Solving High-Order Partial Differential Equations with Indirect Radial Basis Function Networks

N. Mai-Duy* and R.I. Tanner

School of Aerospace, Mechanical and Mechatronic Engineering,

The University of Sydney, NSW 2006, Australia

Submitted to *Int. J. Numer. Meth. Engng*, 21st June 2004; revised 25 November 2004

*Corresponding author: Telephone +61 2 9351 7151, Fax +61 2 9351 7060, E-mail nam.maiduy@aeromech.usyd.edu.au

SUMMARY

This paper reports a new numerical method based on radial basis function networks (RBFNs) for solving high-order partial differential equations (PDEs). The variables and their derivatives in the governing equations are represented by integrated RBFNs. The use of the integration process in constructing neural networks allows the straightforward implementation of multiple boundary conditions and the accurate approximation of high-order derivatives. The proposed RBFN method is verified successfully through the solution of thin-plate bending and viscous flow problems which are governed by biharmonic equations. For thermally driven cavity flows, the solutions are obtained up to a high Rayleigh number of 10^7.

KEY WORDS: radial basis functions; approximation; multiple boundary conditions; high order; derivatives; partial differential equations

1 INTRODUCTION

Neural networks have found application in many disciplines: neurosciences, mathematics, statistics, physics, computer science and engineering [1]. The concept of using radial basis function networks (RBFNs) to solve partial differential equations (PDEs) was introduced by Kansa in 1990 [2]. Since then it has received much attention from the science and engineering communities. The construction of networks can be based on a differentiation process (the direct RBFN approach - DRBFNs) [2] or based on an integration process (the indirect RBFN approach - IRBFNs) [3]. For a typical RBFN-based numerical method, each dependent variable and its derivatives are represented by DRBFNs/IRBFNs. The governing differential equations together with boundary conditions are then discretized by point collocation. Most RBF publications were concerned with the solution of second-order PDEs, e.g., [4-9]. Recently, the IRBFN method has been developed for the solution of high-order ordinary differential equations (ODEs) without prior conversions into the equivalent systems of first-order ODEs [10]. In this paper, the unsymmetric indirect RBF collocation method is extended to solve high-order PDEs directly.

Consider problems governed by multi-harmonic equations, such as thin-plate bending and Stokes flow problems involving biharmonic equations. To solve these problems, new variables are usually introduced in order to transform the multi-harmonic equations into the coupled sets of harmonic equations from which the conventional low-order methods of discretization such as the boundary element methods (BEMs), finite difference methods (FDMs) or finite element methods (FEMs) can be applied for obtaining a numerical solution. However, the drawbacks of this transformation are that it produces large system matrices as well as one usually needs to derive the computational boundary conditions for new variables. All of these difficulties can be overcome by developing high-order numerical methods to solve the governing

multi-harmonic equations directly. In addition, the high-order methods are known to have the capability to achieve accurate results using relatively low numbers of degrees of freedom (DOF) or to allow the savings in computational effort and virtual storage for a given accuracy [11].

In solving high-order PDEs in a direct manner, special attention needs to be paid to the two important issues, namely the implementation of multiple boundary conditions and the approximation of high-order derivatives. To properly implement the multiple boundary conditions, there are a number of techniques available in the literature, e.g., the δ-technique [12], the modified weighting matrix approach [13], the approach of directly substituting the boundary conditions into the discrete governing equations [14], the general approach [15] and the generalized differential quadrature rule (GDQR) technique [16]. The basic ideas of these techniques are to try adding "extra boundary points" to the original set of data points so that each boundary point has only one condition (the δ-technique), reducing the number of data points used for discretizing the governing equations in an appropriate manner (the direct substitution technique) and employing the same number of unknown variables as that of the conditions at a point (the GDQR technique), in order to form a square system matrix. More detailed discussions can be found in [16,17]. In the context of the numerical solution of differential equations, high-order derivatives are undesirable in general because they can introduce large approximation error. The use of higher-order conventional Lagrange polynomials does not guarantee to yield a better quality (smoothness) of approximation [18].

The two important issues mentioned above can be treated effectively here by using integrated RBFNs. In the present unsymmetric indirect RBF collocation approach, the construction of neural networks representing the variable and its derivatives is based on integration. The governing equations and boundary conditions are discretized by means of point collocation. The increase of a number of columns due

to the presence of integration constants is brought into balance with the increase of rows due to the discretization of multiple boundary conditions, and hence it can lead to a square system matrix whatever the order of PDEs. On the other hand, the integration process appears to be suitable for the approximation of high-order derivatives. It can be argued as follows. Due to the lack of theory, it is very difficult to choose RBFN parameters, such as RBF widths, properly. Consequently, some oscillation can be induced between the nodal function values when using the RBFN interpolation scheme. This behaviour has no relation at all to that of the true function. In practice, the oscillation has been often observed in regions near the boundaries. It can be seen that by differentiating RBFNs, the spurious oscillations will be strongly magnified with an increase in order of derivatives (the slope of the curve). However, it is expected that the process of integrating RBFNs (the area under curve) can suppress "noise", thereby resulting in smoother approximating derivatives.

The present unsymmetric IRBFN collocation method will be verified with the solution of thin-plate bending and viscous flow problems that are governed by biharmonic equations. In the simulation of natural convection flows, the resultant matrices become much larger due to the increased number of the field variables and the requirement of dense data densities for the simulations at high Rayleigh numbers. To keep the system matrix size comparable to that associated with the DRBFN approach, the multiple spaces of IRBFN weights for each variable are converted into the single space of nodal variable values.

The remainder of the paper is organized as follows. In section 2, RBFNs and their two direct and indirect approaches for the approximation of a function and its derivatives are reviewed. Section 3 presents IRBFNs for solving high-order PDEs without prior conversions into the equivalent systems of low-order PDEs. In sections 4 and 5, the present IRBFN method is verified through the solution of thin-plate bending

problems and thermally driven cavity flows that are governed by biharmonic equations. Section 6 gives some concluding remarks. Finally, the analytic forms of new basis functions obtained from integrating RBFs are given in the appendix.

2 RADIAL BASIS FUNCTION NETWORKS

RBFNs allow a conversion of a function from low-dimensional space (e.g., 1D-3D) to high-dimensional space in which the function will be expressed as a linear combination of RBFs [1]

$$y(\mathbf{x}) \approx f(\mathbf{x}) = \sum_{i=1}^{m} w^{(i)} g^{(i)}(\mathbf{x}), \tag{1}$$

where y and f are the exact and approximate functions, respectively, superscripts denote the elements of a set of neurons, $\mathbf{x}$ the input vector, m the number of RBFs, $\{w^{(i)}\}_{i=1}^{m}$ the set of network weights to be found, and $\{g^{(i)}(\mathbf{x})\}_{i=1}^{m}$ the set of RBFs.

According to Micchelli's theorem, there is a large class of radial basis functions, e.g., multiquadrics, inverse multiquadrics and Gaussian functions, whose design matrices (interpolation matrices, coefficient matrices) obtained from (1) are always invertible, provided that the data points are distinct. This is all that is required for the non-singularity of the design matrix, whatever the number of data points and the dimension of the problem [1]. On the other hand, the Cover's theorem, which can be stated as follow "the higher the dimension of hidden space, the more accurate the approximation will be" [1], indicates the property of "mesh convergence" of RBFNs. Furthermore, it has been proved that RBFNs are capable of representing any continuous function to a prescribed degree of accuracy (universal approximation theorem) [19]. These important theorems can be seen to provide the theoretical basis for the design of RBFNs to the field of numerical solution of PDEs.

Since multiquadrics (MQ) are ranked as the most accurate among RBFs [20] and possess exponential convergence with the refinement of spatial discretization [21,22], this study will employ these basis functions whose form is

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

where $\mathbf{c}^{(i)}$ and $a^{(i)}$ are the centre and width of the ith MQ basis function, respectively, and superscript T denotes the transpose of a vector. To make the training process simple, the centres and widths of RBFs are chosen in advance. For the former, the set of centres is chosen to be the same as the set of collocation points, while for the latter, the following relation is used

$$a^{(i)} = \beta d^{(i)}, \tag{3}$$

where β is a positive scalar, and $d^{(i)}$ is the minimum of distances from the ith center to its neighbours. Relation (3) allows the RBF width a to be broader in the area of lower data densities and narrower in the area of higher data densities. The network weights are then found by minimizing the sum of squares of the difference between actual and required outputs.

2.1 Direct approach

In the direct RBFN (DRBFN) approach, the RBFN (1) is utilized to represent the original function y, and subsequently, its derivatives are computed by differentiating

(1) as

$$y(\mathbf{x}) \approx f(\mathbf{x}) = \sum_{i=1}^{m} w^{(i)} g^{(i)}(\mathbf{x}), \tag{4}$$

$$\frac{\partial y^k(\mathbf{x})}{\partial x_j^k} \approx \frac{\partial^k f(\mathbf{x})}{\partial x_j^k} = \frac{\partial^k \left(\sum_{i=1}^{m} w^{(i)} g^{(i)}(\mathbf{x})\right)}{\partial x_j^k} = \sum_{i=1}^{m} w^{(i)} h_{[x_j]}^{[k](i)}(\mathbf{x}), \tag{5}$$

where subscripts j denote the scalar components of a vector, and $\left\{h_{[x_j]}^{[k](i)}(\mathbf{x})\right\}_{i=1}^{m} = \left\{\partial^k g^{(i)}(\mathbf{x})/\partial x_j^k\right\}_{i=1}^{m}$ the set of newly derived basis functions in the approximation of the kth-order derivative of a function y with respect to the x_j variable.

2.2 Indirect approach

In the indirect (IRBFN) approach, RBFNs (1) are used to represent the highest-order derivatives of a function y, e.g., $\partial^k y/\partial x_j^k$. Lower-order derivatives and the function itself are then obtained by integrating those RBFNs as

$$\frac{\partial^k y(\mathbf{x})}{\partial x_j^k} \approx \frac{\partial^k f(\mathbf{x})}{\partial x_j^k} = \sum_{i=1}^{m} w_{[x_j]}^{(i)} g^{(i)}(\mathbf{x}), \tag{6}$$

$$\frac{\partial^{k-1} y(\mathbf{x})}{\partial x_j^{k-1}} \approx \frac{\partial^{k-1} f(\mathbf{x})}{\partial x_j^{k-1}} = \sum_{i=1}^{m+q_1} w_{[x_j]}^{(i)} H_{[x_j]}^{[k-1](i)}(\mathbf{x}), \tag{7}$$

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

$$y(\mathbf{x}) \approx f_{[x_j]}(\mathbf{x}) = \sum_{i=1}^{m+q_k} w_{[x_j]}^{(i)} H_{[x_j]}^{[0](i)}(\mathbf{x}), \tag{8}$$

where subscripts $[x_j]$ denote the quantities associated with the process of integration in the x_j direction; $q_1, \cdots, q_k$ the numbers of nodal constants arising from the integration process (integration constants are directly captured here) ($q_2 = 2q_1, \cdots, q_k = kq_1$); and $H_{[x_j]}^{[k-1](i)} = \int g^{(i)} dx_j, H_{[x_j]}^{[k-2](i)} = \int H^{[k-1](i)} dx_j, \cdots, H_{[x_j]}^{[0](i)} = \int H^{[1](i)} dx_j$ ($i = 1, 2, \cdots, m$) newly derived basis functions in the approximation of $(k-1)$th-order derivative,

$(k-2)$th-order derivative, $\cdots$, the original function y, respectively. Note that for convenience of notation, the integration constants (unknowns) and their associated basis functions (known polynomials) are also denoted by the notations $w^{(i)}$ and $H^{[.](i)}(\mathbf{x})$, respectively, but with $i > m$.

The evaluation of (6)-(8) at a set of collocation points $\{\mathbf{x}^{(i)}\}_{i=1}^{n}$ yields the following systems of equations

$$\begin{aligned}
\mathbf{f}_{[x_j]}^{[k]}(\mathbf{x}) &= \mathbf{G}(\mathbf{x})\mathbf{w}_{[x_j]}, && (9)\\
\mathbf{f}_{[x_j]}^{[k-1]}(\mathbf{x}) &= \mathbf{H}_{[x_j]}^{[k-1]}(\mathbf{x})\mathbf{w}_{[x_j]}, && (10)\\
\cdots \quad &\cdots \quad \cdots\cdots\cdots\cdots && \\
\mathbf{f}_{[x_j]}(\mathbf{x}) &= \mathbf{H}_{[x_j]}^{[0]}(\mathbf{x})\mathbf{w}_{[x_j]}, && (11)
\end{aligned}$$

where $\mathbf{G}_{[x_j]}, \mathbf{H}_{[x_j]}^{[k-1]}, \cdots, \mathbf{H}_{[x_j]}^{[0]}$ are the network-design-matrices associated with the approximation of the kth-order derivative, $(k-1)$th-order derivative, $\cdots$, the original function with respect to the x_j variable, respectively; $\mathbf{w}_{[x_j]} = \{w_{[x_j]}^{(i)}\}_{i=1}^{m+q_k}$ the set of network weights in the x_j direction to be found; $\mathbf{f}_{[x_j]} = \{f_{[x_j]}(\mathbf{x}^{(i)})\}_{i=1}^{n}$; $\mathbf{f}_{[x_j]}^{[1]} = \{\frac{\partial f(\mathbf{x}^{(i)})}{\partial x_j}\}_{i=1}^{n}$; and $\mathbf{f}_{[x_j]}^{[k]} = \{\frac{\partial^k f(\mathbf{x}^{(i)})}{\partial x_j^k}\}_{i=1}^{n}$. For the purpose of computation, the matrices $\mathbf{G}_{[x_j]}, \mathbf{H}_{[x_j]}^{[k-1]}, \cdots, \mathbf{H}_{[x_j]}^{[1]}$ will be augmented using zero-submatrices so that they have the same size as the matrix $\mathbf{H}_{[x_j]}^{[0]}$. In the approximation of a function and its derivatives, the IRBFN weight vectors $\mathbf{w}_{[x_j]}$ with $j = \{1, 2, \cdots\}$ can be determined by making use of the functional networks (11).

For convenience of presentation, the following discussions are restricted to biharmonic boundary value problems. However, they can be extended straightforwardly to multi-harmonic ones, and furthermore, all of the advantageous features found in the case of biharmonic problems can be preserved. The cross derivative $\partial f^4(\mathbf{x})/\partial x_i^2 \partial x_j^2$ can be computed by using the relevant network-design-matrices associated with the second-order derivatives. Although the order of differentiation makes no difference

theoretically, due to numerical error, it will be better to take the average of the two equivalent representations

$$\frac{\partial^4 f}{\partial x_i^2 \partial x_j^2} = \frac{1}{2}\left[\frac{\partial^2}{\partial x_i^2}\left(\frac{\partial^2 f}{\partial x_j^2}\right) + \frac{\partial^2}{\partial x_j^2}\left(\frac{\partial^2 f}{\partial x_i^2}\right)\right], \tag{12}$$

or in the matrix form

$$\mathbf{f}_{,\mathbf{iijj}} = \frac{1}{2}\left[\left(\mathbf{H}^{[2]}_{[x_i]}(\mathbf{H}^{[0]}_{[x_i]})^{-1}\right)\mathbf{H}^{[2]}_{[x_j]}\mathbf{w}_{[x_j]} + \left(\mathbf{H}^{[2]}_{[x_j]}(\mathbf{H}^{[0]}_{[x_j]})^{-1}\right)\mathbf{H}^{[2]}_{[x_i]}\mathbf{w}_{[x_i]}\right]. \tag{13}$$

Expressions of f and its derivatives at point $\mathbf{x}_0$ in terms of network weights $\mathbf{w}_{[x_j]} = \{w^{(i)}_{[x_j]}\}_{i=1}^{m+q_4}$ can be given by

$$f(\mathbf{x}_0) = \frac{1}{N}\sum_{j=1}^{N}\left(\left[H^{[0](1)}_{[x_j]}(\mathbf{x}_0), \cdots, H^{[0](m+1)}_{[x_j]}(\mathbf{x}_0), \cdots, H^{[0](m+q_1+1)}_{[x_j]}(\mathbf{x}_0), \cdots\right]\mathbf{w}_{[x_j]}\right), \tag{14}$$

$$\frac{\partial f^2(\mathbf{x}_0)}{\partial x_j^2} = \left[H^{[2](1)}_{[x_j]}(\mathbf{x}_0), \cdots, H^{[2](m+1)}_{[x_j]}(\mathbf{x}_0), \cdots, H^{[2](m+q_1+1)}_{[x_j]}(\mathbf{x}_0), \cdots\right]\mathbf{w}_{[x_j]}, \tag{15}$$

$$\frac{\partial^4 f(\mathbf{x}_0)}{\partial x_j^4} = \left[g^{(1)}_{[x_j]}(\mathbf{x}_0), \cdots, 0, \cdots, 0, \cdots\right]\mathbf{w}_{[x_j]}, \tag{16}$$

$$\begin{aligned}\frac{\partial^4 f(\mathbf{x}_0)}{\partial x_i^2 \partial x_j^2} = \frac{1}{2}(&\left[H^{[2](1)}_{[x_i]}(\mathbf{x}_0), \cdots, H^{[2](m+1)}_{[x_i]}(\mathbf{x}_0), \cdots\right](\mathbf{H}^{[0]}_{[x_i]})^{-1}\left(\mathbf{H}^{[2]}_{[x_j]}\mathbf{w}_{[x_j]}\right)\\ &+ \left[H^{[2](1)}_{[x_j]}(\mathbf{x}_0), \cdots, H^{[2](m+1)}_{[x_j]}(\mathbf{x}_0), \cdots\right](\mathbf{H}^{[0]}_{[x_j]})^{-1}\left(\mathbf{H}^{[2]}_{[x_i]}\mathbf{w}_{[x_i]}\right)),\end{aligned} \tag{17}$$

where N is the dimension of the problem. In (14), due to numerical errors, the approximate function f at $\mathbf{x}_0$ is taken to be the average of the $f_{[x_j]}(\mathbf{x}_0)$'s.

3 IRBFNs for solving high-order PDEs

Indirect RBFNs are employed to represent the solution of high-order PDEs via a point collocation mechanism. The governing equations are solved without splitting them into the equivalent systems of low-order PDEs. New constants arising from the integration process are used for the treatment of multiple boundary conditions.

In constructing system/interpolation matrices, the increase of rows due to the discretization of multiple boundary conditions is brought into balance with the increase of columns due to the presence of integration constants, and hence it can lead to a square system of equations whatever the order of DEs.

The detailed implementation of IRBFNs for the solution of high-order ODEs was reported previously in [10]. In present work that deals with high-order PDEs, two versions of indirect RBFNs are employed. The first version uses the IRBFN formulation directly in terms of network weights, while in the second version, the formulation is expressed in terms of nodal variable values. The following is concerned with biharmonic equations governing thin-plate problems and viscous flows. However, the formulations can be extended to solve multi-harmonic equations in a straightforward manner.

4 THIN-PLATE PROBLEMS

4.1 Governing equations

In this section, the deflection and free vibration of thin rectangular plates are considered. The governing equations can be written as

$$\nabla^4 v = F(\mathbf{x}), \qquad \text{(deflection)} \tag{18}$$

$$\nabla^4 v = \Omega^2 v, \qquad \text{(free vibration)} \tag{19}$$

and the boundary conditions are given by

$$v = 0, \ \frac{\partial v}{\partial n} = 0 \quad \text{for a clamped edge,} \tag{20}$$

$$v = 0, \ \frac{\partial^2 v}{\partial n^2} = 0 \quad \text{for a simply supported edge,} \tag{21}$$

where $\mathbf{x}$ is the position vector of a point in the domain of interest, v the deflection/the mode shape function, Ω the frequency, F the forcing term, and n the direction normal to the edge.

4.2 IRBFN formulation

The rectangular domain of interest is discretized as shown in Figure 1. It is different from the 1D-IRBFN analysis that there exist two RBF networks representing the variable v here. These functional networks need to be enforced to be identical, resulting in a constraint equation $v_{[x_1]}(\mathbf{x}) = v_{[x_2]}(\mathbf{x})$. The IRBFN formulation can be written as

$$SSE_1 + SSE_2 + SSE_3 \to 0, \tag{22}$$

where SSE_1, SSE_2 and SSE_3 are the sums of squared errors which are employed to ensure that IRBFNs satisfy the governing equation, the identical representations of the original function (i.e., $v_{[x_1]}(\mathbf{x}) = v_{[x_2]}(\mathbf{x})$), and the boundary conditions, respectively. Let nip and nbp_j denote the number of interior points and the number of boundary points on the plate edge normal to the x_j direction, respectively. It can be seen that the direct matrix obtained from (22) will have the number of rows as $[(nip + 2nbp_1 + 2nbp_2) + (nip) + (4nbp_1 + 4nbp_2)]$ that corresponds to SSE_1, SSE_2 and SSE_3, respectively, and the number of columns as $[(nip + 2nbp_1 + 4nbp_1) + (nip + 2nbp_2 + 4nbp_2)]$ that corresponds to the network weights in the x_1- and x_2- directions, respectively. Hence, the formulation (22) directly results in a square system matrix.

4.3 Numerical examples

A number of examples are presented in this section to demonstrate the effectiveness of the present method. In all following test cases, the width of the ith RBF ($a^{(i)}$) is simply chosen to be the minimum distance from the ith centre to neighbouring centres ($\beta = 1$). The accuracy of numerical solution produced by an approximation scheme can be measured via the norm of relative errors of the solution as follows

$$N_e = \sqrt{\frac{\sum_{i=1}^{n} \left[v_e(\mathbf{x}^{(i)}) - v(\mathbf{x}^{(i)})\right]^2}{\sum_{i=1}^{n} v_e(\mathbf{x}^{(i)})^2}}, \tag{23}$$

where n is the number of collocation points, $\mathbf{x}^{(i)}$ the ith collocation point, v and v_e the calculated and exact solutions, respectively. Another important measure is the convergence rate of the solution with the refinement of spatial discretization

$$N_e(h) \approx \gamma h^{\alpha} = O(h^{\alpha}) \tag{24}$$

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

4.3.1 Benchmark test problem

Consider a typical benchmark test problem [23] governed by the homogeneous biharmonic equation (i.e., $F(x_1, x_2) = 0$). The exact solution is given by

$$v = \frac{1}{2} x_1 \left(\sin x_1 \cosh x_2 - \cos x_1 \sinh x_2 \right). \tag{25}$$

The rectangular domain of interest is taken to be $[-2, 2] \times [-2, 2]$. Eight uniform densities, namely $5 \times 5, 7 \times 7, \cdots, 19 \times 19$ data points, are employed to study convergence. Two different types of boundary conditions, namely $\{v; \partial v/\partial n\}$ and $\{v; \partial^2 v/\partial n^2\}$, are considered (Figure 1). All boundary data are nonzero.

Boundary conditions given in terms of v and $\partial v/\partial n$

Figure 2 shows that the present method yields high accuracy and high rates of convergence. The obtained solutions converge apparently as $O(h^{5.3})$ and $O(h^{4.6})$ for v and its Laplacian u ($u = \nabla^2 v$), respectively, where h is the centre spacing. At the highest density of 19×19 data points, the error norms are 5.8×10^{-7} for v and 4.8×10^{-5} for u.

Boundary conditions given in terms of v and $\partial^2 v/\partial n^2$

Similarly, accurate results and high rates of convergence are obtained for this type of boundary conditions (Figure 3). The convergence rates are of $O(h^{5.0})$ and $O(h^{4.3})$ for v and u, respectively, where h is the centre spacing. At the finest density of 19×19, the norms of relative errors are 2.5×10^{-5} for v and 4.7×10^{-5} for u.

4.3.2 Thin-plate bending problems

Simply supported thin rectangular plates under sinusoidal load

A function $F(x_1, x_2)$ on the right hand side of (18) is given by

$$F(x_1, x_2) = \frac{q_0}{D} \sin \frac{\pi x_1}{a} \sin \frac{\pi x_2}{b}, \tag{26}$$

where q_0 represents the intensity of the load at the centre of a plate, a and b the lengths of the plate edges, and $D = Eh^3/12(1-\nu^2)$ the flexural rigidity in which E the modulus of elasticity, ν Poisson's ratio, and h the thickness of a plate. The bending moments in the x_1- and x_2- directions are given by

$$M_x = -D\left(\frac{\partial^2 v}{\partial x_1^2} + \nu \frac{\partial^2 v}{\partial x_2^2}\right), \tag{27}$$

$$M_y = -D\left(\frac{\partial^2 v}{\partial x_2^2} + \nu \frac{\partial^2 v}{\partial x_1^2}\right). \tag{28}$$

Analytical solutions can be found in [24]. The following parameters are used

$$a = b = 200\text{cm},\ h = 10\text{cm},\ \nu = 0.3,\ E = 2.1 \times 10^6 \text{kg/cm}^2 \text{ and } q_0 = 0.5\text{kg/cm}^2.$$

Three uniform discretizations of 4×4, 6×6 and 8×8 are employed. With relatively low numbers of DOF, very accurate results are obtained. For example, at the highest density of 8×8, error-norms are 1.2×10^{-6} and 5.8×10^{-5} for the deflection and bending moments, respectively. Convergence with spatial refinement is very fast, e.g., up to $O(h^{10.1})$ for v and $O(h^{9.3})$ for M_x and M_y.

Clamped thin rectangular plates under a uniform load

Consider a plate of dimension $[-a/2, a/2] \times [-b/2, b/2]$ with built-in edges. The

results of deflection at the centre and bending moment at $(x = a/2, y = 0)$ for the ratio b/a varying from 1.0 to 2.0 with an increment of 0.1 are shown in Table 1 together with the analytical results. With only coarse data density of 11×11, the computed results agree well with the analytical results [24].

4.3.3 Plate vibration problems

To further verify the present unsymmetric IRBFN collocation method, the free vibration of simply supported plates is considered here. The IRBFN discretization of (19) results in the following system of algebraic equations for the unknown vector of network weights

$$\mathbf{A}\mathbf{w} = \Omega^2 \mathbf{B}\mathbf{w}, \tag{29}$$

$$\mathbf{C}\mathbf{w} = \mathbf{0}, \tag{30}$$

where $\mathbf{A}$ and $\mathbf{B}$ are the matrices obtained from discretizing the governing equations, and $\mathbf{C}$ the matrix obtained from two sources, namely the discretization of boundary conditions and the condition of unity for two functional networks associated with the x_1- and x_2- directions ($v_{[x_1]}(\mathbf{x}) = v_{[x_2]}(\mathbf{x})$). To obtain natural frequencies Ω_i, the system (29)-(30) needs to be modified as follows

$$[\mathbf{A_1}, \mathbf{A_2}]\{\mathbf{w_1}; \mathbf{w_2}\} = \Omega^2 [\mathbf{B_1}, \mathbf{B_2}]\{\mathbf{w_1}; \mathbf{w_2}\}, \tag{31}$$

$$[\mathbf{C_1}, \mathbf{C_2}]\{\mathbf{w_1}; \mathbf{w_2}\} = \mathbf{0}, \tag{32}$$

By solving (32), the subset $\mathbf{w_2}$ will be expressed as

$$\mathbf{w_2} = -\mathbf{C_2}^{-1}\mathbf{C_1}\mathbf{w_1} \tag{33}$$

Substitution of (33) into (31) yields

$$\left(\mathbf{A_1} - \mathbf{A_2}\mathbf{C_2}^{-1}\mathbf{C_1}\right)\mathbf{w_1} = \Omega^2\left(\mathbf{B_1} - \mathbf{B_2}\mathbf{C_2}^{-1}\mathbf{C_1}\right)\mathbf{w_1} \tag{34}$$

from which the natural frequencies can be found by using the QZ algorithm. The subset $\mathbf{w_2}$ contains the network weights associated with the interior points and the integration constants.

Consider a simply supported square plate of a unit size. Three discretizations of 7×7, 9×9 and 11×11 data points are employed. The results of the natural frequency for the first five modes are tabulated in Table 2. The solutions obtained by the differential quadrature method [17] and by the Rayleigh-Ritz method [25] are also included for comparison. All results are in good agreement. It is found that using a low density of 7×7 is able to yield accurate results, where the maximum error relative to the the Rayleigh-Ritz solutions is 0.4%.

5 NATURAL CONVECTION FLOWS

Heat transfer by natural convection in an enclosed cavity has received a great deal of attention in recent years due to its wide applications in engineering. This problem is known to provide a good means of testing and validating numerical methods.

5.1 Governing equations

Consider the two-dimensional, steady-state, laminar, buoyancy-induced flow of an incompressible fluid of density ρ and viscosity μ. With the employment of Boussinesq approximation, i.e., the fluid is assumed to have constant properties except for the

generation of buoyant force, the dimensionless governing equations in terms of a stream function ϕ and a temperature T can be written as

$$\nabla^4\phi - \frac{\partial T}{\partial x_1} = \frac{Ra}{Pr}\left[\frac{\partial\phi}{\partial x_2}\left(\frac{\partial^3\phi}{\partial x_1^3} + \frac{\partial^3\phi}{\partial x_1\partial x_2^2}\right) - \frac{\partial\phi}{\partial x_1}\left(\frac{\partial^3\phi}{\partial x_1^2\partial x_2} + \frac{\partial^3\phi}{\partial x_2^3}\right)\right], \quad (35)$$

$$\nabla^2 T = Ra\left(\frac{\partial\phi}{\partial x_2}\frac{\partial T}{\partial x_1} - \frac{\partial\phi}{\partial x_1}\frac{\partial T}{\partial x_2}\right). \quad (36)$$

The independent dimensionless parameters appearing in the equations (35)-(36) are the Rayleigh number (Ra) and the Prandtl number (Pr). More details can be found in [26].

The domain of interest here is a square cavity of a unit size. Non-slip boundary conditions ($\phi = 0, \partial\phi/\partial n = 0$) are applied along all the walls. The left and right vertical walls are kept at temperatures 1 and 0, respectively, while the horizontal walls are insulated ($\partial T/\partial n = 0$). The thermally driven cavity flow contains no singularities which makes it more realistic than the lid driven cavity flow problem [27]. A Boussinesq fluid of the Prandtl number of 0.71 is considered. The benchmark solutions for this problem can be found in [28] for $10^3 \leq Ra \leq 10^6$ and in [29] for $Ra \geq 10^6$. The former used second-order, finite central difference approximations and a Richardson extrapolation scheme, while the latter employed a pseudo-spectral Chebyshev algorithm with increasing the spatial resolution up to a 128×128 polynomial expansion.

5.2 IRBFN formulation

In the simulation of natural convection flows, the obtained system matrices are much larger due to the increased number of the field variables and the requirement of dense data densities for the simulations at high Rayleigh numbers. To keep the system matrix size comparable to that associated with the DRBFN approach, the multiple

spaces of IRBFN weights for each variable (ϕ and T) need to be converted into the single space of nodal variable values. All given derivatives on the boundaries are treated here by taking them into account in the conversion process as follows. Consider the field variable ϕ. The process of converting the multiple spaces of network weights $\{w_{[x_j]}^{(i)}\}_{i=1}^{n+q_4}$ with $j = \{1, 2\}$ into the single space of the nodal variable values $\{\phi^{(i)}\}_{i=1}^{n}$ is based on the following sum of squared errors

$$\sum_{k=1}^{n}\left(\sum_{i=1}^{n+q_4} H_{[x_j]}^{[0](i)}(\mathbf{x}^{(k)})w_{[x_j]}^{(i)} - \phi(\mathbf{x}^{(k)})\right)^2 + \sum_{k=1}^{2nbp_j}\left(\sum_{i=1}^{n+q_3} H_{[x_j]}^{[1](i)}(\mathbf{x}^{(k)})w_{[x_j]}^{(i)} - \frac{\partial\phi(\mathbf{x}^{(k)})}{\partial x_j}\right)^2 \rightarrow 0, \tag{37}$$

where n is the number of collocation points (also centres) including corner points, nbp_j the number of boundary points on the cavity edge normal to the x_j direction, and q_3 and q_4 the numbers of nodal integration constants. The first term is the functional network representing the variable ϕ over the whole domain, while the second term represents the given Neumann boundary conditions $\partial\phi/\partial x_j$ along two boundaries normal to the x_j direction. After solving (37) using the general linear least squares technique [18], the network weights will be expressed in terms of the nodal variable values

$$\mathbf{w}_{[x_j]} = \mathbf{C}_{[x_j]}^{-1}\boldsymbol{\phi}, \tag{38}$$

where $\mathbf{C}_{[x_j]}$ is the conversion matrix associated with the x_j direction. Like the approach previously presented for thin-plate problems, the increased number of columns due to the integration constants is offset by the increased number of rows due to the boundary conditions in the construction of conversion matrices $\mathbf{C}_{[x_j]}$.

Expressions of ϕ and its derivatives at point $\mathbf{x}_0$ in terms of $\boldsymbol{\phi} = \{\phi^{(i)}\}_{i=1}^{n}$ will be

expressed as

$$\phi(\mathbf{x}_0) = \frac{1}{2}\sum_{j=1}^{2}\left(\left[H_{[x_j]}^{[0](1)}(\mathbf{x}_0),\cdots,H_{[x_j]}^{[0](m+1)}(\mathbf{x}_0),\cdots\right]\mathbf{C}_{[x_j]}^{-1}\boldsymbol{\phi}\right), \tag{39}$$

$$\frac{\partial\phi^2(\mathbf{x}_0)}{\partial x_j^2} = \left[H_{[x_j]}^{[2](1)}(\mathbf{x}_0),\cdots,H_{[x_j]}^{[2](m+1)}(\mathbf{x}_0),\cdots\right]\mathbf{C}_{[x_j]}^{-1}\boldsymbol{\phi}, \tag{40}$$

$$\frac{\partial^4\phi(\mathbf{x}_0)}{\partial x_j^4} = \left[g_{[x_j]}^{(1)}(\mathbf{x}_0),\cdots,0,\cdots\right]\mathbf{C}_{[x_j]}^{-1}\boldsymbol{\phi}, \tag{41}$$

$$\begin{aligned}\frac{\partial^4\phi(\mathbf{x}_0)}{\partial x_i^2\partial x_j^2} = \frac{1}{2}(&\left[H_{[x_i]}^{[2](1)}(\mathbf{x}_0),\cdots,H_{[x_i]}^{[2](m+1)}(\mathbf{x}_0),\cdots\right](\mathbf{H}_{[x_i]}^{[0]})^{-1}\left(\mathbf{H}_{[x_j]}^{[2]}\mathbf{C}_{[x_j]}^{-1}\boldsymbol{\phi}\right)\\ &+\left[H_{[x_j]}^{[2](1)}(\mathbf{x}_0),\cdots,H_{[x_j]}^{[2](m+1)}(\mathbf{x}_0),\cdots\right](\mathbf{H}_{[x_j]}^{[0]})^{-1}\left(\mathbf{H}_{[x_i]}^{[2]}\mathbf{C}_{[x_i]}^{-1}\boldsymbol{\phi}\right)).\end{aligned} \tag{42}$$

Similarly, the multiple spaces of network weights $\{w_{[x_j]}^{(i)}\}_{i=1}^{n+q_2}$ with $j=\{1,2\}$ for the field variable T can be converted into the single space of nodal variable values $\{T^{(i)}\}_{i=1}^{n}$. The IRBFN representations for ϕ and T are substituted into the governing equations (35)-(36). By discretizing the governing equations at a set of interior points and then imposing the given Dirichlet boundary conditions, a square system of algebraic equations will be obtained. The dimension of the system matrix is only $2nip \times 2nip$ in which nip is the number of interior points. It can be seen that by making use of all given derivatives on the boundaries in conversion processes, one can produce a square system of algebraic equations whatever the order of the governing PDEs.

In present work, the non-linearity of the resultant discretized system is handled by using trust region methods that retain two best features: rapid local convergence of the Newtonian iteration method and strong global convergence of the Cauchy method [30].

5.3 Numerical results

The width of the ith RBF ($a^{(i)}$) is simply chosen to be the minimum distance from the ith centre to neighbouring centres ($\beta = 1$). A number of uniform densities, namely 11×11, 21×21, 31×31, 41×41, 51×51 and 57×57 data points, are employed to study this problem for a wide range of the Rayleigh number from 10^3 to 10^7. The natural convection flow is solved as steady flow problem. In the following simulations, the initial solutions are chosen to be zero-solutions (the condition of fluid at rest) if $Ra = 10^3$ and lower-Ra-number solutions if $Ra > 10^3$. The sizes of the system matrices obtained here are smaller than those yielded through the case of introducing a new variable (the vorticity $\omega = \nabla^2\phi$). For example, with a density of 31×31 points, the present matrix size is 58×58, while it is 87×87 in the case of using the stream function, vorticity and temperature formulation. The dimensionless velocity in the present non-dimensional scheme is related to the one in [28] according to

$$Ra(u_j)_{\text{present}} = (u_j)_{\text{benchmark}}.$$

Some important measures associated with this type of flow are

- Maximum horizontal velocity $(u_1)_{max}$ on the vertical mid-plane and its location,
- Maximum vertical velocity $(u_2)_{max}$ on the horizontal mid-plane and its location,
- The average Nusselt number throughout the cavity, which is defined as

$$\overline{Nu} = \int_0^1 Nu(x_1)dx_1, \tag{43}$$

$$Nu(x_1) = \int_0^1 (u_1 T - \frac{\partial T}{\partial x_1})dx_2. \tag{44}$$

Integrals (44) and (43) are computed using Simpson rule. Results of $(u_1)_{max}$, $(u_2)_{max}$, their positions and $\overline{N_u}$ for $Ra = \{10^4, 10^5, 10^6, 10^7\}$ are displayed in Tables 3-6, respectively, where finer densities are used for higher Ra values. In all cases, the errors relative to the benchmark solutions consistently reduce with an increase in data density, indicating "mesh convergence".

The successful simulations of natural convection flow at $Ra = 10^7$ using relatively coarse densities $(21 \times 21, \cdots, 57 \times 57)$ demonstrate the effectiveness of the proposed methods. As can be seen from Figure 4, velocity boundary layers significantly increase in strength with an increasing Rayleigh number.

General results for flow at $Ra = 10^7$ in the form of stream function, vorticity and temperature contour plots are displayed in Figure 5. Each plot draws 21 contour lines whose values vary uniformly from the minimum to maximum values. The temperature, vorticity and stream function fields are skew-symmetric with regard to the geometric centre of the cavity (centro-symmetric).

6 CONCLUDING REMARKS

Two versions of indirect RBFNs are developed for the numerical solution of high-order PDEs. In the first version, the IRBFN formulations are written directly in terms of network weights, while in the second version, they are expressed in terms of nodal variable values. The first version is more straightforward to implement, but produces much larger system matrices than the second version. The emphasis here is placed on the advantages of using integration in constructing neural networks with regard to the implementation of multiple boundary conditions (straightforwardly) and the approximation of high-order derivatives (accurately). The present unsymmetric IRBFN collocation method is verified successfully through the solution of

thin-plate bending problems and natural convection flows which are governed by biharmonic equations. Accurate results and high rates of convergence are achieved.

APPENDIX

The following are new basis functions obtained from integrating MQ-RBFs by using MATHEMATICA.

$$H^{[3](i)}_{[x_j]} = \frac{(x_j - c_j^{(i)})}{2} A + \frac{C}{2} B, \tag{45}$$

$$H^{[2](i)}_{[x_j]} = \left(\frac{(x_j - c_j^{(i)})^2}{6} - \frac{C}{3} \right) A + \frac{C(x_j - c_j^{(i)})}{2} B, \tag{46}$$

$$H^{[1](i)}_{[x_j]} = \left(\frac{(x_j - c_j^{(i)})^3}{24} - \frac{13C(x_j - c_j^{(i)})}{48} \right) A + \left(\frac{C(x_j - c_j^{(i)})^2}{4} - \frac{C^2}{16} \right) B, \tag{47}$$

$$H^{[0](i)}_{[x_j]} = \left(\frac{C^2}{45} - \frac{83C(x_j - c_j^{(i)})^2}{720} + \frac{(x_j - c_j^{(i)})^4}{120} \right) A + \left(\frac{4C(x_j - c_j^{(i)})^3}{48} - \frac{3C^2(x_j - c_j^{(i)})}{48} \right) B, \tag{48}$$

where

$$r = \|\mathbf{x} - \mathbf{c}^{(i)}\|, \quad A = \sqrt{r^2 + a^{(i)2}},$$
$$B = \ln\left((x_j - c_j^{(i)}) + \sqrt{r^2 + a^{(i)2}} \right), \quad C = r^2 - (x_j - c_j^{(i)})^2 + a^{(i)2}.$$

ACKNOWLEDGEMENTS

N. Mai-Duy wishes to thank the University of Sydney for a Sesqui postdoctoral research fellowship. The computing facilities provided by the APAC National Facilities are greatly acknowledged. The authors would like to thank the referees for their helpful comments.

REFERENCES

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

2. Kansa EJ. 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* 1990; **19**(8/9): 147-161.

3. Mai-Duy N, Tran-Cong T. Numerical solution of differential equations using multiquadric radial basis function networks. *Neural Networks* 2001; **14**(2): 185-199.

4. Sharan M, Kansa EJ, Gupta S. Application of the multiquadric method for numerical solution of elliptic partial differential equations. *Journal of Applied Science and Computation* 1997; **84**: 275-302.

5. Zerroukat M, Power H, Chen CS. A numerical method for heat transfer problems using collocation and radial basis functions. *International Journal for Numerical Methods in Engineering* 1998; **42**: 1263-1278.

6. Mai-Duy N, Tran-Cong T. Numerical solution of Navier-Stokes equations using multiquadric radial basis function networks. *International Journal for Numerical Methods in Fluids* 2001; **37**: 65-86.

7. Larsson E, Fornberg B. A numerical study of some radial basis function based solution methods for elliptic PDEs. *Computers and Mathematics with Applications* 2003; **46**: 891-902.

8. Shu C, Ding H, Yeo KS. 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* 2003; **192**: 941-954.

9. Sarler B, Perko J, Chen CS. Radial basis function collocation method solution of natural convection in porous media. *International Journal of Numerical Methods for Heat & Fluid Flow* 2004; **14**(2): 187-212.

10. Mai-Duy N. Solving high order ordinary differential equations with radial basis function networks. *International Journal for Numerical Methods in Engineering* 2005; **62**: 824-852.

11. Canuto C, Hussaini MY, Quarteroni A, Zang TA. *Spectral Methods in Fluid Dynamics*. Springer-Verlag: New York, 1988.

12. Jang SK, Bert CW, Striz AG. Application of differential quadrature to static analysis of structural components. *International Journal for Numerical Methods in Engineering* 1989; **28**(3): 561-577.

13. Wang X, Bert CW. A new approach in applying differential quadrature to static and free vibrational analyses of beams and plates. *Journal of Sound and Vibration* 1993; **162**(3): 566-572.

14. Shu C, Du H. Implementation of clamped and simply supported boundary conditions in the GDQ free vibration analysis of beams and plates. *International Journal of Solids and Structures* 1997; **34**(7): 819-835.

15. Shu C, Du H. A generalized approach for implementing general boundary conditions in the GDQ free vibration analysis of plates. *International Journal of Solids and Structures* 1997; **34**(7): 837-846.

16. Liu GR, Wu TY. Multipoint boundary value problems by differential quadrature method. *Mathematical and Computer Modelling* 2002; **35**: 215-227.

17. Shu C. *Differential Quadrature and Its Application in Engineering*. Springer-Verlag: London, 2000.

18. Press WH, Flannery BP, Teukolsky SA, Vetterling WT. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press: Cambridge, 1988.

19. Park J, Sandberg IW. Universal approximation using radial basis function networks. *Neural Computation* 1991; **3**: 246-257.

20. Franke R. Scattered data interpolation: tests of some methods. *Mathematics of Computation* 1982; **38**(157): 181-200.

21. Madych WR, Nelson SA. Multivariate interpolation and conditionally positive definite functions. *Approximation Theory and its Applications* 1989; **4**: 77-89.

22. Madych WR, Nelson SA. Multivariate interpolation and conditionally positive definite functions, II. *Mathematics of Computation* 1990; **54**(189): 211-230.

23. Zeb A, Elliott L, Ingham DB, Lesnic D. A comparison of different methods to solve inverse biharmonic boundary value problems. *International Journal for Numerical Methods in Engineering* 1999; **45**: 1791-1806.

24. Timoshenko S, Woinowsky-Krieger S. *Theory of Plates and Shells*. McGraw-Hill: New York, 1959.

25. Leissa AW. The free vibration of rectangular plates. *Journal of Sound and Vibration* 1973; **31**: 257-293.

26. Leonard BP, Drummond JE. Why you should not use 'Hybrid', 'Power law' or related exponential schemes for convective modelling-there are much better alternatives. *International Journal for Numerical Methods in Fluids* 1995; **20**: 421-442.

27. Roache PJ. *Verification and Validation in Computational Science and Engineering*. Hermosa Publishers: Albuquerque, 1998.

28. de Vahl Davis G. Natural convection of air in a square cavity: a bench mark numerical solution. *International Journal for Numerical Methods in Fluids* 1983; **3**: 249-264.

29. Le Quere P. Accurate solutions to the square thermally driven cavity at high Rayleigh number. *Computers & Fluids* 1991; **20**(1): 29-41.

30. McCartin BJ. A model-trust region algorithm utilizing a quadratic interpolant. *Journal of Computational and Applied Mathematics* 1998; **91**: 249-259.

Table 1: Clamped rectangular plates of dimension $[-a/2, a/2] \times [-b/2, b/2]$ under a uniform load q, 11×11 data points: deflections at the centre and bending moments at $(x_1 = a/2, x_2 = 0)$. The computed solutions are in good agreement with the analytical solutions [24].

	$v/(qa^4/D) \times 10^2$		$M_x/(qa^2) \times 10$	
b/a	IRBFN	analytical	IRBFN	analytical
1.0	0.1265	0.126	-0.5138	-0.513
1.1	0.1507	0.150	-0.5813	-0.581
1.2	0.1724	0.172	-0.6392	-0.639
1.3	0.1911	0.191	-0.6871	-0.687
1.4	0.2068	0.207	-0.7259	-0.726
1.5	0.2196	0.220	-0.7565	-0.757
1.6	0.2300	0.230	-0.7802	-0.780
1.7	0.2382	0.238	-0.7982	-0.799
1.8	0.2446	0.245	-0.8116	-0.812
1.9	0.2495	0.249	-0.8214	-0.822
2.0	0.2533	0.254	-0.8283	-0.829

Table 2: Free vibration, simply supported plate, $[0,1] \times [0,1]$: natural frequencies. The present results agree well with those obtained by the DQ method [17] and the Rayleigh-Ritz (R-R) method [25]. By regarding the results from the R-R method as exact solutions, it can be seen that the errors (%) consistently reduce with an increase in data density.

	IRBFN			DQ	R-R
Ω	7×7(error)	9×9(error)	11×11(error)	12×12(error)	
Ω_1	19.7403(0.005)	19.7398(0.003)	19.7395(0.001)	19.7392(0.000)	19.7392
Ω_2	49.2892(0.119)	49.3533(0.010)	49.3528(0.009)	49.3495(0.003)	49.3480
Ω_3	49.2892(0.119)	49.3533(0.010)	49.3528(0.009)	49.3495(0.003)	49.3480
Ω_4	78.6314(0.412)	78.9386(0.023)	78.9584(0.002)	78.9589(0.002)	78.9568
Ω_5	98.3690(0.331)	98.7330(0.037)	98.7256(0.030)	98.4154(0.284)	98.6960

Table 3: Natural convection flow, $Ra = 10^4$.

Data density	Characteristic values				
	$(u_1)_{max}$ (error %)	x_2	$(u_2)_{max}$ (error %)	x_1	$\overline{N_u}$ (error %)
11×11	16.330(0.93)	0.823	19.778(0.82)	0.119	2.242(0.04)
21×21	16.209(0.19)	0.823	19.653(0.18)	0.118	2.244(0.04)
Benchmark	16.178	0.823	19.617	0.119	2.243

Table 4: Natural convection flow, $Ra = 10^5$.

Data density	Characteristic values				
	$(u_1)_{max}$ (error %)	x_2	$(u_2)_{max}$ (error %)	x_1	$\overline{N_u}$ (error %)
11×11	32.58(6.19)	0.856	59.71(12.9)	0.075	4.127(8.67)
21×21	34.99(0.74)	0.854	68.93(0.49)	0.065	4.532(0.28)
31×31	34.86(0.37)	0.854	68.78(0.27)	0.065	4.526(0.15)
Benchmark	34.73	0.855	68.59	0.066	4.519

Table 5: Natural convection flow, $Ra = 10^6$.

Data density	Characteristic values				
	$(u_1)_{max}$ (error %)	x_2	$(u_2)_{max}$ (error %)	x_1	$\overline{N_u}$ (error %)
11×11	65.31(1.05)	0.900	107.14(51.1)	0.061	5.840(33.6)
21×21	61.59(4.70)	0.857	198.62(9.45)	0.040	8.153(7.35)
31×31	64.62(0.01)	0.850	218.86(0.22)	0.037	8.782(0.20)
Benchmark	64.63	0.850	219.36	0.037	8.800

Table 6: Natural convection flow, $Ra = 10^7$.

Data density	Characteristic values				
	$(u_1)_{max}$ (error %)	x_2	$(u_2)_{max}$ (error %)	x_1	$\overline{N_u}$ (error %)
21×21	129.2(13.0)	0.908	390.8(44.0)	0.031	11.203(32.1)
31×31	132.1(11.0)	0.909	561.4(19.6)	0.024	14.075(14.8)
41×41	139.1(6.3)	0.895	648.8(7.2)	0.022	15.628(5.4)
51×51	144.7(2.6)	0.884	683.9(2.1)	0.021	16.264(1.5)
57×57	146.7(1.2)	0.880	692.9(0.8)	0.021	16.422(0.6)
Benchmark	148.5	0.879	699.1	0.021	16.523

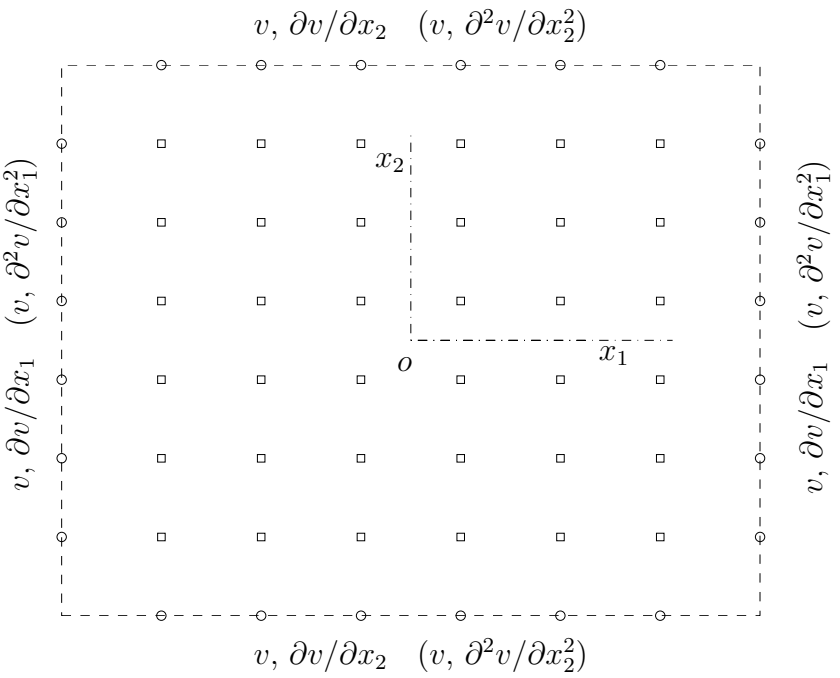


Figure 1: Biharmonic problem: geometry definition, boundary conditions and discretization. Legends ∘: boundary point and □: interior point. The domain is simply represented by the set of discrete points.

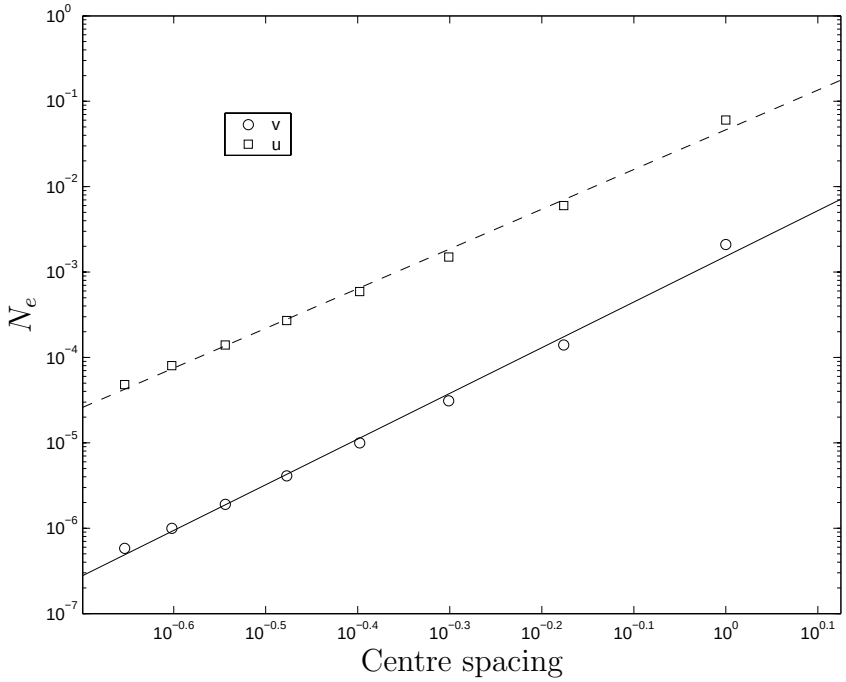


Figure 2: Biharmonic problem, domain $[-2,2]\times[-2,2]$, boundary conditions v and $\partial v/\partial n$: accurate results and high rates of convergence are achieved. The solutions converge apparently as $O(h^{5.3})$ (solid line) and $O(h^{4.6})$ (dashed line) for v and u, respectively, where h is the centre spacing. At the highest density of 19×19, error-norms are 5.8×10^{-7} for v and 4.8×10^{-5} for u.

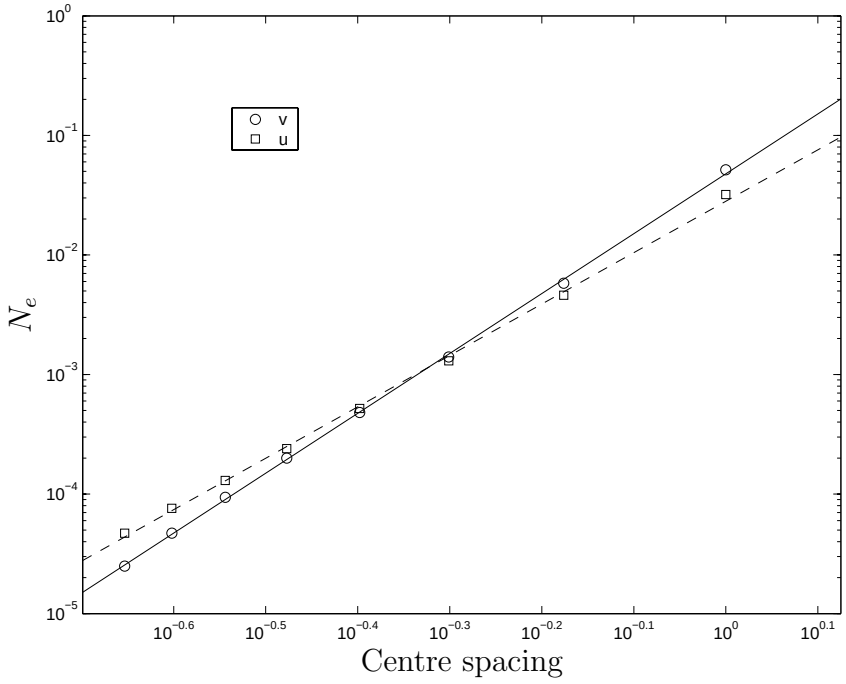


Figure 3: Biharmonic problem, domain $[-2,2]\times[-2,2]$, boundary conditions v and $\partial^2 v/\partial n^2$: accurate results and high rates of convergence are achieved. The solutions converge apparently as $O(h^{5.0})$ (solid line) and $O(h^{4.3})$ (dashed line) for v and u, respectively, where h is the centre spacing. At the highest density of 19×19, error-norms are 2.5×10^{-5} for v and 4.7×10^{-5} for u.

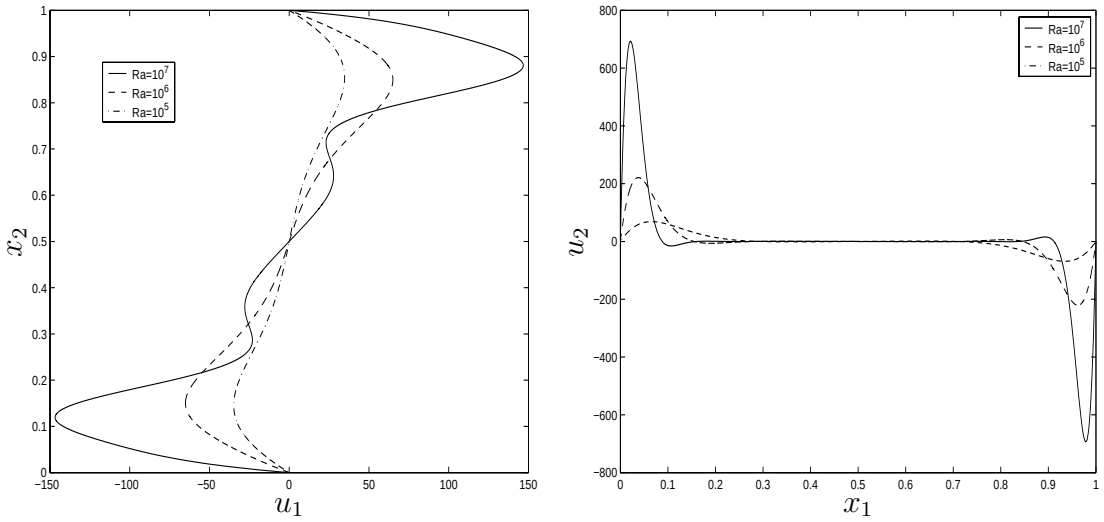
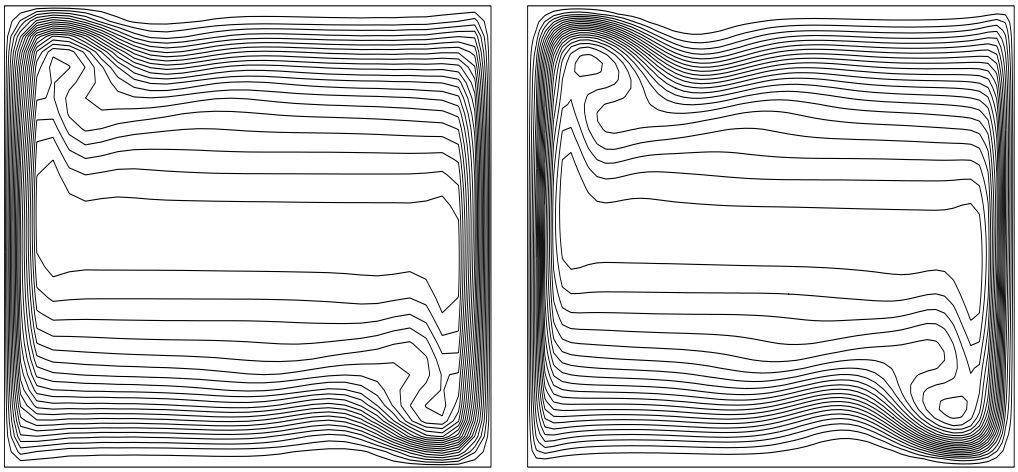
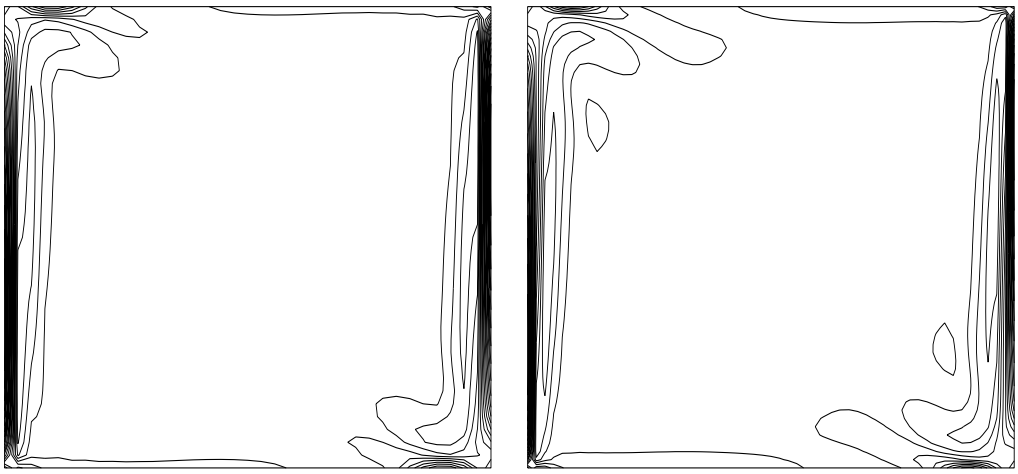


Figure 4: Natural convection flow, 57×57: velocity profiles along the centre lines for various Rayleigh numbers. The velocity boundary layers significantly increase in strength with an increasing Rayleigh number.

a) stream function



b) vorticity



c) temperature

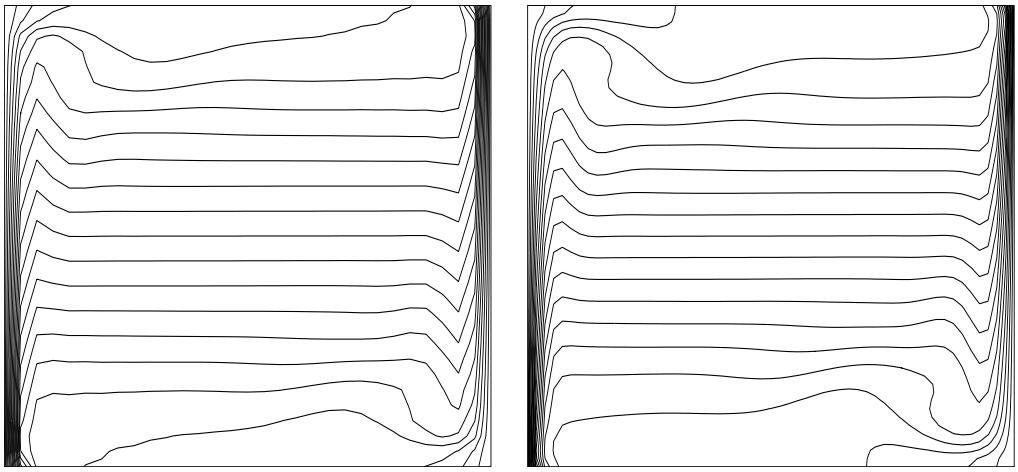


Figure 5: Natural convection flow, $Ra = 10^7$: contour plots of stream function, vorticity and temperature for the two densities 31×31 (left) and 57×57 (right). The successful simulation of flow at $Ra = 10^7$ using only 31×31 data points demonstrates the effectiveness of the proposed method.