Indirect RBFN Method with Scattered Points for Numerical Solution of PDEs

Nam Mai-Duy*

School of Aerospace, Mechanical and Mechatronic Engineering,

The University of Sydney, NSW 2006, Australia

Submitted to Computer Modeling in Engineering & Sciences,
June 2, 2003; revised March 18, 2004

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

Abstract This paper is concerned with the use of the indirect radial basis function network (RBFN) method in solving partial differential equations (PDEs) with scattered points. Indirect RBFNs (Mai-Duy and Tran-Cong, 2001a), which are based on an integration process, are employed to approximate the solution of PDEs via point collocation mechanism in the set of randomly distributed points. The method is tested with the solution of Poisson's equations and the Navier-Stokes equations (Boussinesq material). Good results are obtained using relatively low numbers of data points. For example, the natural convection flow in a square cavity at Rayleigh number of $1.e6$ is simulated successfully using only 1693 random collocation points.

1 Introduction

The traditional methods (cf. finite difference method (FDM), finite element method (FEM), boundary element method (BEM) and finite volume method (FVM)) have been well established and have acquired a great success in solving engineering and science problems. In those methods, each dependent variable in the governing equations is approximated by a set of piecewise continuous functions defined over grids, volumes or elements (mesh). The generation of mesh can require a lot of effort especially for 3D problems or even for 2D problems with large deformations or complex geometries. Meshing or re-meshing remains one of the biggest challenges in the traditional methods.

The idea of developing numerical methods without using a mesh for the solution of PDEs has recently received a great deal of attention from researchers. As a result, the number of reports on mesh-free numerical methods and their applications has increased quickly. Reviews on meshless methods can be found in Jin, Li and Aluru (2001), Li and Liu (2002), Atluri and Shen (2002) and Liu (2003). Based on the criterion of a mesh requirement, they can be classified into the so-called "meshless methods" and "truly meshless methods" categories. In the first category, some background mesh is still necessary for the purpose of either interpolation of the solution or integration of the weak form/inverse statement (e.g. element-free Galerkin method (Belytschko, Lu and Gu, 1994)), while in the second category, no mesh is required for any process involved in a method (e.g. RBFN-based method (Kansa, 1990), meshless local Petrov-Galerkin method (Atluri and Shen, 2002) and local radial point interpolation method (Liu and Gu, 2001)). Solutions of a variety problems such as 3D elasticity (Li, Shen, Han and Atluri, 2003), thin plate bending problems (Long and Atluri, 2002; Sladek, Sladek and Mang, 2002), free and forced vibrations of thick rectangular plates (Qian, Batra and Chen, 2003), beam problems (Raju and Phillips, 2003), elastodynamic problems in continuously nonhomogeneous solids (Sladek, Sladek and Zhang, 2003), 3D Stokes flows (Tsai, Young and Cheng, 2002), viscous fluid

flows (Mai-Duy and Tran-Cong, 2003b; Shu, Ding and Yeo, 2003) and others by using meshless methods have been reported in the recent literature.

Neural networks, which are able to approximate any continuous function to any degree of accuracy (universal approximation), have found application in many disciplines: neurosciences, mathematics, statistics, physics, computer science and engineering (Haykin, 1999). The application of neural networks for the solution of PDEs was first introduced by Kansa (1990) in the case of radial basis function networks (RBFNs) and by Dissanayake and Phan-Thien (1994) in the case of multilayer perceptron networks (MLPs). Since the training process of the former is generally faster than that of the latter, the RBFN-based methods have been used considerably more than the MLP-based methods in solving PDEs. In a standard RBFN-based method (DRBFN), global RBFNs are employed to represent the solution of PDEs via the process of differentiation and point collocation mechanism. Recently, Mai-Duy and Tran-Cong (2001a) proposed an alternative RBFN approach, namely the indirect RBFN (IRBFN) method, based on an integration process and point collocation to approximate the solution variables. Unlike the case of FEM and BEM, where the integration process is used to reduce the order of the continuity required for the variables, the integration process here is employed only for the purpose of obtaining new basis functions from RBFs. In the context of meshless methods, all relevant integrals in the IRBFN method can be found analytically leading to a truly meshless method in a straightforward manner, while the integration of the weak form associated with FEM or the inverse statement associated with BEM must be computed numerically and hence they require some special treatments (e.g. using local weak form/local inverse statement) in order to achieve a mesh-free feature.

In the previous works, the indirect RBFN method was verified successfully with a series of problems including functional approximations (Mai-Duy and Tran-Cong, 2002,2003a), heat transfer problems (Mai-Duy and Tran-Cong, 2001a,2003b) and Newtonian fluid flows (Mai-Duy and Tran-Cong, 2001b,2003b). Numerical examples showed that the indirect

RBFN method performs better than the usual direct RBFN method (DRBFN) in terms of solution accuracy and convergence rate. However, all those simulations have been carried out with the use of uniformly distributed centres only. The case of regular points allows the nodal constants arising from the integration process to be captured in a direct manner without making any difficulty. In addition, the influence from each centre to its neighbours in the approximation of the solution variables can be adjusted conveniently because of its uniform distribution.

In the present work, scattered points are employed to discretize the domain of analysis and special care needs be paid to the treatment of integration constant values. An advantage of the present procedure is that there is not any connectivity requirement between the data points. The IRBFN method with scattered points is verified through the simulation of heat transfer and natural convection flow problems. In solving the Navier-Stokes equations for 2D problems, it is widely acknowledged that numerical methods usually employ the stream function-vorticity formulation rather than the velocity-pressure formulation. The advantages of the former over the latter are that the number of variables is reduced to two (without pressure) and the continuity equation is automatically satisfied. However, one concern is the need to derive a boundary condition for the vorticity. The stream function-vorticity formulation will be adopted in the present work, where the Dirichlet boundary conditions for the vorticity are generated in a precise manner based on the global approximations. Accurate results are obtained for both heat transfer and natural convection flow problems using relatively low numbers of data points. For example, a Rayleigh number of $1.e6$ is achieved using 1693 randomly distributed data points. It is noted that the DRBFN method and the FEM are also included in some cases to provide the basis for the assessment of the present IRBFN method.

The remainder of the paper is organized as follows. In section 2, the governing equations for heat transfer and natural convection flow problems are given. Section 3 gives a brief review of RBFNs. Section 4 presents the indirect RBFN approach with scattered

points. The direct RBFN approach is also summarized here. In section 5, the present method is tested with the solution of Poisson's equations and the Navier-Stokes equations (Boussinesq material). Section 6 gives some concluding remarks.

2 Governing equations

Heat transfer problems governed by Poisson equations and natural convection flows in a square slot governed by the Navier-Stokes equations (Boussinesq material) are considered in the present work.

2.1 Heat transfer problem

The governing Poisson's equation takes the form

$$\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} = s(\mathbf{x}), \qquad \mathbf{x} \in \Omega, \tag{1}$$

where $\mathbf{x} = (x_1, x_2)$ is the position vector of a point in the analysis domain Ω, u is the dependent variable and s is a known function.

2.2 Natural convection flow in a square slot

The stream function-vorticity formulation governing the temperature and velocity behaviour of natural convection flows is adopted in the present work. The governing equations are non-dimensionalized by using a scheme similar to that of Leonard and Drummond (1995). The cavity dimensions x_i are scaled by the characteristic length L while the temperature T by the temperature difference between the two vertical walls ΔT. The ve-

locities are scaled by $U = \alpha g L^2 \Delta T/\nu$, where α is the coefficient of volumetric expansion, ν is the kinematics viscosity and g the strength of the gravitational field. The vorticity ω and stream function ϕ are scaled with U/L and UL respectively. Applying the standard Boussinesq approximation, i.e. the fluid is assumed to have constant properties except for the generation of buoyant force, yields the following forms of the Poisson, vorticity transport and energy equations

$$\frac{\partial^2 \phi}{\partial x_1^2} + \frac{\partial^2 \phi}{\partial x_2^2} + \omega = 0, \qquad \mathbf{x} \in \Omega, \tag{2}$$

$$\frac{\partial^2 \omega}{\partial x_1^2} + \frac{\partial^2 \omega}{\partial x_2^2} + \frac{\partial T}{\partial x_1} = Gr \left(\frac{\partial \phi}{\partial x_2}\frac{\partial \omega}{\partial x_1} - \frac{\partial \phi}{\partial x_1}\frac{\partial \omega}{\partial x_2} \right), \qquad \mathbf{x} \in \Omega, \tag{3}$$

$$\frac{\partial^2 T}{\partial x_1^2} + \frac{\partial^2 T}{\partial x_2^2} = PrGr \left(\frac{\partial \phi}{\partial x_2}\frac{\partial T}{\partial x_1} - \frac{\partial \phi}{\partial x_1}\frac{\partial T}{\partial x_2} \right), \qquad \mathbf{x} \in \Omega. \tag{4}$$

The dimensionless parameters appearing in the equations (2)-(4) are respectively the Grashof number, the Prandtl number and the Rayleigh number

$$Gr = UL/\nu, \qquad Pr = \nu/\kappa, \qquad Ra = GrPr,$$

where κ is the thermal diffusivity. The vorticity and stream function are defined by

$$\omega = \frac{\partial u_2}{\partial x_1} - \frac{\partial u_1}{\partial x_2}, \tag{5}$$

$$\frac{\partial \phi}{\partial x_2} = u_1, \quad \frac{\partial \phi}{\partial x_1} = -u_2, \tag{6}$$

where u_1 and u_2 are the components of the velocity vector.

The present scheme of non-dimensionalization results in the dimesionless velocity that is related to the one produced by the scheme used in the benchmark solution (de Vahl Davis, 1983) according to

$$Ra(u_{present}) = u_{bench}.$$

Boundary conditions

The boundary conditions for this problem are as follows

$$\psi = 0, \qquad \frac{\partial \phi}{\partial x_2} = 0, \qquad \frac{\partial T}{\partial x_2} = 0 \text{ on the line } x_2 = 0, \tag{7}$$

$$\psi = 0, \qquad \frac{\partial \phi}{\partial x_1} = 0, \qquad T = 0 \quad \text{ on the line } x_1 = 1, \tag{8}$$

$$\psi = 0, \qquad \frac{\partial \phi}{\partial x_2} = 0, \qquad \frac{\partial T}{\partial x_2} = 0 \text{ on the line } x_2 = 1, \tag{9}$$

$$\psi = 0, \qquad \frac{\partial \phi}{\partial x_1} = 0, \qquad T = 1 \quad \text{ on the line } x_1 = 0. \tag{10}$$

3 Radial Basis Function Networks

A function y, to be approximated, can be represented by an RBFN as follows (Haykin, 1999)

$$y(\mathbf{x}) \approx f(\mathbf{x}) = \sum_{i=1}^{m} w^{(i)} g^{(i)}(\mathbf{x}), \tag{11}$$

where superscripts denote elements of a set of neurons (RBFs), $\mathbf{x}$ is 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) obtained from (11) are always invertible provided that the data points are distinct. This is all that is required for nonsingularity of the design matrix, whatever the number of data points and the dimension of problem (Haykin, 1999).

Since multiquadrics (MQ) are ranked the best in terms of accuracy among RBFs (Franke, 1982) and also have exponential convergence with the refinement of spatial discretization (Madych and Nelson, 1989,1990), the present work will employ these basis functions whose

form is

$$g^{(i)}(\mathbf{x}) = \sqrt{(\mathbf{x} - \mathbf{c}^{(i)})^T(\mathbf{x} - \mathbf{c}^{(i)}) + a^{(i)2}}, \tag{12}$$

where $\mathbf{c}^{(i)}$ and $a^{(i)}$ are the centre and the width of the ith MQ basis function respectively and superscript T denotes the transpose of a matrix. To make the training process simple, the centres and the widths of RBFs are chosen in advance. For the latter, the following relation is used

$$a^{(i)} = \beta d^{(i)}, \tag{13}$$

where β is a positive scalar and $d^{(i)}$ is the minimum of distances from the ith center to its neighbours. The relation (13) allows the RBF width a to be broader in the area of lower data density and narrower in the area of higher data density. Hence, with this relation, the widths vary with the centres for the case of scattered points, but remain unchanged for the case of uniformly distributed data points.

4 Direct and Indirect approaches

4.1 Direct approach

In the direct RBFN approach, the RBF network (11) is utilized to represent the original function y and subsequently its derivatives are computed by differentiating (11) as follows

$$y(\mathbf{x}) \approx f(\mathbf{x}) = \sum_{i=1}^{m} w^{(i)} g^{(i)}(\mathbf{x}), \tag{14}$$

$$\frac{\partial y(\mathbf{x})}{\partial x_j} \approx \frac{\partial f(\mathbf{x})}{\partial x_j} = \frac{\partial \left(\sum_{i=1}^{m} w^{(i)} g^{(i)}(\mathbf{x})\right)}{\partial x_j} = \sum_{i=1}^{m} w^{(i)} h^{(i)}_{[x_j]}(\mathbf{x}), \tag{15}$$

$$\frac{\partial^2 y(\mathbf{x})}{\partial x_j^2} \approx \frac{\partial^2 f(\mathbf{x})}{\partial x_j^2} = \frac{\partial \left(\sum_{i=1}^{m} w^{(i)} h^{(i)}_{[x_j]}(\mathbf{x})\right)}{\partial x_j} = \sum_{i=1}^{m} w^{(i)} \bar{h}^{(i)}_{[x_j]}(\mathbf{x}), \tag{16}$$

where subscripts j denote scalar components of a vector; $h^{(i)}_{[x_j]}(\mathbf{x}) = \partial g^{(i)}(\mathbf{x})/\partial x_j$ and $\bar{h}^{(i)}_{[x_j]}(\mathbf{x}) = \partial h^{(i)}(\mathbf{x})/\partial x_j$ are new derived basis functions in the approximation of the first and second order derivatives of y respectively.

4.2 Indirect approach

In this approach, RBFNs are used to represent the highest order derivatives of a function y under consideration, e.g. $\partial^2 y/\partial x_1^2$ and $\partial^2 y/\partial x_2^2$. Lower order derivatives and finally the function itself are then obtained by integrating those RBFNs as follows, e.g. for 2D problems,

$$\frac{\partial^2 y(\mathbf{x})}{\partial x_j^2} \approx \frac{\partial^2 f(\mathbf{x})}{\partial x_j^2} = \sum_{i=1}^{m} w^{(i)}_{[x_j]} g^{(i)}(\mathbf{x}), \tag{17}$$

$$\begin{aligned} \frac{\partial y(\mathbf{x})}{\partial x_j} \approx \frac{\partial f(\mathbf{x})}{\partial x_j} &= \int \left(\sum_{i=1}^{m} w^{(i)}_{[x_j]} g^{(i)}(\mathbf{x}) \right) dx_j + C_{[x_j]}(x_{k,k\neq j}) \\ &= \sum_{i=1}^{m} w^{(i)}_{[x_j]} H^{(i)}_{[x_j]}(\mathbf{x}) + C_{[x_j]}(x_{k,k\neq j}), \end{aligned} \tag{18}$$

$$\begin{aligned} y(\mathbf{x}) \approx f(\mathbf{x}) &= \int \left(\sum_{i=1}^{m} w^{(i)}_{[x_j]} H^{(i)}_{[x_j]}(\mathbf{x}) \right) dx_j + x_j C_{[x_j]}(x_{k,k\neq j}) + D_{[x_j]}(x_{k,k\neq j}) \\ &= \sum_{i=1}^{m} w^{(i)}_{[x_j]} \bar{H}^{(i)}_{[x_j]}(\mathbf{x}) + x_j C_{[x_j]}(x_{k,k\neq j}) + D_{[x_j]}(x_{k,k\neq j}), \end{aligned} \tag{19}$$

where subscripts $[x_j]$ denote the quantities associated with the integration process in the x_j direction; $H^{(i)}_{[x_j]} = \int g^{(i)} dx_j$ and $\bar{H}^{(i)}_{[x_j]} = \int H^{(i)} dx_j$ are new derived basis functions in the approximation of the first order derivative and the target function y respectively; $C_{[x_j]}$ and $D_{[x_j]}$ are the integration constant functions of the variable $x_{k,k\neq j}$.

It can be seen that collocation points with the same $x_{k,k\neq j}$ coordinate have the same values of $C_{[x_j]}$ and $D_{[x_j]}$. As a result, in the case of regularly distributed data points, the number of nodal function values for $C_{[x_j]}$ or $D_{[x_j]}$ is relatively small and therefore they can be employed directly in the network design matrices. In the case of scattered data

points, it is clear that the number of nodal function values for $C_{[x_j]}$ or $D_{[x_j]}$ arising from the integration process is relatively large. The direct use of these nodal values results in very large design matrices. For example, consider a 2D domain and let n be the number of collocation points. The size of the network design matrices obtained can be up to $[n, m+2m]$. To overcome this difficulty, IRBF networks in 1D need be utilized to interpolate the integration constant functions. The univariate functions $C_{[x_j]}$ and $D_{[x_j]}$ can be represented by 1D indirect RBF networks as follows, e.g. for $C_{[x_j]}$,

$$C_{[x_j]}(x_{k,k\neq j}) = \sum_{i=1}^{p} \hat{w}_{[x_j]}^{(i)} \bar{H}_{[x_j]}^{(i)}(x_{k,k\neq j}) + x_{k,k\neq j}\hat{C}_{[x_j]} + \hat{D}_{[x_j]}, \tag{20}$$

where $\hat{C}_{[x_j]}$ and $\hat{D}_{[x_j]}$ are new integration constants (scalars) and p is the number of new centres. In the present work, p is chosen according to the relation

$$p = ft\, n_{x_k}, \tag{21}$$

where ft is a small integer and n_{x_k} is the number of distinct x_k coordinates from the special case of regularly distributed data points. For convenience, the unknown constants ($\hat{w}_{[x_j]}^{(i)}$ and $\hat{C}_{[x_j]}, \hat{D}_{[x_j]}$) and their associated known basis functions (RBFs and polynomial) are also denoted by the notations $w^{(i)}$ and $H_{[x_j]}^{(i)}(\mathbf{x})$ ($\bar{H}_{[x_j]}^{(i)}(\mathbf{x})$) respectively but with $i > m$ and hence equations (17)-(19) can be rewritten in the simple forms

$$\frac{\partial^2 f(\mathbf{x})}{\partial x_j^2} = \sum_{i=1}^{m} w_{[x_j]}^{(i)} g^{(i)}(\mathbf{x}), \tag{22}$$

$$\frac{\partial f(\mathbf{x})}{\partial x_j} = \sum_{i=1}^{m+p+2} w_{[x_j]}^{(i)} H_{[x_j]}^{(i)}(\mathbf{x}), \tag{23}$$

$$f(\mathbf{x}) = \sum_{i=1}^{m+2(p+2)} w_{[x_j]}^{(i)} \bar{H}_{[x_j]}^{(i)}(\mathbf{x}). \tag{24}$$

The evaluation of (22)-(24) at a set of collocation points $\{\mathbf{x}^{(j)}\}_{i=1}^{n}$ yields the system of

equations of the form

$$\mathbf{f}_{,\mathbf{jj}}(\mathbf{x}) = \mathbf{G}(\mathbf{x})\mathbf{w}_{[x_j]}, \tag{25}$$

$$\mathbf{f}_{,\mathbf{j}}(\mathbf{x}) = \mathbf{H}_{[x_j]}(\mathbf{x})\mathbf{w}_{[x_j]}, \tag{26}$$

$$\mathbf{f}(\mathbf{x}) = \bar{\mathbf{H}}_{[x_j]}(\mathbf{x})\mathbf{w}_{[x_j]}, \tag{27}$$

where $\mathbf{G}$, $\mathbf{H}$ and $\bar{\mathbf{H}}$ are the network design matrices associated with the approximation of the second derivative, first derivative and function respectively; $\mathbf{w}_{[x_j]}$ the set of network weights to be found; $\mathbf{f} = \{f(\mathbf{x}^{(i)})\}_{i=1}^{n}$; $\mathbf{f}_{,\mathbf{j}} = \{\frac{\partial f(\mathbf{x}^{(i)})}{\partial x_j}\}_{i=1}^{n}$ and $\mathbf{f}_{,\mathbf{jj}} = \{\frac{\partial^2 f(\mathbf{x}^{(i)})}{\partial x_j^2}\}_{i=1}^{n}$. For the purpose of computation, the matrices $\mathbf{G}$ and $\mathbf{H}$ are augmented using zero-submatrices so that they have the same size as the matrix $\bar{\mathbf{H}}$. Let N be the problem dimensionality. Since there are N vectors of network weights to be found, the size of the system of equations in the indirect approach is about N times as big as that in the direct approach. To overcome this disadvantage, prior conversions of the multiple spaces of network weights into the single space of function values $\mathbf{f}$ are employed. The process is as follows. By solving (27) using the general linear least squares, the set of network weights is expressed in terms of nodal function values as

$$\mathbf{w}_{[x_j]} = \bar{\mathbf{H}}_{[x_j]}^{-1}\mathbf{f}. \tag{28}$$

Substitution of (28) into the system (25)-(27) yields

$$\mathbf{f}_{,\mathbf{jj}} = \mathbf{G}\bar{\mathbf{H}}_{[x_j]}^{-1}\mathbf{f}, \tag{29}$$

$$\mathbf{f}_{,\mathbf{j}} = \mathbf{H}_{[x_j]}\bar{\mathbf{H}}_{[x_j]}^{-1}\mathbf{f}, \tag{30}$$

$$\mathbf{f} = \mathbf{I}\mathbf{f}, \tag{31}$$

where $\mathbf{I}$ is the unit matrix.

It can be seen from (29)-(31) that the function and its derivatives are all expressed in

terms of the nodal function values rather than in terms of the network weights. As a result, the system of equations obtained is normally square with the size being approximately equal to that of the DRBFN method, irrespective of the problem dimensionality. The transformation operation completely eliminates the problem of the large matrix size in the IRBFN method.

4.3 Numerical solution of PDEs using RBFNs

Each variable and its derivatives in the governing equations can be represented by neural networks using either (14)-(16) in the direct approach or (29)-(31) in the indirect approach. The closed-form representations obtained are substituted into the governing equations and/or boundary conditions and the system is then discretized via point collocation mechanism. In the present work, the set of collocation points is chosen to be the same as the set of centres, i.e. $\{x^{(i)}\}_{i=1}^{n} = \{c^{(i)}\}_{i=1}^{m}$ with $n = m$. The RBFN solution satisfies the governing equations pointwise rather than in an average sense. In order to form the square system of equations, the governing equations are applied to the interior points only.

In the indirect RBFN approach, the unknown vector contains the nodal variable values, e.g. for u in Poisson's equation and for $\{\phi, \omega, T\}$ in the Navier-Stokes equations, resulting in ease of dealing with the essential boundary conditions; while in the direct RBFN approach, the unknowns are the network weights (coefficients), leading to a further employment of collocation points along the boundaries to enforce the boundary conditions. However, for both approaches, the determination of the unknowns can be seen to be based on the process of minimizing the following sum squared errors SSE

$$SSE = SSE_1 + SSE_2, \tag{32}$$

where SSE_1 and SSE_2 are the sums of squared errors, which are employed to ensure that neural networks satisfy the governing equations and boundary conditions respectively. The form of SSE_2 depends on the problem to be solved while the term SSE_1 can be written in the general form provided that the governing equations are given. For example, the SSE_1 in the IRBFN formulation takes the form

$$SSE_1 = \sum_{i=1}^{nip} \left[\sum_{j=1}^{m} \left(\mathbf{G}\bar{\mathbf{H}}_{[x_1]}^{-1} \right)_{[i,j]} u^{(j)} + \sum_{j=1}^{m} \left(\mathbf{G}\bar{\mathbf{H}}_{[x_2]}^{-1} \right)_{[i,j]} u^{(j)} - s(\mathbf{x}^{(i)}) \right]^2 \tag{33}$$

for Poisson's equation (1) and

$$\begin{aligned} SSE_1 = & \sum_{i=1}^{nip} \{ \sum_{j=1}^{m} \left(\mathbf{G}\bar{\mathbf{H}}_{[x_1]}^{-1} \right)_{[i,j]} \phi^{(j)} + \sum_{j=1}^{m} \left(\mathbf{G}\bar{\mathbf{H}}_{[x_2]}^{-1} \right)_{[i,j]} \phi^{(j)} - \omega^{(i)}] \}^2 + \\ & \sum_{i=1}^{nip} \{ \sum_{j=1}^{m} \left(\mathbf{G}\bar{\mathbf{H}}_{[x_1]}^{-1} \right)_{[i,j]} \omega^{(j)} + \sum_{j=1}^{m} \left(\mathbf{G}\bar{\mathbf{H}}_{[x_2]}^{-1} \right)_{[i,j]} \omega^{(j)} + \sum_{j=1}^{m} \left(\mathbf{H}_{[x_1]}\bar{\mathbf{H}}_{[x_1]}^{-1} \right)_{[i,j]} T^{(j)} - \\ & Gr [\sum_{j=1}^{m} \left(\mathbf{H}_{[x_2]}\bar{\mathbf{H}}_{[x_2]}^{-1} \right)_{[i,j]} \phi^{(j)} \left(\mathbf{H}_{[x_1]}\bar{\mathbf{H}}_{[x_1]}^{-1} \right)_{[i,j]} \omega^{(j)} - \\ & \sum_{j=1}^{m} \left(\mathbf{H}_{[x_1]}\bar{\mathbf{H}}_{[x_1]}^{-1} \right)_{[i,j]} \phi^{(j)} \left(\mathbf{H}_{[x_2]}\bar{\mathbf{H}}_{[x_2]}^{-1} \right)_{[i,j]} \omega^{(j)}] \}^2 + \\ & \sum_{i=1}^{nip} \{ \sum_{j=1}^{m} \left(\mathbf{G}\bar{\mathbf{H}}_{[x_1]}^{-1} \right)_{[i,j]} T^{(j)} + \sum_{j=1}^{m} \left(\mathbf{G}\bar{\mathbf{H}}_{[x_2]}^{-1} \right)_{[i,j]} T^{(j)} - \\ & PrGr [\sum_{j=1}^{m} \left(\mathbf{H}_{[x_2]}\bar{\mathbf{H}}_{[x_2]}^{-1} \right)_{[i,j]} \phi^{(j)} \left(\mathbf{H}_{[x_1]}\bar{\mathbf{H}}_{[x_1]}^{-1} \right)_{[i,j]} T^{(j)} - \\ & \sum_{j=1}^{m} \left(\mathbf{H}_{[x_1]}\bar{\mathbf{H}}_{[x_1]}^{-1} \right)_{[i,j]} \phi^{(j)} \left(\mathbf{H}_{[x_2]}\bar{\mathbf{H}}_{[x_2]}^{-1} \right)_{[i,j]} T^{(j)}] \}^2 \end{aligned} \tag{34}$$

for the Navier-Stokes equations (2)-(4), where nip is the number of interior points and $[i,j]$ denotes the element located at the row i and column j of a matrix. The present method thus appears to be close to the FDM in the sense that the variables in both methods are all expressed in terms of nodal variable values and the governing equations are discretized via point collocation mechanism, i.e. they are approximated directly in the strong form. However, the IRBFN method is based on high order approximation schemes

with global properties (mesh-free universal approximator RBFNs), while the FDM uses low order approximation schemes with local properties, usually truncated Taylor series expansions or polynomial approximations.

5 Numerical examples

In the following simulations, the variable quantities β, ft and n_{x_j} are simply chosen to be 1, 3 and $\sqrt{nip}$ (nip: number of interior points) respectively, i.e. $a^{(i)} = d^{(i)}$ and $p \approx 3\sqrt{nip}$. The reason for choosing $ft = 3$ will be explained later on. The relation $a^{(i)} = d^{(i)}$ indicates that it is reasonable to assign a larger width where the centres are widely separated and a smaller width where the centres are closer. Results obtained are compared with the exact or benchmark solutions. The FEM and DRBFN method are considered in some cases to provide the basis for the assessment of the present IRBFN method.

5.1 Heat transfer problems

Since exact solutions for all problems under consideration here are available, the accuracy of the solution obtained is measured via the norm of relative errors of the solution as follows

$$N_e = \sqrt{\frac{\sum_{i=1}^{n} \left[u_e(x^{(i)}) - u(x^{(i)})\right]^2}{\sum_{i=1}^{n} u_e(x^{(i)})^2}}, \tag{35}$$

where u and u_e are the calculated and exact solutions respectively.

5.1.1 Poisson's equation on a square domain

The problem here is to determine a function $u(x_1, x_2)$ satisfying the following PDE

$$\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} = \sin(\pi x_1)\sin(\pi x_2) \tag{36}$$

defined on the rectangle $0 \leq x_1 \leq 1, \quad 0 \leq x_2 \leq 1$, subject to the Dirichlet condition $u = 0$ along the whole boundary of the domain. The exact solution is given by

$$u_e(x_1, x_2) = -\frac{1}{2\pi^2}\sin(\pi x_1)\sin(\pi x_2). \tag{37}$$

Four data sets of 32, 52, 89 and 145 randomly distributed interior points are employed to study "mesh" convergence (Figure 1). In the IRBFN approach, it is straightforward to impose a Dirichlet boundary condition since the variable and its derivatives are expressed in terms of nodal variable values. In the DRBFN approach, collocation points along the boundaries are used to enforce the given boundary conditions. For both approaches, the systems obtained are square. Results are displayed in Figure 2, indicating that the IRBFN approach is superior to the DRBFN in terms of both accuracy and convergence rate. Highly accurate results and high rates of convergence with the refinement of spatial discretization are obtained with the present method. The solution converges apparently as $O(nip^{-1.4574})$ for IRBFN and $O(nip^{-0.5081})$ for DRBFN, where nip is the number of interior points. At the finest density of 145 points, the error norms are $2.6437e-4$ and $2.4272e-2$ for IRBFN and DRBFN respectively.

Results, that are just presented, are valid with $ft = 3$. The effect of ft on the accuracy of the solution is studied in detail here. A wide range of ft varying from 1 to 7 with an increment of 1 is employed. As shown in Figure 3, the solution accuracy is improved with an increase in the value of ft. However, when ft is greater than 3, the error norm appears to stay constant. Hence, at $ft = 3$, the number of neurons (RBFs) used in the

approximation of integration constant functions appears to be large enough to capture well these functions. Greater values of ft can not improve the solution, but cause larger sizes for the network design matrices. The value $ft = 3$ is recommended for practical use in the case of scattered points.

5.1.2 Poisson's equation on a unit disk

The problem formulation is given by

$$\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} = -1, \qquad \mathbf{x} \in \Omega, \tag{38}$$

where Ω is the unit disk, subject to the boundary condition $u = 0$ on the boundary $\partial\Omega$. The exact solution is given by

$$u_e(x_1, x_2) = \frac{1 - x_1^2 - x_2^2}{4}.$$

In order to evaluate the performance of the present IRBFN method, the FEM is also employed to solve the problem. Three discretization schemes are shown in Figure 4, for each of which the sets of interior points are exactly the same for both methods. Note that the FEM results are obtained using the PDE tool in MATLAB. The IRBFN method achieves greater accuracy and also higher convergence rate than the FEM as shown in Figure 5. The IRBFN method converges apparently as $O(nip^{-1.2664})$ while the FEM as $O(nip^{-1.0529})$, where nip is the number of interior points. At the finest density of 480 points, the error norms are $2.0144e-5$ and $1.1750e-3$ for IRBFN and FEM respectively.

5.2 Natural convection flow in a square cavity

Natural convection flow in an enclosed square cavity provides a means to test and validate numerical methods. The problem, which is in itself of considerable practical interest, is schematically shown in Figure 6. The domain of interest is a square cavity of a unit size, containing a Boussinesq fluid of Prandtl number of 0.71. Non-slip boundary conditions are applied along all the walls. The left and right vertical walls are kept at temperatures $T = 1$ and $T = 0$ respectively, while the horizontal walls are insulated as described in (7)-(10). Only the IRBFN method is employed here and its results are compared with the benchmark solution of de Vahl Davis (1983). Special attention is given to the treatment for the Dirichlet boundary condition ω and the Neumann condition $\partial T/\partial n$.

5.2.1 Dirichlet condition for the vorticity

The direct employment of the Neumann condition $\partial\phi/\partial n$ over all boundaries via point collocation is not appropriate here due to the singular nature of the system of equations, although the system obtained can be square (Pozrikidis, 1997). Instead, it is used in generating computational boundary conditions for the vorticity ω. The process is as follows. In the first step, the vorticity in (2) can be simplified to be

$$\omega = -\frac{\partial^2\phi}{\partial x_1^2} - \frac{\partial^2\phi}{\partial x_2^2} = -\frac{\partial^2\phi}{\partial x_1^2} \qquad \text{at the side walls,} \tag{39}$$

$$\omega = -\frac{\partial^2\phi}{\partial x_1^2} - \frac{\partial^2\phi}{\partial x_2^2} = -\frac{\partial^2\phi}{\partial x_2^2} \qquad \text{at the top and bottom walls,} \tag{40}$$

using the given boundary conditions for the stream function. In the second step, they are written in terms of the first order derivatives of ϕ

$$\omega^{(i)} = = -\frac{\partial^2 \phi^{(i)}}{\partial x_1^2} = \sum_{j=1}^{m} \left(\mathbf{G}\bar{\mathbf{H}}_{[x_1]}^{-1} \right)_{[i,j]} \frac{\partial \phi}{\partial x_1}^{(j)} \qquad \text{at the side walls,} \tag{41}$$

$$\omega^{(i)} = = -\frac{\partial^2 \phi^{(i)}}{\partial x_2^2} = \sum_{j=1}^{m} \left(\mathbf{G}\bar{\mathbf{H}}_{[x_2]}^{-1} \right)_{[i,j]} \frac{\partial \phi}{\partial x_2}^{(j)} \qquad \text{at the top and bottom walls,} \tag{42}$$

and the resulting expressions (41) and (42) are then simplified by taking into consideration the Neumann conditions for ϕ ($\partial\phi/\partial n$). In the third step, the remainders of the nodal first derivatives of ϕ on the right-hand sides of (41) and (42) are expressed in terms of the nodal stream function values, for example at the boundary point $\mathbf{x^{(i)}}$,

$$\frac{\partial \phi}{\partial x_1}^{(i)} = \sum_{j=1}^{m} \left(\mathbf{H}_{[x_1]}\bar{\mathbf{H}}_{[x_1]}^{-1} \right)_{[i,j]} \phi^{(j)}, \tag{43}$$

$$\frac{\partial \phi}{\partial x_2}^{(i)} = \sum_{j=1}^{m} \left(\mathbf{H}_{[x_2]}\bar{\mathbf{H}}_{[x_2]}^{-1} \right)_{[i,j]} \phi^{(j)}. \tag{44}$$

Hence, the computational Dirichlet boundary conditions for ω are generated and written in terms of the nodal values of ϕ. It is noted that in this process, the natural boundary conditions for the stream function are implemented in "a precise manner" using the global approximations.

5.2.2 Neumann condition for the temperature

The present method implements this type of boundary condition as follows. Along the bottom ($x_2 = 0$) and top ($x_2 = 1$) sides of the domain, normal derivatives $\partial T/\partial n$ are given and hence the task now is to express the boundary values of T there in terms of the interior variable values. This can be achieved by solving the following subsystem of

equations

$$\frac{\partial T(\mathbf{x}^{(i)})}{\partial x_2} = \sum_{j=1}^{m} \left(\mathbf{H}_{[x_2]} \bar{\mathbf{H}}_{[x_2]}^{-1} \right)_{[i,j]} u^{(j)}, \tag{45}$$

where $\mathbf{x}^{(i)} = \{(x_1, x_2 = 0), (x_1, x_2 = 1)\}$. The results obtained from (45) together with the Dirichlet boundary conditions for T in (8) and (10) will be substituted into the final system of equations from which the unknown vector for the temperature T contains only the interior variable values. Once the final system of equations is solved, the numerical solution T along two horizontal sides may be found from (45).

5.2.3 Flow chart

In the present work, the Poisson's equation (2), the vorticity transport equation (3) and the energy equation (4) are solved simultaneously for the whole set of primary variables. The procedural flow chart involves the following steps

1. Input data such as geometries, boundary conditions, a Rayleigh number and data points. Here, a set of collocation points is chosen to be the same as a set of centres,

2. Apply the IRBFN approach for the approximation of each variable and its derivatives present in the relevant PDEs, which results in design matrices in the network weight spaces. It is noted that these matrices depend only on the geometry of the problem and hence they are the same for all variables,

3. Convert the multiple spaces of network weights into the single space of the primary variable values. This step involves some matrix inversions. However, their computational costs are modest because those matrices are relatively small with their size being approximately equal to $n \times n$ in which n is the number of collocation points,

4. Generate the computational Dirichlet boundary conditions for ω and T using the given Neumann boundary conditions for ϕ and T respectively,

5. Form the final system matrix, which involves only linear terms in the governing equations (2)-(4), and then impose the Dirichlet conditions for ϕ , ω and T. This matrix stays the same during the iteration process,

6. Initialize the stream function, vorticity and temperature fields,

7. Compute the nonlinear terms, in which relevant derivative functions are calculated in a straightforward manner using the network design matrices obtained at step 3,

8. Formulate the trust region subproblem and then solve it for the search direction (More and Sorensen, 1983; Branch, Coleman and Li, 1999),

9. Update the solution,

10. Check for convergence. If not yet converged, repeat from step 7,

11. Output the results.

5.2.4 Numerical results

Four data sets of 778, 999, 1168 and 1693 randomly distributed points are employed to study this problem for a wide range of the Rayleigh number $\{1.0e3, 1.0e4, 1.0e5, 1.0e6\}$. General results for this problem in the form of contour plots of the stream function, vorticity and temperature are displayed in Figures 7-10 respectively, where finer discretizations are used for higher Ra values. There is a close agreement with the results available in the literature. The temperature and stream function fields are skew-symmetric with regard to the geometric centre of the cavity (centro-symmetric). Furthermore, the temperature boundary layers at the vertical walls appear to be thinner and the isotherms are nearly horizontal in the core flow as the Rayleigh number increases. Thin boundary layers are also observed for the flow close to the walls.

Another important results associated with this type of flow are

- the Nusselt number defined by

$$\bar{Nu}(x_1) = \int_0^1 (u_1 T - T_{,1}) dx_2, \tag{46}$$

- maximum horizontal velocity on the vertical mid-plane and its location,
- maximum vertical velocity on the horizontal mid-plane and its location,

which are also studied here. Integral (46) is computed using Simpson rule. Results obtained for four data densities are displayed in Table 1, which are in good agreement with the benchmark solution of de Vahl Davis (1983).

5.2.5 Comparison with other meshless RBFN-based methods

Recently, Shu, Ding and Yeo (2003) reported the local RBF-based differential quadrature method (RBF-DQM) for numerical solution of the Navier-Stokes equations. In that method, the PDEs are approximated directly in the strong form and the derivative expressions are obtained by the differential quadrature method using radial basis functions. In solving natural convection flows in a square slot, three data sets of 2570, 5338 and 10305 randomly distributed points were employed to simulate flows for the Rayleigh number of $1.0e4$, $1.0e5$ and $1.0e6$ respectively. Hence, the numbers of scattered points used in the present simulations (999, 1276 and 1693 collocation points) are much less than those employed in the RBF-DQM. Furthermore, in generating Dirichlet boundary conditions for the vorticity, the present method does not require any special discretization in the boundary region, while three layers of orthogonal grid near and including the boundary are required in the method of Shu, Ding and Yeo (2003).

6 Concluding remarks

In this paper, the indirect RBFN method with scattered points for numerical solution of PDEs is reported. The method is truly meshless since there is not any connectivity required between the data points. The sets of nodal constants arising from an integration process are represented by indirect RBFNs using relatively small numbers of centres, thereby keeping the size of the network design matrices small. Like the case of FDM and the differential quadrature method (DQM), a collocation mechanism is employed to discretize the governing equations and boundary conditions. The variables and their derivatives are all expressed in terms of nodal variable values. However, it is different from the FDM and DQM that in the present method, meshless global indirect RBFNs are utilized to represent the solution and can be extended to handle the problems with complex geometries in a straightforward manner. The method produces good results using relatively low collocation densities and requires only a minimal amount of effort to implement.

Acknowledgements The author would like to thank Prof. R. I. Tanner for his helpful comments.

References

1. **Atluri, S.N.; Shen, S.** (2002): *The Meshless Local Petrov-Galerkin Method.* Tech Science Press, Encino.

2. **Belytschko, T.; Lu, Y.Y.; Gu, L.** (1994): Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, vol. 37, pp. 229-256.

3. **Branch, M.A.; Coleman, T.F.; Li, Y.** (1999): A subspace, interior and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM Journal on Scientific Computing*, vol. 21, pp. 1-23.

4. **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. 249-264.

5. **Dissanayake, M.W.M.G.; Phan-Thien, N.** (1994): Neural network based approximations for solving partial differential equations. *Communications in Numerical Methods in Engineering*, vol. 10, pp. 195-201.

6. **Franke, R.** (1982): Scattered data interpolation: tests of some methods. *Mathematics of Computation*, vol. 38(157), pp. 181-200.

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

8. **Jin, X.; Li, G.; Aluru, N.R.** (2001): On the equivalence between least-squares and kernel approximations in meshless methods. *Computer Modeling in Engineering & Sciences*, vol. 2(4), pp. 447-462.

9. **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(8/9), pp. 147-161.

10. **Leonard, B.P.; Drummond, J.E.** (1995): 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*, vol. 20, pp. 421-442.

11. **Li, S.; Liu, W.K.** (2002): Meshfree and particle methods and their applications. *Applied Mechanics Reviews*, vol. 55(1), pp. 1-34.

12. **Li, Q.; Shen, S.; Han, Z.D.; Atluri, S.N.** (2003): Application of meshless local Petrov-Galerkin (MLPG) to problems with singularities, and material discontinu-

ities, in 3-D elasticity. *Computer Modeling in Engineering & Sciences*, vol. 4(5), pp. 571-586.

13. **Liu, G.R.** (2003): *Mesh Free Methods - Moving Beyond The Finite Element Method.* CRC Press, Boca Raton.

14. **Liu, G.R.; Gu, Y.T.** (2001): A local radial point interpolation method (LRPIM) for free vibration analyses of 2-D solids. *Journal of Sound and Vibration*, vol. 246(1), pp. 29-46.

15. **Long, S.; Atluri, S.N.** (2002): A meshless local Petrov-Galerkin method for solving bending problem of a thin plate. *Computer Modeling in Engineering & Sciences*, vol. 3(1), pp. 53-64.

16. **Madych, W.R.; Nelson, S.A.** (1989): Multivariate interpolation and conditionally positive definite functions. *Approximation Theory and its Applications*, vol. 4, pp. 77-89.

17. **Madych, W.R.; Nelson, S.A.** (1990): Multivariate interpolation and conditionally positive definite functions, II. *Mathematics of Computation*, vol. 54(189), pp. 211-230.

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

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

20. **Mai-Duy, N.; Tran-Cong, T.** (2002): Mesh-free radial basis function network methods with domain decomposition for approximation of functions and numerical

solution of Poisson's equations. *Engineering Analysis with Boundary Elements*, vol. 26, pp. 133-156.

21. **Mai-Duy, N.; Tran-Cong, T.** (2003a): Approximation of function and its derivatives using radial basis function network methods. *Applied Mathematical Modelling*, vol. 27, pp. 197-220.

22. **Mai-Duy, N.; Tran-Cong, T.** (2003b): Indirect RBFN method with thin plate splines for numerical solution of differential equations. *Computer Modeling in Engineering & Sciences*, vol. 4(1), pp. 85-102.

23. **More, J.J.; Sorensen, D.C.** (1983): Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, vol. 3, pp. 553-572.

24. **Pozrikidis, C.** (1997): *Introduction to Theoretical and Computational Fluid Dynamics*. Oxford University Press, New York.

25. **Qian, L.F.; Batra, R.C.; Chen, L.M.** (2003): Free and forced vibrations of thick rectangular plates using higher-order shear and normal deformable plate theory and meshless Petrov-Galerkin (MLPG) method. *Computer Modeling in Engineering & Sciences*, vol. 4(5), pp. 519-534.

26. **Raju, I.S.; Phillips, D.R.** (2003): Further developments in the MLPG method for beam problems. *Computer Modeling in Engineering & Sciences*, vol. 4(1), pp. 141-160.

27. **Shu, C.; Ding, H.; 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. 941-954.

28. **Sladek, J.; Sladek, V.; Mang, H.A.** (2002): Meshless formulations for simply supported and clamped plate problems. *International Journal for Numerical*

Methods in Engineering, vol. 55, pp. 359-375.

29. **Sladek, J.; Sladek, V.; Zhang, C.** (2003): Application of meshless local Petrov-Galerkin (MLPG) method to elastodynamic problems in continuously nonhomogeneous solids. *Computer Modeling in Engineering & Sciences*, vol. 4(6), pp. 637-648.

30. **Tsai, C.C.; Young, D.L.; Cheng, A.H.-D.** (2002): Meshless BEM for three-dimensional Stokes flows. *Computer Modeling in Engineering & Sciences*, vol. 3(1), pp. 117-128.

Table 1: Natural convection flow in a square slot: Comparison of the IRBFN results with the benchmark solution (de Vahl Davis, 1983).

Solution	Ra	u_{1max}	(x_2)	u_{2max}	(x_1)	$\bar{Nu}(x_1 = 0)$
Benchmark	$1.0e3$	3.649	(0.813)	3.697	(0.178)	1.117
Computed	$1.0e3$	3.659	(0.814)	3.711	(0.178)	1.132
(690 interior points)						
Benchmark	$1.0e4$	16.178	(0.823)	19.617	(0.119)	2.238
Computed	$1.0e4$	16.190	(0.823)	19.643	(0.118)	2.278
(999 interior points)						
Benchmark	$1.0e5$	34.73	(0.855)	68.59	(0.066)	4.509
Computed	$1.0e5$	34.47	(0.851)	69.05	(0.063)	4.515
(1168 interior points)						
Benchmark	$1.0e6$	64.63	(0.850)	219.36	(0.0379)	8.817
Computed	$1.0e6$	64.39	(0.847)	226.26	(0.0380)	8.660
(1693 interior points)						

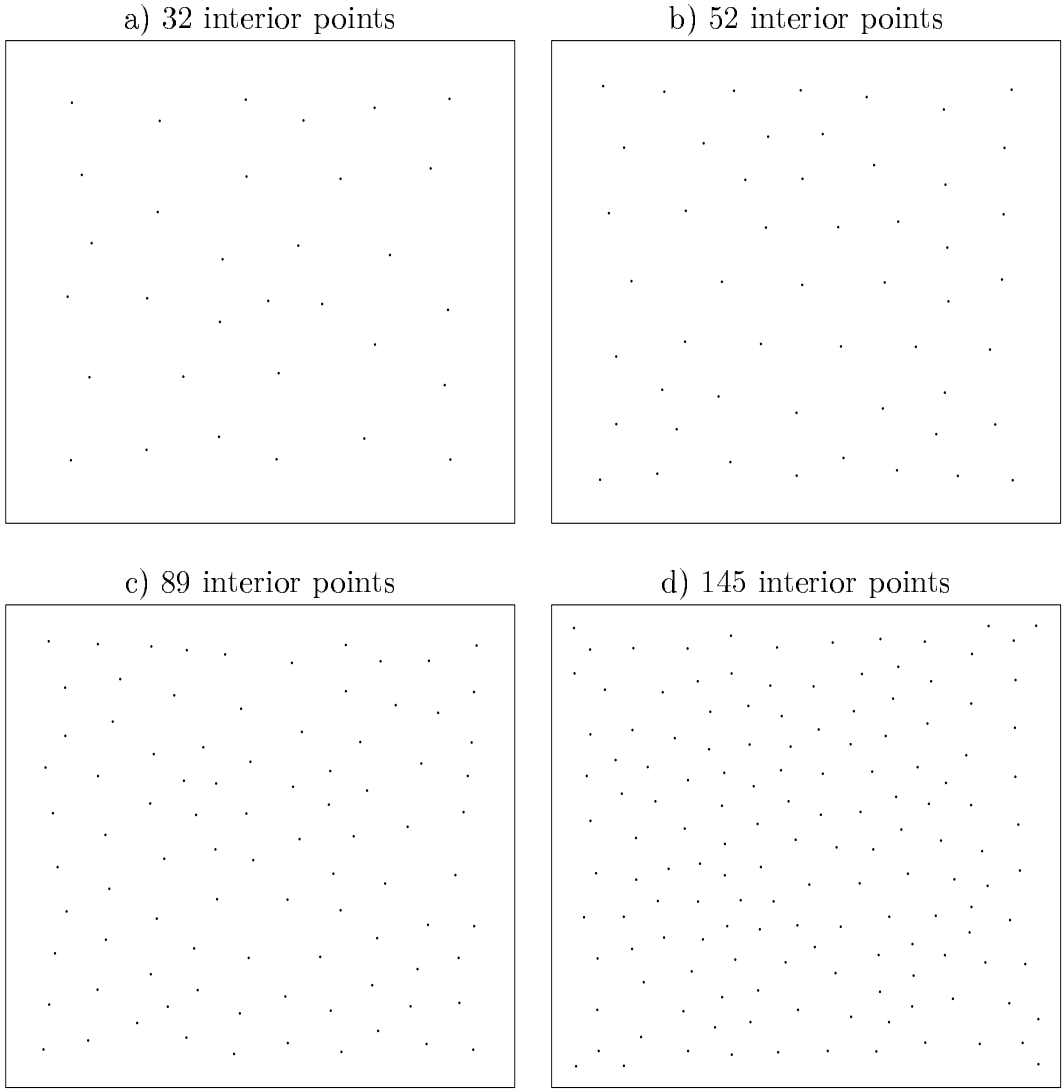

Figure 1: Poisson's equation on a square domain: discretizations by the present IRBFN method. The problem domain is simply represented by a set of scattered points.

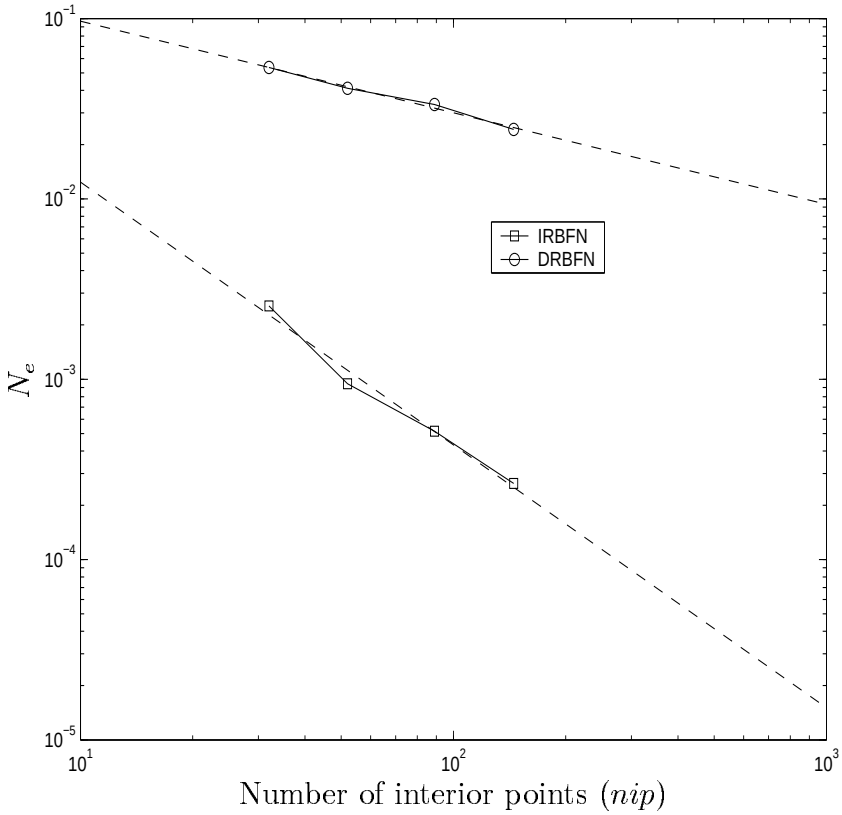

Figure 2: Poisson's equation on a square domain: Comparison of the performance between the IRBFN and DRBFN methods. The former yields more accurate results and higher rate of convergence than the latter. The IRBFN solution converges apparently as $O(nip^{-1.4574})$ while the DRBFN solution apparently as $O(nip^{-0.5081})$.

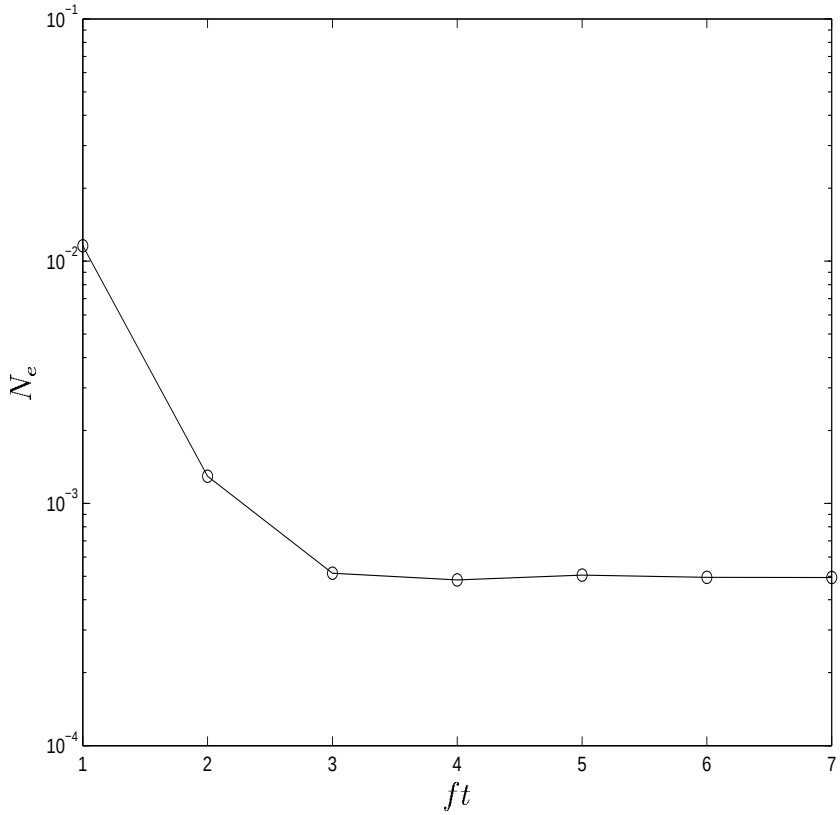

Figure 3: Poisson's equation on a square domain: Effect of ft on the solution accuracy. Increasing the value of ft can improve the solution accuracy. However, when ft is greater than 3, the error norm N_e appears to stay constant. Apparently, at $ft = 3$, the number of neurons (RBFs) used in the approximation of integration constant functions is large enough to capture well the solution. With this value of ft, the "most accurate" solution is achieved, while keeping the network design matrices small.

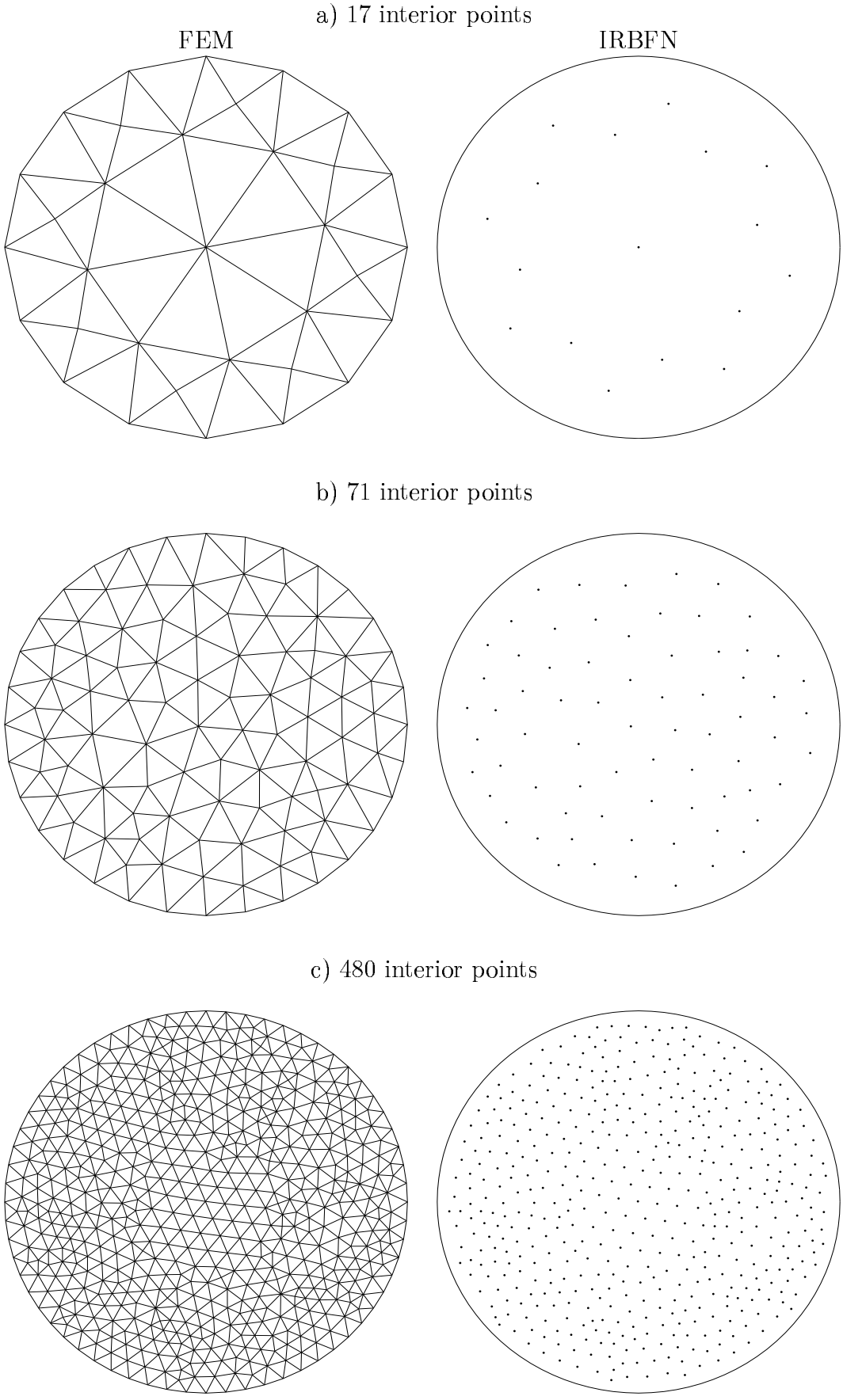

Figure 4: Poisson's equation on a unit disk: discretizations by the FEM and the IRBFN method using the same sets of interior points. The problem domain in the latter is simply represented by a set of scattered points.

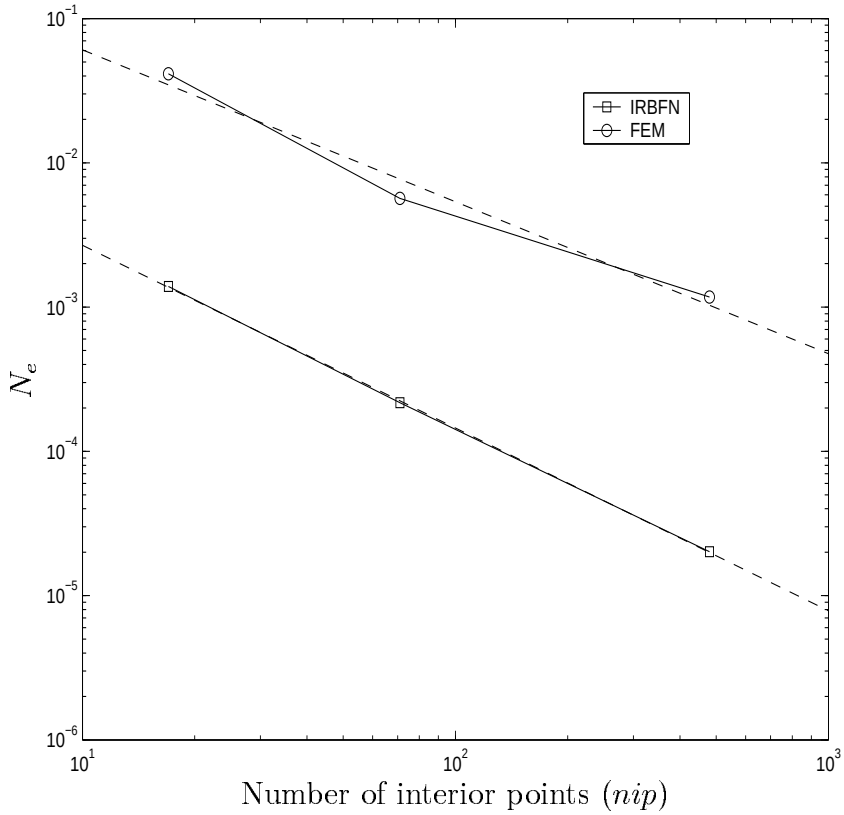

Figure 5: Poisson's equation on a unit disk: Comparison of the performance between the FEM and IRBFN method. The former yields more accurate results and higher rate of convergence than the latter. Solutions converge apparently as $O(nip^{-1.2664})$ and $O(nip^{-1.0529})$ for the IRBFN method and the FEM respectively.

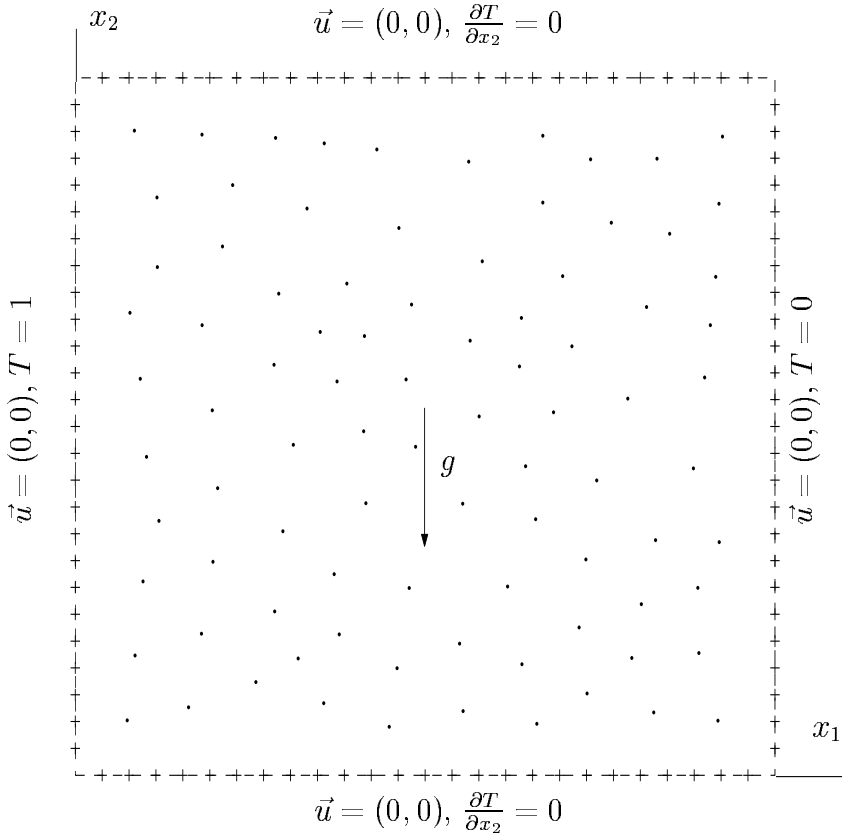

Figure 6: Natural convection flow in a square cavity: geometry definition, boundary conditions and discretization. Legends +: boundary point and .: interior point. The problem domain is simply represented by a set of scattered points.

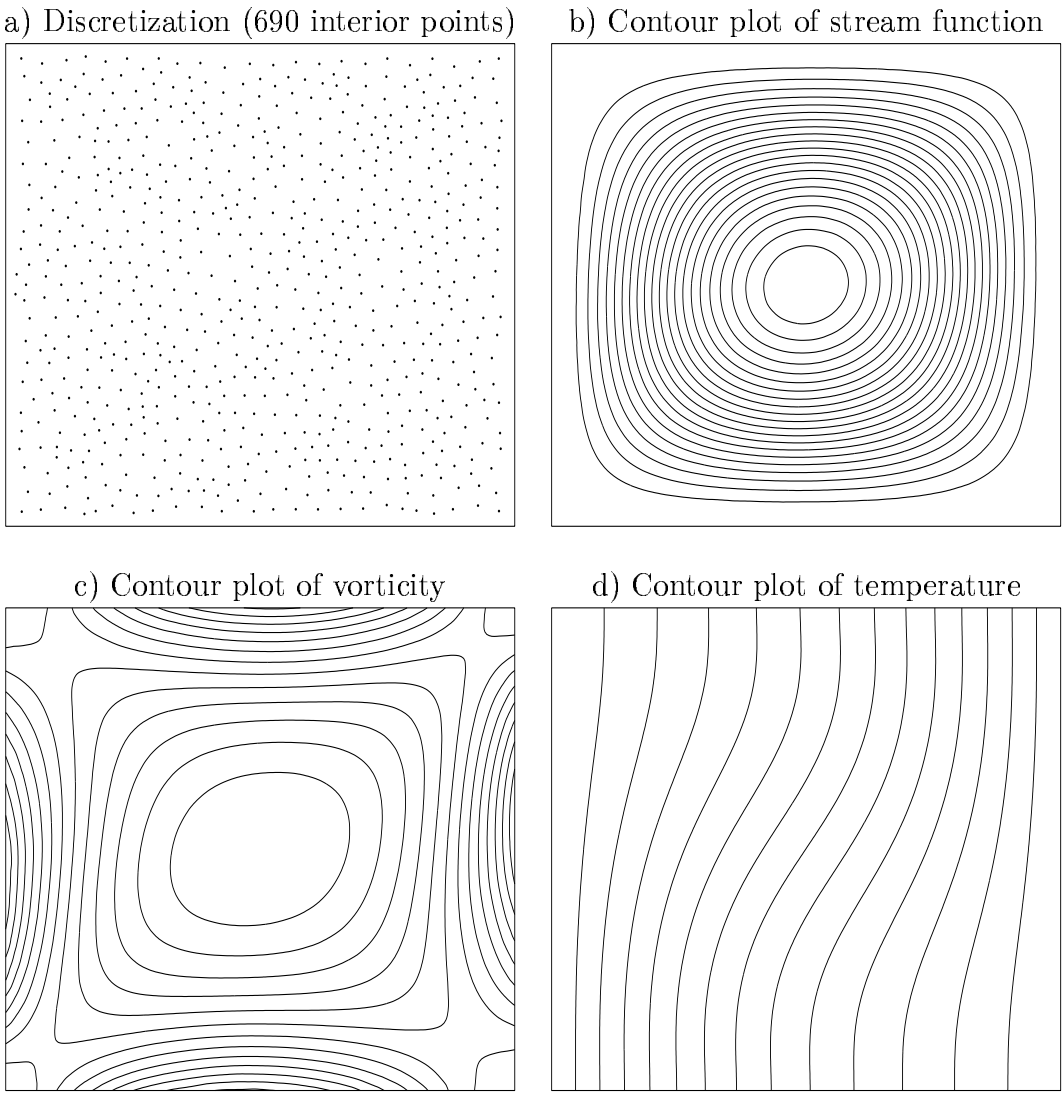

Figure 7: Natural convection flow, 690 interior points, $Ra = 1.e3$: discretization and some contour plots.

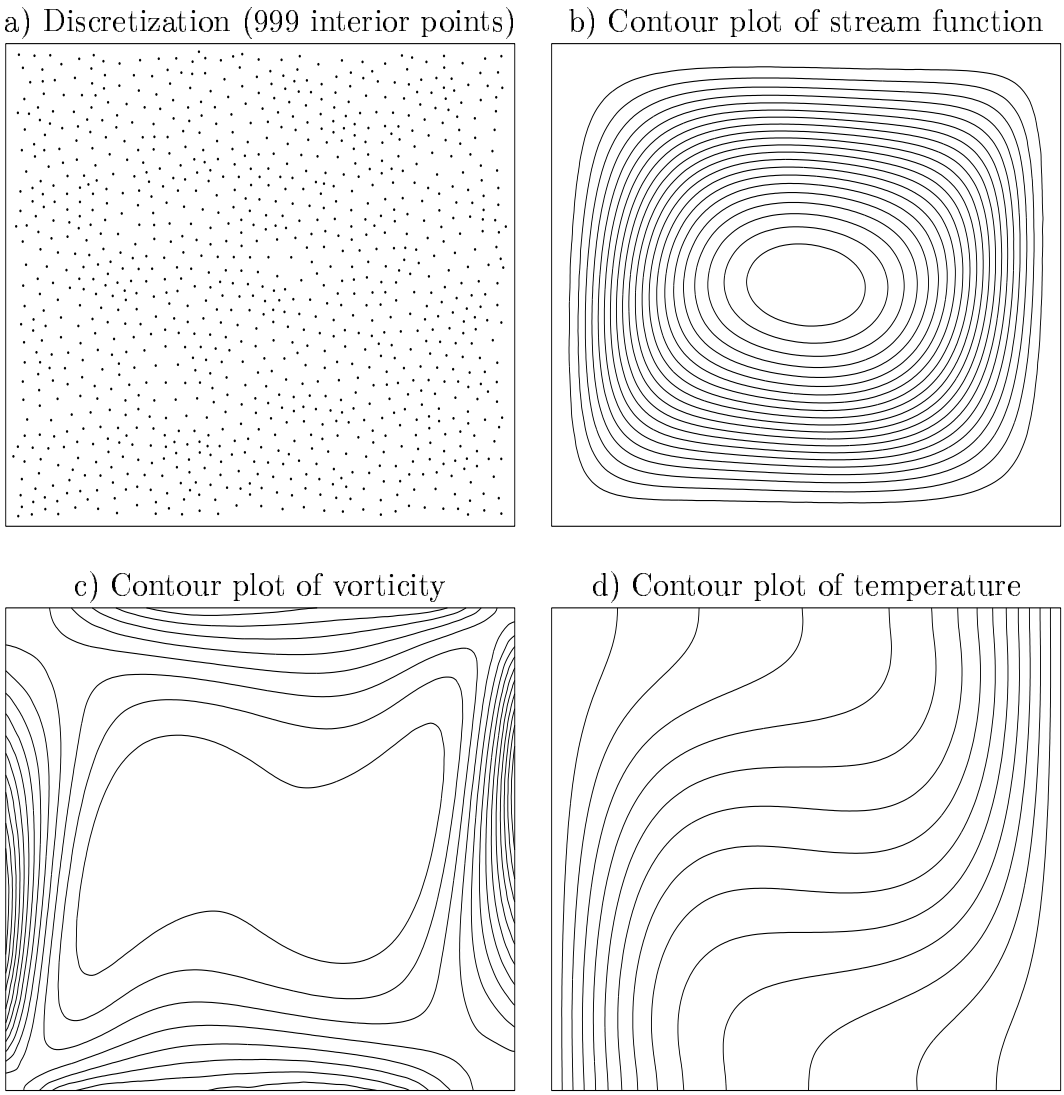

Figure 8: Natural convection flow, 999 interior points, $Ra = 1.e4$: discretization and some contour plots.

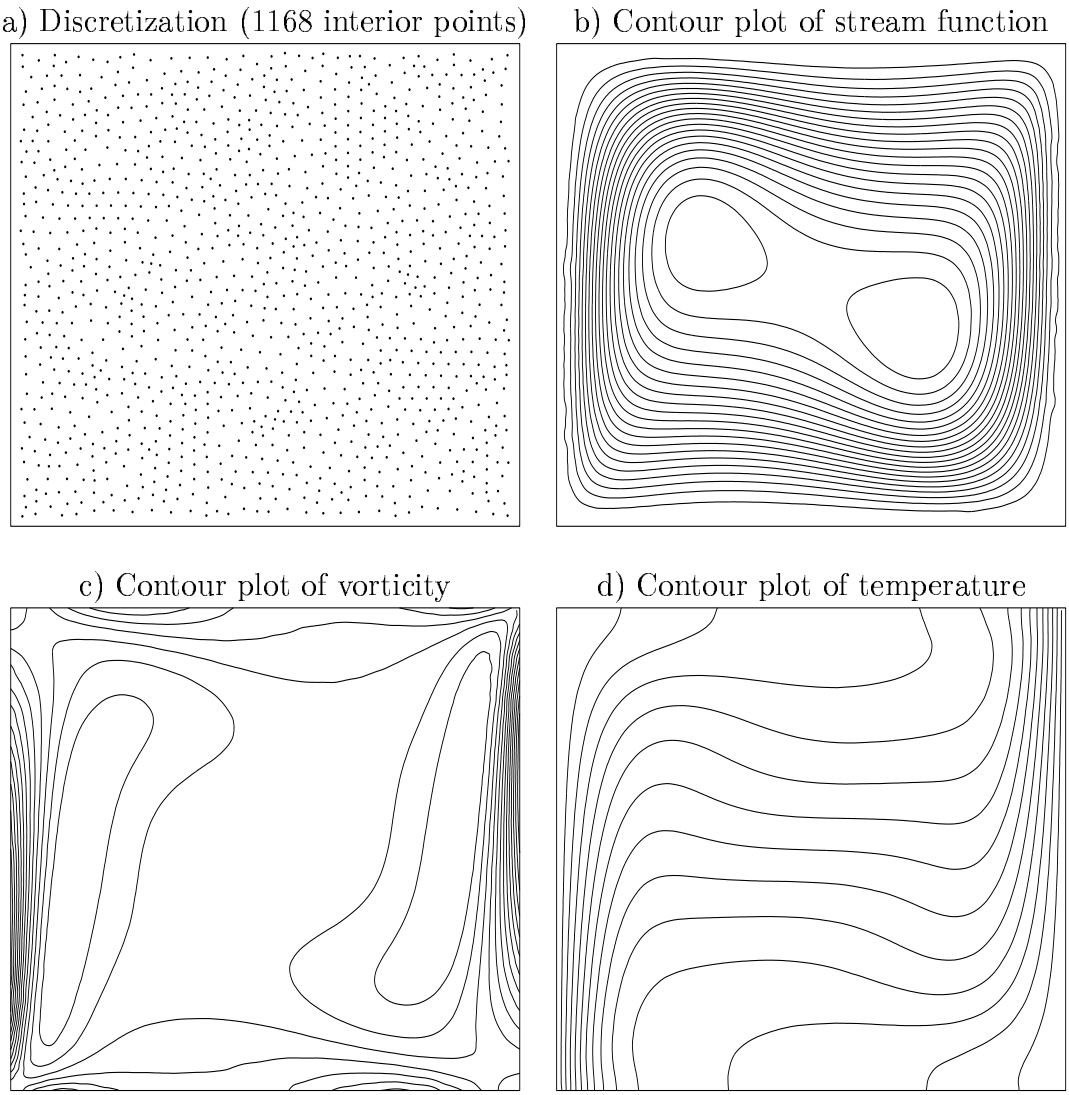

Figure 9: Natural convection flow, 1168 interior points, $Ra = 1.e5$: discretization and some contour plots.

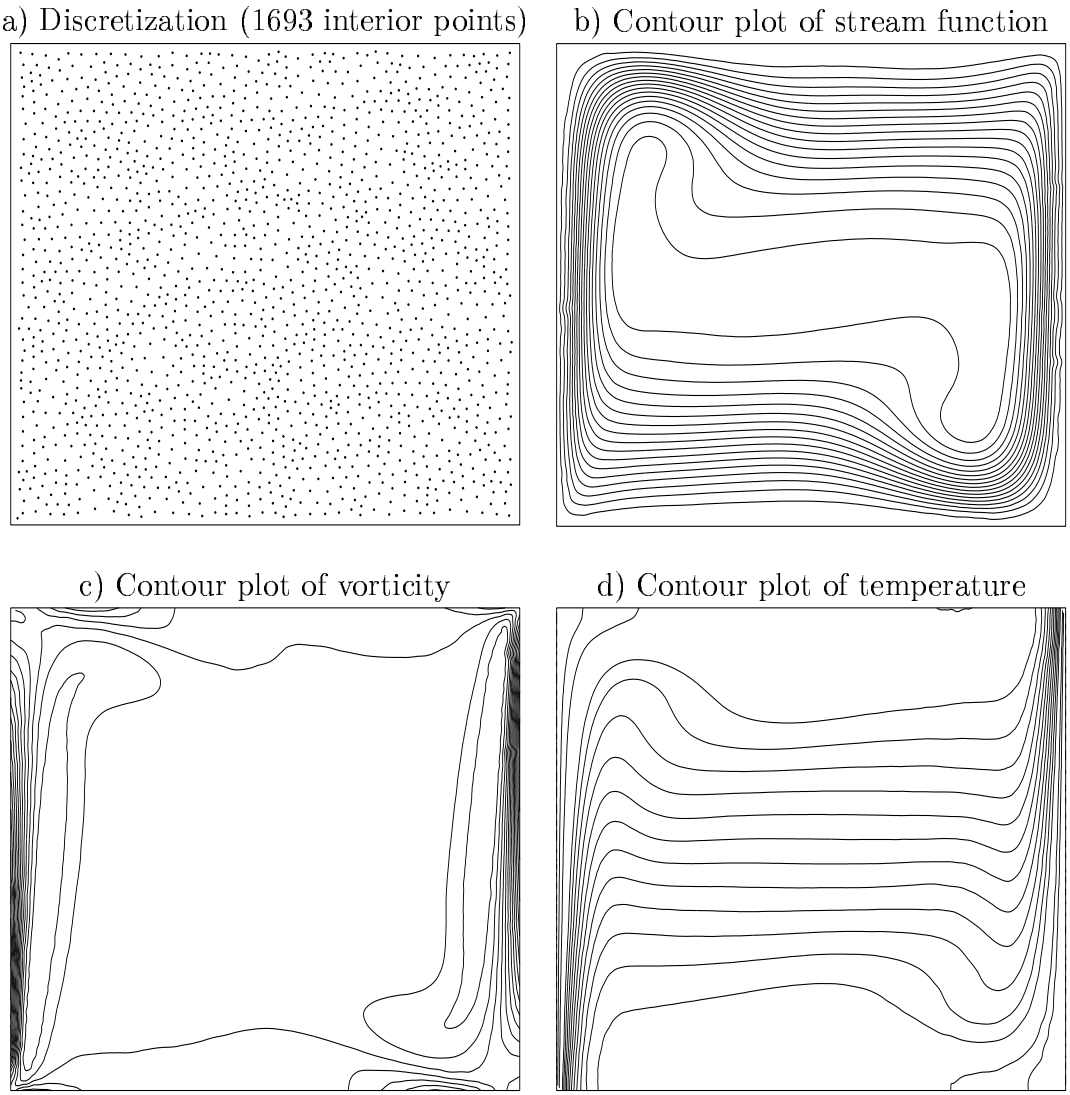

Figure 10: Natural convection flow, 1693 interior points, $Ra = 1.e6$: discretization and some contour plots.