impose the boundary conditions $\partial u/\partial n$; these ip chosen here for collocating the biharmonic equation are (x_i, y_j) with $3 \le i \le N_x - 2$ and $3 \le j \le N_y - 2$. A comparison of the accuracy of the present method between the two versions is shown in Figure 3. The NSCM version does not perform as well, probably due to the fact that the governing equation is not forced to be satisfied exactly at every interior point. For the SCM version, a high degree of accuracy and fast convergence are achieved.

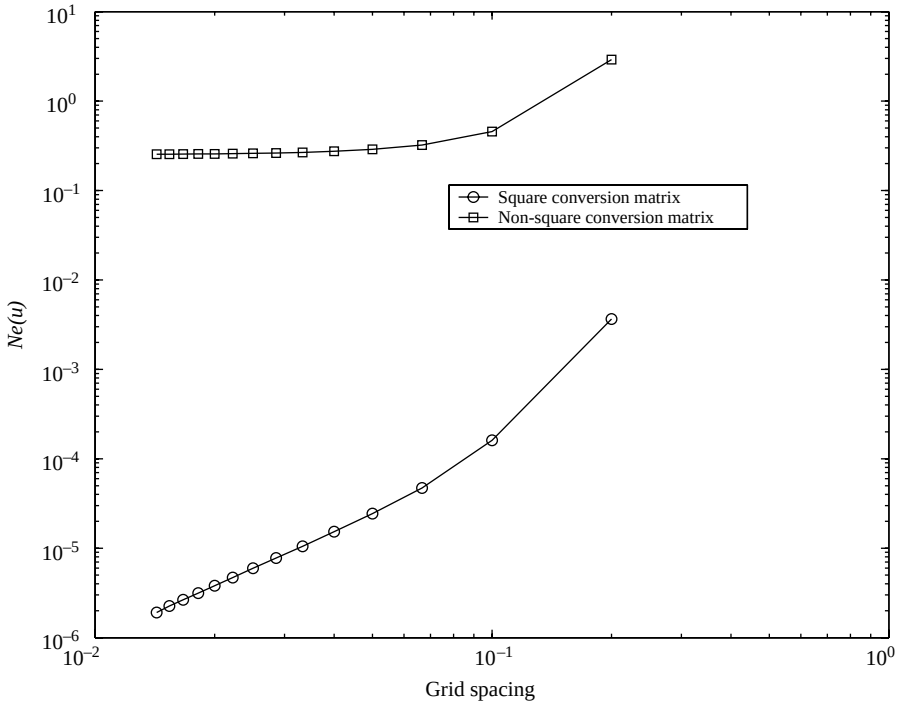

Figure 3.
Biharmonic problem: comparison of the accuracy of the 1D-IRBF method between the two versions

The question here is whether the accuracy of the NSCM approach is improved when one tries to impose the normal derivative boundary conditions through the conversion process. In this case, the dimension of conversion matrices is $(N_x + 2) \times (N_x + 4)$. Table II reveals that this implementation leads to a significant improvement in accuracy over the traditional node-reduction technique.

Numerical results show that both SCM and NSCM versions of the proposed 1D-IRBF method produce very accurate results. The former requires the inversion of underdetermined matrices, while the latter involves the inversion of square matrices. Fortunately, these calculations are conducted in 1D-domains only, and hence they do not add greatly to the computational cost. The classical DRBFN method does not require such inversions, but its approximations involve all data points. It should be emphasized the final matrix obtained by the proposed method has the same dimension as that yielded by the DRBFN method.

For the SCM version, to impose the governing equation on the boundaries, the values of relevant derivatives at the boundary points should be known or should be written in terms of nodal variable values. It is difficult to meet this requirement when extending the SCM version to the case of irregular domains. One possible way to overcome these difficulties is to apply a technique by Sanmiguel-Rojas *et al.* (2005) to generate a non-uniform Cartesian grid in which all the boundary nodes are regular nodes of the grid. For the NSCM version, one does not need to concern these issues when extending it to irregular domains.

4. Natural convection flow in a square slot

The proposed method is applied here to simulate the thermally-driven cavity flow in a square slot. For this problem, the governing equation presents the coupling of momentum (fourth-order PDE, streamfunction formulation) and energy (second-order PDE) equations. Very thin boundary layers are formed at high values of the Rayleigh

Grid	NSCM1	NSCM2
6 × 6	2.9 (0)	9.6 (−4)
11 × 11	4.5 (−1)	6.3 (−5)
16 × 16	3.2 (−1)	2.5 (−5)
21 × 21	2.8 (−1)	1.7 (−5)
26 × 26	2.7 (−1)	1.2 (−5)
31 × 31	2.6 (−1)	9.3 (−6)
36 × 36	2.6 (−1)	7.0 (−6)
41 × 41	2.6 (−1)	5.5 (−6)
46 × 46	2.5 (−1)	4.4 (−6)
51 × 51	2.5 (−1)	3.6 (−6)
56 × 56	2.5 (−1)	3.0 (−6)
61 × 61	2.5 (−1)	2.5 (−6)
66 × 66	2.5 (−1)	2.2 (−6)
71 × 71	2.5 (−1)	1.8 (−6)

Table II.
Fourth-order PDE: relative L_2 errors obtained by the NSCM1 and NSCM2 approaches

Notes: For NSCM1, the multiple boundary conditions are implemented by the node-reduction technique (collocating the governing equation at a smaller number of ip), while for NSCM1, they are imposed by means of integration constants. The latter outperforms the former regarding accuracy and convergence rate. $a\ (-b)$ means $a \times 10^{-b}$

number, thereby making the numerical simulation difficult. This problem provides a good means for testing and validating new numerical methods. From the literature, a range of the Rayleigh number from 10^3 to 10^6 is usually employed for the verification of numerical methods. It will be shown that converged solutions for higher Rayleigh numbers are achieved with the present method, and the obtained RBF results are in very good agreement with the spectral ones. The non-dimensional equations governing the temperature T and streamfunction ψ behaviour can be written as:

$$\frac{\partial T}{\partial t} + \left(\frac{\partial \psi}{\partial y}\frac{\partial T}{\partial x} - \frac{\partial \psi}{\partial x}\frac{\partial T}{\partial y}\right) = \frac{1}{\sqrt{\mathrm{Ra}\,\mathrm{Pr}}}\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right), \tag{66}$$

$$\begin{aligned} &-\frac{\partial}{\partial t}\left(\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2}\right) + \frac{\partial \psi}{\partial y}\left[\frac{\partial^3 \psi}{\partial x^3} + \frac{\partial^2}{\partial y^2}\left(\frac{\partial \psi}{\partial x}\right)\right] - \frac{\partial \psi}{\partial x}\left[\frac{\partial^2}{\partial x^2}\left(\frac{\partial \psi}{\partial y}\right) + \frac{\partial^3 \psi}{\partial y^3}\right] \\ &= -\sqrt{\frac{\mathrm{Pr}}{\mathrm{Ra}}}\left[\frac{\partial^4 \psi}{\partial x^4} + \frac{\partial^2}{\partial x^2}\left(\frac{\partial^2 \psi}{\partial y^2}\right) + \frac{\partial^2}{\partial y^2}\left(\frac{\partial^2 \psi}{\partial x^2}\right) + \frac{\partial^4 \psi}{\partial y^4}\right] + \frac{\partial T}{\partial x}, \end{aligned} \tag{67}$$

where Ra and Pr are the Rayleigh number and the Prandtl number, respectively. The variables are normalized here using reference quantities recommended by Ostrach (1988) for the case of $\sqrt{\mathrm{Ra}/\mathrm{Pr}} > 1$ and $\mathrm{Pr} < 1$. These non-dimensional equations are written in detail in order to show how their derivative terms are computed, e.g. mixed third-order partial derivatives are determined through relevant second- and first-order derivatives. A square cavity of a unit size, containing a fluid of $\mathrm{Pr} = 0.71$, is considered. Non-slip boundary conditions ($\psi = 0$ and $\partial\psi/\partial n$) are imposed along all the walls. The thermal boundary conditions are $T = 0.5$ and $T = -0.5$ (isothermal) along the left and right walls, respectively, and $\partial T/\partial y = 0$ (adiabatic) along the bottom and top walls. The benchmark solutions for this problem can be found in de Vahl Davis (1983) for $10^3 \leq \mathrm{Ra} \leq 10^6$ and in Le Quere (1991) for $\mathrm{Ra} \geq 10^6$. The former used second-order, finite-central-difference approximations and a Richardson extrapolation scheme, while the latter employed a pseudo-spectral Chebyshev algorithm using the spatial resolution up to a 128 × 128 polynomial expansion.

The present solution procedure involves the following steps:

(1) Guess a set of initial conditions: T, ψ and their spatial derivatives.

(2) Discretize in time using a first-order accuracy finite-difference scheme, where the diffusive and convective terms are treated implicitly and explicitly, respectively.

(3) Discretize in space using 1D-IRBF schemes, solve the:

- energy equation (66) for T; and
- momentum equation (67) for ψ.

(4) Check to see whether the solution has reached a steady state:

$$\frac{\sqrt{\sum_{i=1}^{N}\left(\psi_i^{k+1} - \psi_i^{k}\right)^2}}{\sqrt{\sum_{i=1}^{N}\left(\psi_i^{k+1}\right)^2}} < \varepsilon, \tag{68}$$

where k is the time level, ε is the tolerance and N is the number of collocation points.

(5) If it is not satisfied, advance time step and repeat from step 2. Otherwise, stop the computation and output the results.

Normal derivative boundary conditions $\partial\psi/\partial n$ and $\partial T/\partial y$ are imposed by means of integration constants. The energy equation (second-order PDE) is collocated not only at the ip but also at the boundary points on the bottom and top walls.

A range of $\mathrm{Ra} = \{10^5, 10^6, 10^7\}$ is considered here. The computed solution at the lower and nearest value of Ra is taken to be the initial solution. Seven uniform grids, namely $21 \times 21, 31 \times 31, \ldots, 81 \times 81$, are employed to study the convergence behaviour of the method. Time steps are chosen to be 0.1 for $\mathrm{Ra} = 10^5$, 0.05 for $\mathrm{Ra} = 10^6$ and 0.01 for $\mathrm{Ra} = 10^7$.

The following quantities are of interest to this type of flow, the:

- maximum horizontal velocity, $u_{\max}$ on the vertical mid-plane of the cavity and its location;
- maximum vertical velocity, $v_{\max}$ on the horizontal mid-plane of the cavity and its location; and
- average Nusselt number throughout the cavity:

$$\bar{N}_u = \int_0^1 N_u(x)\mathrm{d}x \tag{69}$$

$$N_u(x) = \int_0^1 \left(uT - \frac{\partial T}{\partial x}\right)\mathrm{d}y. \tag{70}$$

The present velocity components are related to the corresponding benchmark solutions (de Vahl Davis, 1983) according to the relation:

$$\sqrt{\mathrm{Ra}\,\mathrm{Pr}}\,(u, v)_{\text{present}} = (u, v)_{\text{benchmark}}.$$

The results obtained are given in Tables III-V, which show the rapid improvement in accuracy as the density increases. Although very sharp gradients are formed at high values of the Rayleigh number, the present method achieves very accurate results. For example, the maximum error is less than 0.2 per cent for $\mathrm{Ra} = 10^7$ using a grid of 81×81.

Table III.
Natural convection, $\mathrm{Ra} = 10^5$

Grid	$u_{\max}$ (error percentage)	y	$v_{\max}$ (error percentage)	x	$\bar{N}_u$ (error percentage)
21×21	34.96 (0.66)	0.855	68.73 (0.20)	0.066	4.492 (0.60)
31×31	34.87 (0.40)	0.855	68.76 (0.25)	0.066	4.512 (0.16)
41×41	34.80 (0.20)	0.855	68.69 (0.15)	0.066	4.516 (0.07)
Benchmark (de Vahl Davis, 1983)	34.73	0.855	68.59	0.066	4.519

Grid	u_{max} (error percentage)	y	v_{max} (error percentage)	x	$\bar{N}_u$ (error percentage)
51 × 51	65.04 (0.32)	0.850	220.96 (0.16)	0.038	8.816 (0.10)
61 × 61	64.98 (0.23)	0.850	220.83 (0.10)	0.038	8.819 (0.07)
71 × 71	64.94 (0.17)	0.850	220.74 (0.06)	0.038	8.820 (0.06)
81 × 81	64.91 (0.12)	0.850	220.69 (0.04)	0.038	8.821 (0.05)
Benchmark (de Vahl Davis, 1983)	64.63	0.850	219.36	0.038	8.800
Benchmark (Le Quere, 1991)	64.83	0.850	220.6	0.038	8.825

Note: It is noted that the percentage errors presented here are relative to the spectral results of Le Quere (1991)

Table IV.
Natural convection, $Ra = 10^6$

Grid	u_{max} (error percentage)	y	v_{max} (error percentage)	x	$\bar{N}_u$ (error percentage)
51 × 51	143.9 (3.16)	0.885	680.8 (2.63)	0.021	16.126 (2.40)
61 × 61	146.9 (1.14)	0.881	693.1 (0.87)	0.021	16.370 (0.93)
71 × 71	148.2 (0.27)	0.879	698.0 (0.17)	0.021	16.463 (0.36)
81 × 81	148.8 (0.13)	0.878	699.7 (0.07)	0.021	16.499 (0.15)
Benchmark (Le Quere, 1991)	148.6	0.879	699.2	0.021	16.523

Table V.
Natural convection, $Ra = 10^7$

Figure 4 shows the streamlines, isotherms and iso-vorticity lines of the flow at $Ra = 10^7$ using a grid of 81 × 81. Every plot contains 17 contour lines whose values vary linearly from the minimum to maximum values, and they look reasonable when compared to the benchmark solutions.

5. Concluding remarks

This paper reports a collocation method based on a 1D-IRBF interpolation scheme for the solution of second- and fourth-order PDEs in rectangular domains. The use of integrated RBFs allows the:

- normal derivative boundary condition to be incorporated more efficiently; and
- exact satisfaction of PDEs not-only at the ip, but also at the boundary points.

The imposition of normal derivative boundary conditions through the conversion process is recommended for use. On the other hand, the employment of a 1D interpolation scheme instead of 2D interpolation scheme permits relatively large numbers of data points to be employed. The present 1D global IRBF scheme is demonstrated on a number of test cases, including the benchmark natural convection problem; a very high degree of accuracy is achieved using relatively coarse grids. Extension of the proposed method to the case of non-rectangular domains is currently underway, and it will be reported in future work.

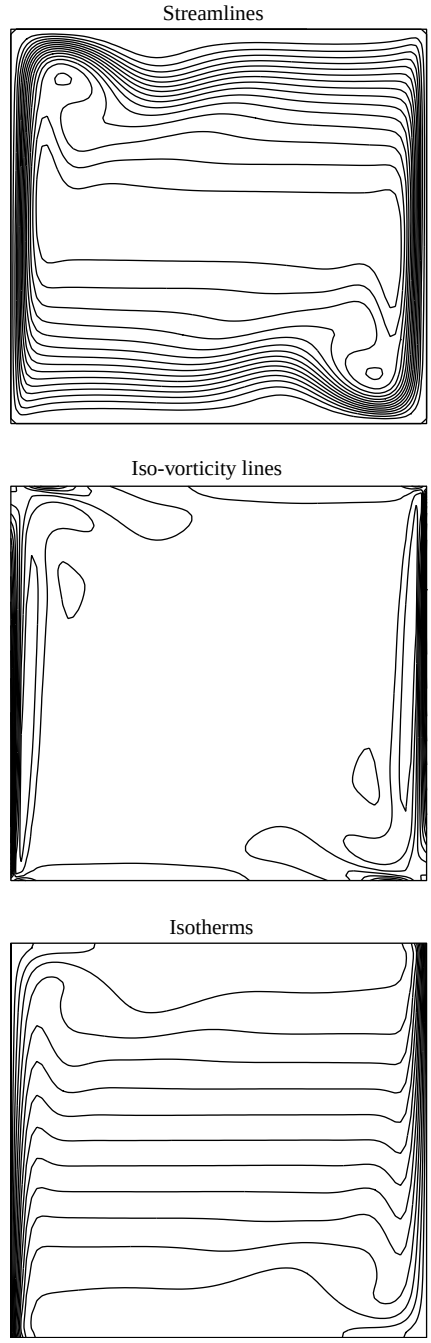

Figure 4.
Natural convection: flow at Ra $= 10^7$ using 81×81

References

Chen, W., Shu, C., He, W. and Zhong, T. (2000), "The application of special matrix product to differential quadrature solution of geometrically nonlinear bending of orthotropic rectangular plates", *Computers & Structures*, Vol. 74, pp. 65-76.

Cheng, A.H-D., Golberg, M.A., Kansa, E.J. and Zammito, G. (2003), "Exponential convergence and H-c multiquadric collocation method for partial differential equations", *Numerical Methods for Partial Differential Equations*, Vol. 19, pp. 571-94.

de Vahl Davis, G. (1983), "Natural convection of air in a square cavity: a bench mark numerical solution", *International Journal for Numerical Methods in Fluids*, Vol. 3, pp. 249-64.

Dubal, M.R. (1994), "Domain decomposition and local refinement for multiquadric approximations I: second-order equations in one-dimension", *Journal of Applied Science and Computation*, Vol. 1 No. 1, pp. 146-71.

Fasshauer, G.E. (1997), "Solving partial differential equations by collocation with radial basis functions", in LeMehaute, A., Rabut, C. and Schumaker, L.L. (Eds), *Surface Fitting and Multiresolution Methods*, Vanderbilt University Press, Nashville, TN, pp. 131-8.

Fedoseyev, A.I., Friedman, M.J. and Kansa, E.J. (2002), "Improved multiquadric method for elliptic partial differential equations via PDE collocation on the boundary", *Computers & Mathematics with Applications*, Vol. 43 Nos 3/5, pp. 439-55.

Haykin, S. (1999), *Neural Networks: A Comprehensive Foundation*, Prentice-Hall, Englewood Cliffs, NJ.

Kansa, E.J. (1990), "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*, Vol. 19 Nos 8/9, pp. 147-61.

Kansa, E.J. and Hon, Y.C. (2000), "Circumventing the ill-conditioning problem with multiquadric radial basis functions: applications to elliptic partial differential equations", *Computers and Mathematics with Applications*, Vol. 39, pp. 123-37.

Larsson, E. and Fornberg, B. (2003), "A numerical study of some radial basis function based solution methods for elliptic PDEs", *Computers and Mathematics with Applications*, Vol. 46, pp. 891-902.

Lee, C.K., Liu, X. and Fan, S.C. (2003), "Local multiquadric approximation for solving boundary value problems", *Computational Mechanics*, Vol. 30 Nos 5/6, pp. 396-409.

Le Quere, P. (1991), "Accurate solutions to the square thermally driven cavity at high Rayleigh number", *Computers & Fluids*, Vol. 20 No. 1, pp. 29-41.

Li, J. and Hon, Y.C. (2004), "Domain decomposition for radial basis meshless methods", *Numerical Methods for Partial Differential Equations*, Vol. 20, pp. 450-62.

Madych, W.R. and Nelson, S.A. (1988), "Multivariate interpolation and conditionally positive definite functions", *Approximation Theory and its Applications*, Vol. 4, pp. 77-89.

Madych, W.R. and Nelson, S.A. (1990), "Multivariate interpolation and conditionally positive definite functions, II", *Mathematics of Computation*, Vol. 54 No. 189, pp. 211-30.

Mai-Duy, N. (2005), "Solving high order ordinary differential equations with radial basis function networks", *International Journal for Numerical Methods in Engineering*, Vol. 62, pp. 824-52.

Mai-Duy, N. and Tanner, R.I. (2005), "Solving high order partial differential equations with radial basis function networks", *International Journal for Numerical Methods in Engineering*, Vol. 63, pp. 1636-54.

Mai-Duy, N. and Tran-Cong, T. (2001), "Numerical solution of differential equations using multiquadric radial basis function networks", *Neural Networks*, Vol. 14 No. 2, pp. 185-99.

Mai-Duy, N. and Tran-Cong, T. (2003), "Approximation of function and its derivatives using radial basis function networks", *Applied Mathematical Modelling*, Vol. 27, pp. 197-220.

Mai-Duy, N. and Tran-Cong, T. (2005), "An efficient indirect RBFN-based method for numerical solution of PDEs", *Numerical Methods for Partial Differential Equations*, Vol. 21, pp. 770-90.

Ostrach, S. (1988), "Natural convection in enclosures", *Journal of Heat Transfer*, Vol. 110, pp. 1175-90.

Park, J. and Sandberg, I.W. (1991), "Universal approximation using radial basis function networks", *Neural Computation*, Vol. 3, pp. 246-57.

Power, H. and Barraco, V. (2002), "A comparison analysis between unsymmetric and symmetric radial basis function collocation methods for the numerical solution of partial differential equations", *Computers and Mathematics with Applications*, Vol. 43, pp. 551-83.

Sanmiguel-Rojas, E., Ortega-Casanova, J., del Pino, C. and Fernandez-Feria, R. (2005), "A Cartesian grid finite-difference method for 2D incompressible viscous flows in irregular geometries", *Journal of Computational Physics*, Vol. 204, pp. 302-18.

Sarler, B., Perko, J. and Chen, C.S. (2004), "Radial basis function collocation method solution of natural convection in porous media", *International Journal of Numerical Methods for Heat & Fluid Flow*, Vol. 14 No. 2, pp. 187-212.

Shu, C. (2000), *Differential Quadrature and its Application in Engineering*, Springer-Verlag, London.

Shu, C., Chen, W. and Du, H. (2000), "Free vibration analysis of curvilinear quadrilateral plates by the differential quadrature method", *Journal of Computational Physics*, Vol. 163, pp. 452-66.

Shu, C., Ding, H. and Yeo, K.S. (2003), "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*, Vol. 192, pp. 941-54.

Zerroukat, M., Power, H. and Chen, C.S. (1998), "A numerical method for heat transfer problems using collocation and radial basis functions", *International Journal for Numerical Methods in Engineering*, Vol. 42, pp. 1263-78.

Corresponding author
N. Mai-Duy can be contacted at: nam.maiduy@aeromech.usyd.edu.au