Copyright © 2009 Tech Science Press CMES, vol.51, no.3, pp.213-238, 2009

A Cartesian-Grid Discretisation Scheme Based on Local Integrated RBFNs for Two-Dimensional Elliptic Problems

N. Mai-Duy[1] and T. Tran-Cong[1]

Abstract: This paper reports a new numerical scheme based on Cartesian grids and local integrated radial-basis-function networks (IRBFNs) for the solution of second-order elliptic differential problems defined on two-dimensional regular and irregular domains. At each grid point, only neighbouring nodes are activated to construct the IRBFN approximations. Local IRBFNs are introduced into two different schemes for discretisation of partial differential equations, namely point collocation and control-volume (CV)/subregion-collocation. Linear (e.g. heat flow) and nonlinear (e.g. lid-driven triangular-cavity fluid flow) problems are considered. Numerical results indicate that the local IRBFN CV scheme outperforms the local IRBFN point-collocation scheme regarding accuracy. Moreover, the former shows a similar level of the matrix condition number and a significant improvement in accuracy over a linear CV method.

Keywords: local approximations, integrated RBFNs, point collocation, subregion collocation, second-order differential problems.

1 Introduction

RBF-based discretisation methods have emerged as a new attractive solver for partial differential equations (PDEs) (e.g. [Fasshauer (2007)]). They have the capability to work well for problems defined on irregular domains. Very accurate results can be achieved using only a relatively-small number of nodes. However, RBF matrices are dense and generally ill-conditioned. To resolve this problem, local RBF methods have been developed, resulting in having to solve a sparse system of algebraic equations. The RBF approximations are constructed locally on small overlapping regions which are represented by a set of structured points or a set of scattered points. Works reported include [Lee, Liu, and Fan (2003); Shu, Ding, and Yeo (2003); Tolstykh and Shirobokov (2003); Chantasiriwan (2004); Shu, Ding,

---

[1] Computational Engineering and Science Research Centre, Faculty of Engineering and Surveying, University of Southern Queensland, Toowoomba, QLD 4350, Australia.

and Yeo (2005); Tolstykh and Shirobokov (2005); Wright and Fornberg (2006); Kosec and Sarler (2008a); Kosec and Sarler (2008b); Orsini, Power, and Morvan (2008); Sanyasiraju and Chandhini (2008); Vertnik and Sarler (2009)].

To transform a PDE into a set of algebraic equations, one needs to discretise the problem domain. For irregular domains, this task can be expensive and time-consuming. It can be seen that using Cartesian grids to represent the domain is economical. Considerable effort has been put into the development of Cartesian-grid-based computational techniques (e.g. [Johansen and Colella (1998); Jomaa and Macaskill (2005); Hu, Young, and Fan (2008); Parussini and Pediroda (2008); Pasquim and Mariani (2008); Bourantas, Skouras, and Nikiforidis (2009)]).

The proposed numerical procedure combines strengths of the local RBF approach and the Cartesian-grid approach for solving 2D differential problems. At each grid point, only neighbouring nodes are activated to construct the RBF approximations. Unlike local RBF techniques reported in the literature, RBFNs are employed here to approximate highest-order derivatives in a given PDE and subsequently integrated to obtain expressions for lower-order derivatives and the field variable [Mai-Duy and Tran-Cong (2001)]. This use of integration to construct the approximations provides an effective way to circumvent the problem of reduced convergence rate caused by differentiation and to implement derivative boundary conditions (e.g. [Mai-Duy (2005); Mai-Duy and Tran-Cong (2006)]). In this study, we introduce local integrated RBFNs into two PDE discretisation formulations, namely point collocation and control-volume (CV)/subregion-collocation, and then conduct some numerical experiments to investigate accuracy of the two local IRBFN techniques.

Recirculating viscous flows in enclosed cavities have received a great deal of attention in fluid mechanics community as they can produce interesting flow features at different Reynolds numbers. Examples of this type include lid-driven flows in square and triangular cavities. For such problems, at the two top corners, the velocity is discontinuous and the stress is unbounded. These pose great challenges for numerical simulation. In contrast to the square cavity problem, the triangular cavity flow presents a severe test for structured-grid-based numerical methods (e.g. [Jyotsna and Vanka (1995)]). The latter is chosen here to investigate the performance of the present local IRBFN CV technique. The flow is simulated using the streamfunction and vorticity formulation discretised on rectangular grids. Attractive features of the proposed technique are (i) no coordinate transformations are required and (ii) computational boundary conditions for the vorticity are derived in a new effective way, where analytic formulae that need only nodal values of the streamfuntion on one grid line [Le-Cao, Mai-Duy, and Tran-Cong (2009)] are utilised.

The remainder of this paper is organised as follows. A brief review of integrated

RBFNs is given in Section 2. The proposed computational procedure is presented in Section 3 and numerically verified through a series of examples in Section 4. Section 5 concludes the paper.

2 Integrated radial-basis-function networks

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

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

where the superscript (i) is the summation index, $\mathbf{x}$ 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.

This study is concerned with second-order differential problems in two dimensions. The integral approach [Mai-Duy and Tran-Cong (2001)] uses RBFNs (1) to represent the second-order derivatives of the field variable u in a given PDE. Approximate expressions for the first-order derivatives and the variable itself are then obtained through integration as

$$\begin{aligned}
\frac{\partial^2 u(\mathbf{x})}{\partial x_j^2} &= \sum_{i=1}^{m} w_{[x_j]}^{(i)} G^{(i)}(\mathbf{x}), && (2)\\
\frac{\partial u(\mathbf{x})}{\partial x_j} &= \sum_{i=1}^{m} w_{[x_j]}^{(i)} H_{[x_j]}^{(i)}(\mathbf{x}) + C_{1[x_j]}(x_k), && (3)\\
u_{[x_j]}(\mathbf{x}) &= \sum_{i=1}^{m} w_{[x_j]}^{(i)} \overline{H}_{[x_j]}^{(i)}(\mathbf{x}) + x_j C_{1[x_j]}(x_k) + C_{2[x_j]}(x_k), && (4)
\end{aligned}$$

where the subscript $[x_j]$ is used to denote the quantities associated with the process of integration with respect to the x_j variable; $C_{1[x_j]}(x_k)$ and $C_{2[x_j]}(x_k)$ the constants of integration which are univariate functions of the variable other than x_j (i.e. x_k with $k \neq j$); $H_{[x_j]}^{(i)}(\mathbf{x}) = \int G^{(i)}(\mathbf{x}) dx_j$ and $\overline{H}_{[x_j]}^{(i)}(\mathbf{x}) = \int H_{[x_j]}^{(i)}(\mathbf{x}) dx_j$. The reader is referred to [Mai-Duy and Tran-Cong (2001); Mai-Duy and Tanner (2005); Mai-Duy (2005);Mai-Duy and Tran-Cong (2005)] for further details (e.g. explicit forms of integrated and differentiated RBFs).

3 Proposed technique

The 2D problem domain is discretised using a Cartesian grid. Boundary points are generated through the intersection of the grid lines and the boundaries. For a reference point, we form two local integrated networks using (2)-(4): one associated with the x_1 coordinate and the other with the x_2 coordinate. The two networks are constructed on the same set of $l \times l$ grid lines. The reference point may not be the centre of the local grid when the construction process is carried out near the boundary (Figure 1).

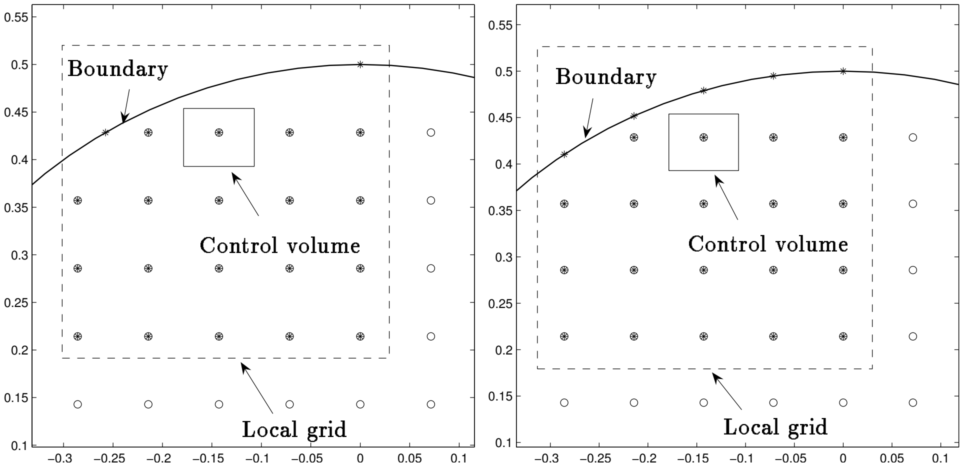


Figure 1: Local networks in x_1 (top) and x_2 (bottom) ($*$: RBF centre and o: interior point).

For local grids entirely embedded in the domain, the two networks have the same set of RBF centres which are chosen to be the interior grid nodes. The value of m in (4) is equal to l^2.

For local grids that are cut by irregular boundary, one generally has different sets of RBF centres for the two associated networks. A set of the RBF centres for the x_j network is comprised of the interior grid nodes and the boundary nodes generated by the x_j grid lines. The value of m in (4) may be less than l^2 (Figure 1).

We also employ IRBFNs to represent the variation of the constants of integration. The construction process for $C_{1[x_j]}(x_k)$ is exactly the same as that for $C_{2[x_j]}(x_k)$. To simplify the notation, the subscripts $_{1[x_j]}$ and $_{2[x_j]}$ are dropped. The function $C(x_k)$

is constructed through

$$\begin{aligned}
\frac{d^2C(x_k)}{dx_k^2} &= \sum_{i=1}^{l} w^{(i)} g^{(i)}(x_k), & (5)\\
\frac{dC(x_k)}{dx_k} &= \sum_{i=1}^{l} w^{(i)} h^{(i)}(x_k) + c_1, & (6)\\
C(x_k) &= \sum_{i=1}^{l} w^{(i)} \bar{h}^{(i)}(x_k) + x_k c_1 + c_2, & (7)
\end{aligned}$$

where c_1 and c_2 are the constants of integration which are simply unknown values, and $g^{(i)}$, $h^{(i)}$ and $\bar{h}^{(i)}$ the one-dimensional forms of $G^{(i)}$, $H^{(i)}$ and $\overline{H}^{(i)}$, respectively. Collocating (7) at the local grid points $x_k^{(i)}$ with $i = \{1, 2, \cdots, l\}$ leads to

$$\widehat{C} = \widehat{\mathscr{T}}\widehat{w}, \tag{8}$$

where $\widehat{C}$ and $\widehat{w}$ are the vectors of length l and $(l+2)$, respectively, and $\widehat{\mathscr{T}}$ is the transformation matrix of dimensions $l \times (l+2)$

$$\widehat{C} = \left(C(x_k^{(1)}), C(x_k^{(2)}), \cdots, C(x_k^{(l)})\right)^T = \left(C^{(1)}, C^{(2)}, \cdots, C^{(l)}\right)^T,$$

$$\widehat{w} = \left(w^{(1)}, w^{(2)}, \cdots, w^{(l)}, c_1, c_2\right)^T,$$

$$\widehat{\mathscr{T}} = \begin{bmatrix}
\bar{h}^{(1)}(x_k^{(1)}), & \bar{h}^{(2)}(x_k^{(1)}), & \cdots, & \bar{h}^{(l)}(x_k^{(1)}), & x_k^{(1)}, & 1\\
\bar{h}^{(1)}(x_k^{(2)}), & \bar{h}^{(2)}(x_k^{(2)}), & \cdots, & \bar{h}^{(l)}(x_k^{(2)}), & x_k^{(2)}, & 1\\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots\\
\bar{h}^{(1)}(x_k^{(l)}), & \bar{h}^{(2)}(x_k^{(l)}), & \cdots, & \bar{h}^{(l)}(x_k^{(l)}), & x_k^{(l)}, & 1
\end{bmatrix}.$$

Taking (8) into account, the value of C in (7) at an arbitrary point x_k can be computed in terms of nodal values of C as

$$C(x_k) = \left[\bar{h}^{(1)}(x_k), \bar{h}^{(2)}(x_k), \cdots, \bar{h}^{(l)}(x_k), x_k, 1\right] \widehat{\mathscr{T}}^{+}\widehat{C}, \tag{9}$$

or

$$C(x_k) = \sum_{i=1}^{l} P^{(i)}(x_k) C^{(i)}, \tag{10}$$

where $P^{(i)}(x_k)$ is the product of the first vector on RHS and the ith column of $\widehat{\mathscr{T}}^{+}$, and $\widehat{\mathscr{T}}^{+}$ is the generalised inverse of $\widehat{\mathscr{T}}$ of dimensions $(l+2) \times l$, which can be obtained using the SVD technique. It is noted that both $P^{(i)}(x_k)$ and $\widehat{\mathscr{T}}^{+}$ could not be defined explicitly.

Substitution of (10) into (3) and (4) yields

$$\frac{\partial u(\mathbf{x})}{\partial x_j} = \sum_{i=1}^{m} w^{(i)}_{[x_j]} H^{(i)}_{[x_j]}(\mathbf{x}) + \sum_{i=1}^{l} P^{(i)}_{[x_j]}(x_k) C^{(i)}_{1[x_j]}, \tag{11}$$

$$u_{[x_j]}(\mathbf{x}) = \sum_{i=1}^{m} w^{(i)}_{[x_j]} \overline{H}^{(i)}_{[x_j]}(\mathbf{x}) + \sum_{i=1}^{l} x_j P^{(i)}_{[x_j]}(x_k) C^{(i)}_{1[x_j]} + \sum_{i=1}^{l} P^{(i)}_{[x_j]}(x_k) C^{(i)}_{2[x_j]}. \tag{12}$$

For convenience of presentation, expressions (2), (11) and (12) can be rewritten as

$$\frac{\partial^2 u(\mathbf{x})}{\partial x_j^2} = \sum_{i=1}^{m+2l} w^{(i)}_{[x_j]} G^{(i)}_{[x_j]}(\mathbf{x}), \tag{13}$$

$$\frac{\partial u(\mathbf{x})}{\partial x_j} = \sum_{i=1}^{m+2l} w^{(i)}_{[x_j]} H^{(i)}_{[x_j]}(\mathbf{x}), \tag{14}$$

$$u_{[x_j]}(\mathbf{x}) = \sum_{i=1}^{m+2l} w^{(i)}_{[x_j]} \overline{H}^{(i)}_{[x_j]}(\mathbf{x}), \tag{15}$$

where

$\{G^{(i)}_{[x_j]}(\mathbf{x})\}_{i=m+1}^{m+2l} \equiv \{0\}_{i=1}^{2l}$,

$\{H^{(i)}_{[x_j]}(\mathbf{x})\}_{i=m+1}^{m+l} \equiv \{P^{(i)}_{[x_j]}(x_k)\}_{i=1}^{l}$, $\{H^{(i)}_{[x_j]}(\mathbf{x})\}_{i=m+l+1}^{m+2l} \equiv \{0\}_{i=1}^{l}$,

$\{\overline{H}^{(i)}_{[x_j]}(\mathbf{x})\}_{i=m+1}^{m+l} \equiv \{x_j P^{(i)}_{[x_j]}(x_k)\}_{i=1}^{l}$, $\{\overline{H}^{(i)}_{[x_j]}(\mathbf{x})\}_{i=m+l+1}^{m+2l} \equiv \{P^{(i)}_{[x_j]}(x_k)\}_{i=1}^{l}$,

$\{w^{(i)}_{[x_j]}\}_{i=m+1}^{m+l} \equiv \{C^{(i)}_{1[x_j]}\}_{i=1}^{l}$, and $\{w^{(i)}_{[x_j]}\}_{i=m+l+1}^{m+2l} \equiv \{C^{(i)}_{2[x_j]}\}_{i=1}^{l}$.

We seek the solution in terms of nodal values of the field variable u. To do so, (15) is collocated at the nodal points on the local grid, from which the relationship between the network-weight space and the physical space can be established as

$$\widetilde{u}_{[x_j]} = \widetilde{\mathscr{T}}_{[x_j]} \widetilde{w}_{[x_j]}, \tag{16}$$

$$\widetilde{w}_{[x_j]} = \widetilde{\mathscr{T}}^{+}_{[x_j]} \widetilde{u}_{[x_j]}, \tag{17}$$

where $\widetilde{u}_{[x_j]}$ is the vector of length m consisting of the nodal values of u on the local grid, $\widetilde{w}_{[x_j]}$ the vector of length $(m+2l)$ made up of the RBF weights and the nodal values of $C^{(i)}_{1[x_j]}$ and $C^{(i)}_{2[x_j]}$, and $\widetilde{\mathscr{T}}^{+}_{[x_j]}$ the generalised inverse of $\widetilde{\mathscr{T}}_{[x_j]}$. The transformation matrix $\widetilde{\mathscr{T}}_{[x_j]}$ has the entries $\widetilde{\mathscr{T}}_{[x_j]rs} = \overline{H}^{(s)}_{[x_j]}(\mathbf{x}^{(r)})$, where $1 \leq r \leq m$ and $1 \leq s \leq (m+2l)$.

It is noted that the two vectors, $\widetilde{u}_{[x_1]}$ and $\widetilde{u}_{[x_2]}$, are unknown. From now on, they are forced to be identical

$$\widetilde{u}_{[x_1]} \equiv \widetilde{u}_{[x_2]} \equiv \widetilde{u}. \tag{18}$$

The values of u, $\partial u/\partial x_j$ and $\partial^2 u/\partial x_j^2$ at an arbitrary point $\mathbf{x}$ can be computed in terms of nodal variable values as

$$u(\mathbf{x}) = \frac{1}{2}\sum_{j=1}^{2} u_{[x_j]}(\mathbf{x}) = \frac{1}{2}\sum_{j=1}^{2}\left(\left[\overline{H}_{[x_j]}^{(1)}(\mathbf{x}), \overline{H}_{[x_j]}^{(2)}(\mathbf{x}), \cdots, \overline{H}_{[x_j]}^{(m+2l)}(\mathbf{x})\right] \widetilde{\mathscr{T}}_{[x_j]}^{+}\right)\widetilde{u}, \tag{19}$$

$$\frac{\partial u(\mathbf{x})}{\partial x_j} = \left[H_{[x_j]}^{(1)}(\mathbf{x}), H_{[x_j]}^{(2)}(\mathbf{x}), \cdots, H_{[x_j]}^{(m+2l)}(\mathbf{x})\right] \widetilde{\mathscr{T}}_{[x_j]}^{+}\widetilde{u}, \tag{20}$$

$$\frac{\partial^2 u(\mathbf{x})}{\partial x_j^2} = \left[G^{(1)}(\mathbf{x}), G^{(2)}(\mathbf{x}), \cdots, G^{(m+2l)}(\mathbf{x})\right] \widetilde{\mathscr{T}}_{[x_j]}^{+}\widetilde{u}. \tag{21}$$

In (19), there are two integrated networks in the x_1 and x_2 directions that produce two values of the function at a point. Theoretically, they are the same. However, due to approximation errors, these two values are not identical. As a result, the function value is computed by taking the average of the two.

For the point-collocation formulation, there are no integrations required for the discretisation. The process of converting the PDE into a set of algebraic equations is straightforward.

For the control-volume formulation, one has to define a control volume for each node, over which the PDE will be integrated. The control volume is formed using the lines that are parallel to the x_1 and x_2 axes and go through the middle points between the reference node and its neighbours or appropriate points on the boundary (Figure 1). Integrals can be calculated using high-order (e.g. 5-points used here) Gauss quadrature since the present approximation scheme allows the accurate evaluation of the variable u and its derivatives at any point within the local grid.

The use of local integrated networks results in a sparse system of simultaneous equations. It can be seen that operations on zero elements are unnecessary. Avoiding these operations provides a considerable saving in time. By taking account of sparseness of the system matrix, one has the capability to reduce the computational time and storage facilities. Such sparse equation sets can be solved effectively by means of iterative solvers (e.g. generalised minimum residual methods).

4 Numerical examples

For all numerical examples presented here, the approximations are constructed on local grids of 5×5. IRBFNs are implemented with the multiquadric (MQ) basis function whose form is given

$$G^{(i)}(\mathbf{x}) = \sqrt{(\mathbf{x}-\mathbf{c}^{(i)})^T(\mathbf{x}-\mathbf{c}^{(i)}) + a^{(i)2}}, \tag{22}$$

where $\mathbf{c}^{(i)}$ and $a^{(i)}$ are the centre and width of the ith MQ basis function, respectively. The set of centres and the set of collocation points are identical. It is well known that the width/shape-parameter strongly affects accuracy of the RBF scheme. However, it is still very difficult to determine the optimal value of the shape parameter in practice. In this study, we do not focus on the study of the RBF width. All MQ centres are associated with the same width that is chosen to be the grid size. For problems whose exact solution is available, we use the discrete relative L_2 norm of u, denoted by $Ne(u)$, to measure accuracy of an approximate scheme. We apply the matrix 1-norm estimation algorithm for estimating condition numbers of the system matrix. Furthermore, linear CV (central difference) techniques, which are similar to those described in [Patankar (1980)], are referred to as a standard CV technique.

4.1 Test problem

Consider the following PDE

$$\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} = 0 \tag{23}$$

with Dirichlet boundary conditions. Two computational domains, namely a unit square $0 \leq x_1, x_2 \leq 1$ and a circle centered at the origin with radius of 0.5, are considered. The exact solution is given by

$$u_e = \frac{1}{\sinh(\pi)} \sin(\pi x_1) \sinh(\pi x_2) \tag{24}$$

from which the boundary values of u can be derived.

The point-collocation formulation consists in forcing (23) to be satisfied exactly at discrete points in order to form a determined set of algebraic equations. It means that (23) needs be collocated at the interior grid nodes.

For the control-volume formulation, (23) is forced to be satisfied in the mean. Integrating (23) over a control volume Ω_i, we have

$$\int_{\Omega_i} \nabla^2 u d\Omega_i = 0. \tag{25}$$

Using the divergence theorem, (25) becomes

$$\int_{\Gamma_i} (\nabla u \cdot \mathbf{n}) d\Gamma_i = 0, \tag{26}$$

where Γ_i is the boundary of Ω_i and $\mathbf{n}$ the outward normal unit vector. To compute $\partial u/\partial x_j$ on the faces that are parallel to the x_k $(k \neq j)$ direction, we use the x_j network.

Uniform Cartesian grids are employed to represent the problem domain.

Table 1: Rectangular domain: Condition numbers of the system matrix by standard CV, local IRBFN collocation and local IRBFN CV methods. Notice that $a(b)$ means $a \times 10^b$.

Grid	Standard-CV	IRBFN-collocation	IRBFN-CV
15×15	1.1(2)	1.8(2)	1.2(2)
27×27	3.9(2)	6.4(2)	4.3(2)
39×39	8.5(2)	1.3(3)	9.3(2)
51×51	1.4(3)	2.4(3)	1.6(3)
63×63	2.2(3)	3.7(3)	2.4(3)
75×75	3.2(3)	5.2(3)	3.5(3)
87×87	4.3(3)	7.1(3)	4.7(3)
99×99	5.6(3)	9.2(3)	6.2(3)
111×111	7.1(3)	1.1(4)	7.8(3)
123×123	8.7(3)	1.4(4)	9.6(3)
135×135	1.0(4)	1.7(4)	1.1(4)
147×147	1.2(4)	2.0(4)	1.3(4)
159×159	1.4(4)	2.4(4)	1.6(4)
171×171	1.7(4)	2.7(4)	1.8(4)
183×183	1.9(4)	3.1(4)	2.1(4)
195×195	2.2(4)	3.6(4)	2.4(4)
207×207	2.5(4)	4.0(4)	2.7(4)

In the case of rectangular domain, condition numbers of the system matrix by the present local collocation and CV techniques are presented in Table 1. Results obtained by the standard CV method are also included for comparison purposes. It can be seen that the three methods yield a similar level of the matrix condition number. The use of local approximations leads to a significant improvement in stability over that of global approximations. It was reported in the literature that the global RBF matrices may be ill-conditioned when using 1000 nodes. Here, with 42849 nodes taken, condition numbers of the RBF matrix are only $O(10^4)$. In terms of accuracy, both RBF methods are more accurate than the standard CV method as

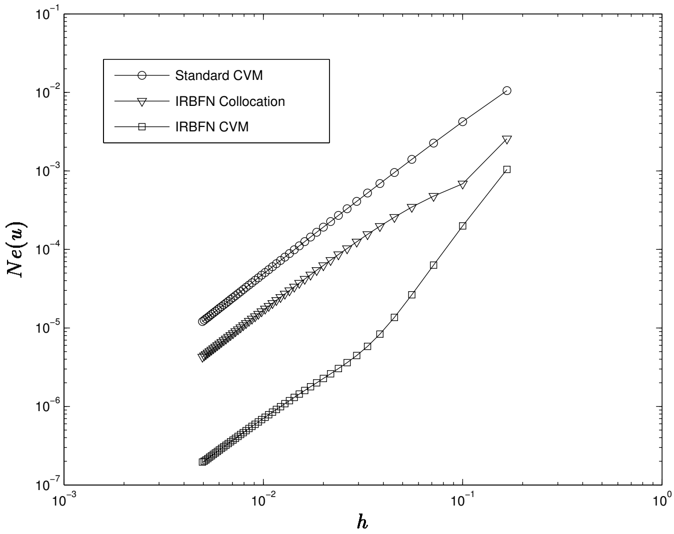


Figure 2: Rectangular domain, $[7 \times 7, 11 \times 11, \cdots, 203 \times 203]$: Error versus grid size for standard CVM/FDM and local IRBFN methods.

shown in Figure 2. The IRBFN-CV method outperforms the IRBFN-collocation method. Given a grid size, the CPU time for the IRBFN-CVM solution is seen to be greater than that for the standard-CVM solution. However, from Figure 2, the IRBFN-CVM is much more accurate than the standard CVM. To achieve a similar level of accuracy, it is necessary to use denser grids for the standard CVM. For example, to yield $Ne = 1.9 \times 10^{-7}$, one needs to employ approximately a grid of 1701×1701 for the standard CVM (this grid density is estimated through extrapolation) and only 203×203 for the IRBFN CVM. It is noted that very high grid densities lead to ill-conditioned matrices. For a given accuracy, the IRBFN CVM can thus be more efficient than the standard CVM. Figure 3 shows the locations of nonzero entries in the IRBFN system matrix.

In the case of circular domain, we generate boundary nodes through the intersection of the grid lines and the boundary. It can be seen that there may be some interior grid nodes that are very close to the boundary. A parameter $\Delta = h/8$ is introduced here. Interior nodes, which fall within a small distance Δ to the boundary, are set aside. The matrix condition number and the accuracy of the three methods are shown in Table 2 and Figure 4. Remarks for this case are similar to those for the rectangular case.

Theoretical studies [Sarra (2006)] showed that IRBFNs have higher approximation

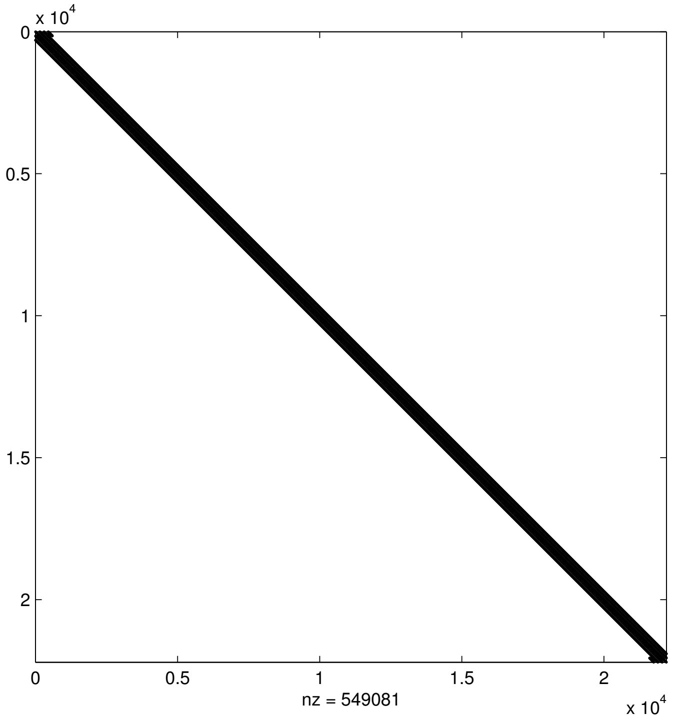


Figure 3: Rectangular domain, 151 × 151: The structure of the 22201 × 22201 IRBFN system matrix.

power than differentiated RBFNs. The implementation of local- and global-IRBFN versions incorporating Cartesian grids shares many common features. The latter was presented in detail in our previous works [Mai-Duy and Tran-Cong (2009a); Mai-Duy and Tran-Cong (2009b)]. For the handling of a Neumann boundary condition on a curved boundary, the reader is referred to [Mai-Duy and Tran-Cong (2009b)] for a full detail. Generally speaking, global versions are capable of giving more accurate results than local versions. However, global schemes produce fully-populated matrices that may limit the number of nodes to a few hundreds only. For problems, which require a relatively-dense discretisation for an accurate simulation, the use of local schemes is a preferred option.

Numerical experiments studied here indicate that the control volume formulation works better for local IRBFNs than the collocation formulation. The IRBFN-CV method is now applied to simulate some heat transfer and fluid flow problems.

Table 2: Circular domain: Condition numbers of the system matrix by standard CV, local IRBFN collocation and local IRBFN CV methods. Notice that $a(b)$ means $a \times 10^b$.

Grid	Standard-CV	IRBFN-collocation	IRBFN-CV
15×15	9.8(1)	2.1(2)	1.0(2)
27×27	3.3(2)	1.0(3)	3.7(2)
39×39	8.5(2)	3.2(3)	8.6(2)
51×51	1.3(3)	4.7(3)	1.4(3)
63×63	2.2(3)	8.1(3)	2.4(3)
75×75	2.8(3)	8.3(3)	3.2(3)
87×87	3.8(3)	1.1(4)	4.4(3)
99×99	5.6(3)	1.8(4)	7.0(3)
111×111	6.5(3)	1.9(4)	7.4(3)
123×123	8.8(3)	2.7(4)	1.1(4)
135×135	1.0(4)	3.5(4)	1.1(4)
147×147	1.3(4)	4.6(4)	1.6(4)
159×159	1.5(4)	5.8(4)	1.7(4)
171×171	1.8(4)	6.5(4)	2.4(4)
183×183	2.0(4)	7.5(4)	2.2(4)
195×195	2.2(4)	6.6(4)	2.3(4)
207×207	2.7(4)	8.5(4)	3.2(4)

4.2 Heat flow

Find the temperature θ such that

$$\nabla.\left(\mathbf{v}\theta - \frac{1}{Pe}\nabla\theta\right) = 0, \quad \mathbf{x} \in \Omega, \tag{27}$$

where $\mathbf{v}$ is a prescribed velocity, Ω the domain and Pe the Peclet number. Here, Ω and $\mathbf{v}$ are taken as $[0,1] \times [-0.5,0.5]$ and $(1,0)^T$, respectively. Boundary conditions are prescribed as follows

$$\theta = 0, \quad \text{for} \quad x_2 = -0.5 \text{ and } x_2 = 0.5, \tag{28}$$

$$\theta = \cos(\pi x_2) \quad \text{for} \quad x_1 = 0, \text{ and} \tag{29}$$

$$\theta = 0 \quad \text{for} \quad x_1 = 1. \tag{30}$$

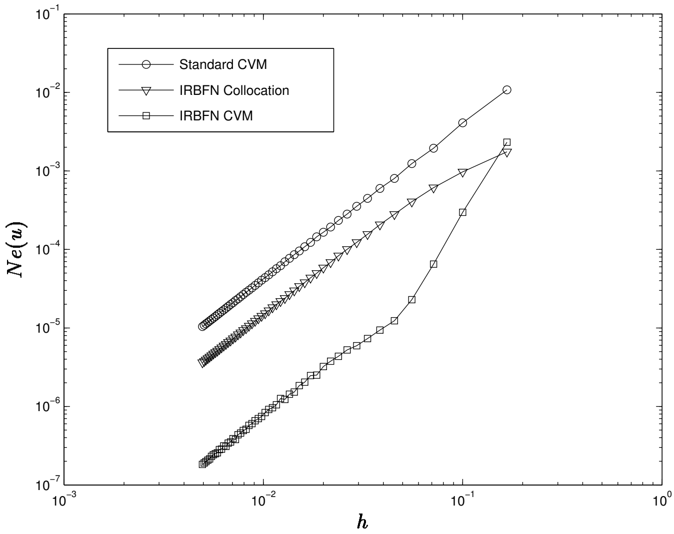


Figure 4: Circular domain, $[7\times 7, 11\times 11, \cdots, 203\times 203]$: Error versus grid size for standard CVM/FDM and local IRBFN methods.

The exact solution to this problem can be verified to be

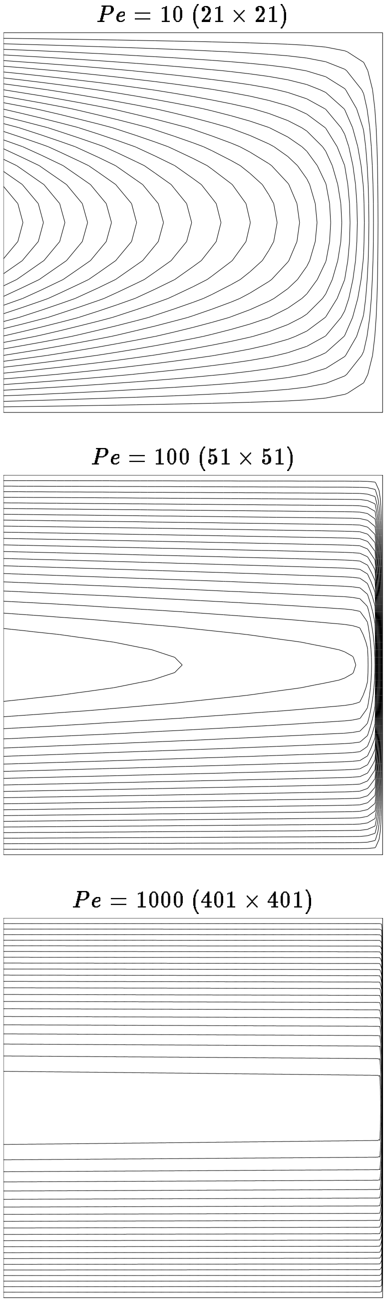
$$\theta_e = \frac{\cos(\pi x_2)}{\exp(a)-\exp(b)}\left(\exp(a+bx_1)-\exp(b+ax_1)\right), \tag{31}$$

where $a = 0.5\left(Pe+\sqrt{Pe^2+4\pi^2}\right)$ and $b = 0.5\left(Pe-\sqrt{Pe^2+4\pi^2}\right)$. This problem is taken from [Kohno and Bathe (2006)].

The temperature boundary layer becomes thinner with increasing Pe. At $Pe = 1000$, very steep boundary layer is formed. Figure 5 shows the temperature contours for three different values of Pe by the present CV method. Its accuracy is better than that of the standard CV method as shown in Table 3. Figure 6 displays variations of temperature along the centre line. It can be seen that the proposed method produces very accurate results for all cases. Figure 7 show that there are no fluctuations in the IRBFN CVM solution.

4.3 Lid-driven triangular-cavity flow

Consider the steady recirculating flow of a Newtonian fluid in an equilateral triangular cavity. Figure 8 shows the cavity geometry. We take $P = \sqrt{3}$ and $Q = 3$. The lid moves from left to right with a unit velocity ($\mathbf{v} = (1,0)^T$), while the left



$Pe = 10\ (21 \times 21)$

$Pe = 100\ (51 \times 51)$

$Pe = 1000\ (401 \times 401)$

Figure 5: Heat flow: Temperature distribution for a wide range of *Pe* by the local IRBFN-CV method. There are 21 contour lines whose values vary linearly between the two extremes.

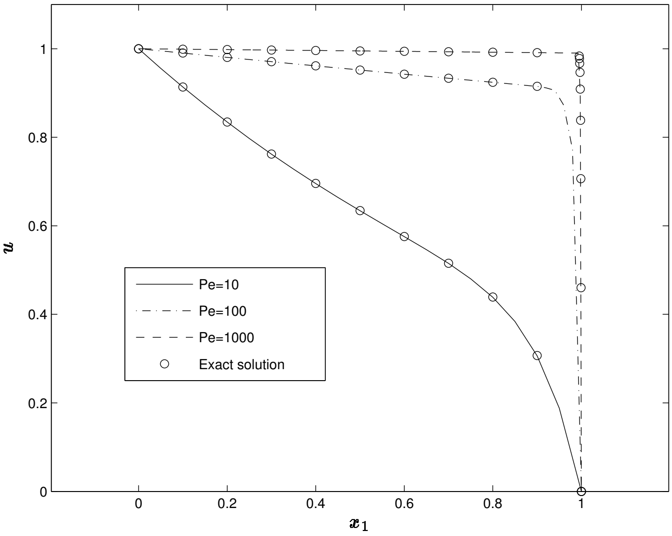


Figure 6: Heat flow: temperature variations on the horizontal centreline for different values of *Pe*. Exact values are also included for comparison purposes.

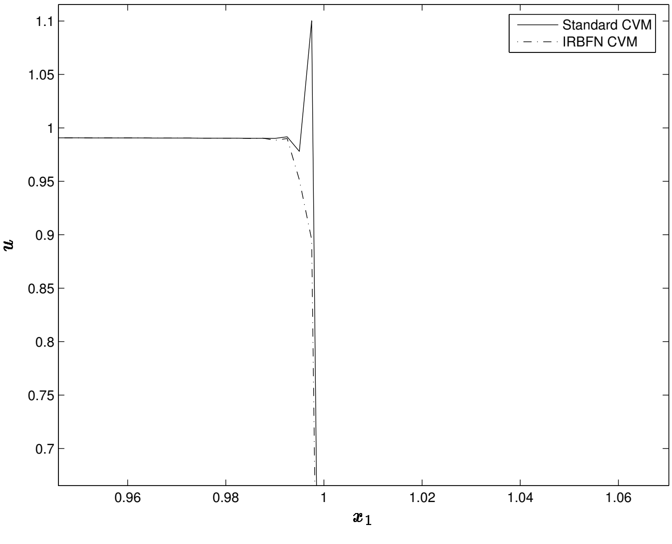


Figure 7: Heat flow: variations of temperature on the centreline in the boundary layer by the two techniques for $Pe = 1000$ using the same grid.

Table 3: Heat Flow, $Pe = 1000$: Error $Ne(u)$ by standard CV and local IRBFN CV methods. Notice that $a(-b)$ means $a \times 10^{-b}$.

Grid	Standard-CV	IRBFN-CV
11×11	2.69(-1)	1.00(-1)
51×51	1.83(-2)	3.69(-3)
101×101	4.25(-3)	9.36(-4)
151×151	1.83(-3)	3.47(-4)
201×201	1.01(-3)	1.75(-4)
251×251	6.45(-4)	1.11(-4)
301×301	4.46(-4)	8.32(-5)
351×351	3.27(-4)	6.92(-5)
401×401	2.50(-4)	6.15(-5)

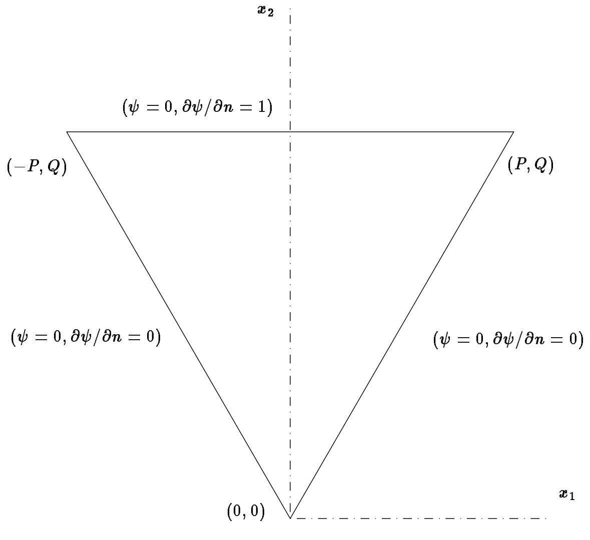


Figure 8: Cavity flow: geometry and boundary conditions.

and right walls are stationary ($\mathbf{v} = (0,0)^T$). The problem was numerically studied by different techniques, including the finite-different method with a transformed geometry (e.g. [Ribbens, Watson, and Wang (1994)] and the finite-element method with a flow-condition-based interpolation (e.g. [Kohno and Bathe (2006)]).

Governing equations:

The governing equations are obtained from the streamfunction-vorticity formulation

$$\omega = \nabla^2 \psi, \tag{32}$$

$$\frac{\partial \omega}{\partial t} + \left(v_1 \frac{\partial \omega}{\partial x_1} + v_2 \frac{\partial \omega}{\partial x_2} \right) = \frac{1}{Re} \nabla^2 \omega, \tag{33}$$

where Re is the Reynolds number defined as $Re = UL/\nu$ (L: the characteristic length, U: the characteristic velocity and ν the kinematic viscosity), ψ the streamfunction, ω the vorticity, $v_1 = \partial \psi / \partial x_2$ and $v_2 = -\partial \psi / \partial x_1$. The reference length and velocity are presently chosen as $L = Q/3$ and $U = 1$ (the velocity of the lid), respectively.

Spatial discretisation:

The equilateral triangular cavity is discretised using a Cartesian grid. Grid nodes inside the cavity are taken to be the interior grid points. For this particular geometry, boundary nodes are generated using only grid lines that contain at least one interior grid node. With this approach, the set of RBF centres/collocation-points do not include the two top corners and hence infinite values of the vorticity do not enter the discrete system.

Boundary conditions:

Boundary conditions for the velocity lead to $\psi = 0$ and $\partial \psi / \partial x_2 = 1$ on the lid, and $\psi = 0$ and $\partial \psi / \partial n = 0$ on the remaining walls (n: the direction normal to the wall).

We will use the boundary values $\psi = 0$ for solving the streamfunction equation. From $\psi = 0$ and $\partial \psi / \partial n = 0$, one obtains $\partial \psi / \partial x_1 = 0$ and $\partial \psi / \partial x_2 = 0$ along the boundaries, which are used for solving the vorticity equation. A vorticity boundary condition is computed through

$$\omega_b = \frac{\partial^2 \psi_b}{\partial x_1^2} + \frac{\partial^2 \psi_b}{\partial x_2^2}, \tag{34}$$

where the subscript b is used to indicate the boundary value. On the top wall, (34) reduces to

$$\omega_b = \frac{\partial^2 \psi_b}{\partial x_2^2}. \tag{35}$$

On the side walls, there are two possible cases

(i) A boundary point is also a grid node and the number of nodal points on the two associated grid lines are sufficiently-large for an accurate approximation, and

(ii) A boundary point is not a grid node or the number of nodal points on one of the two associated grid lines is too small.

To compute ω_b, one can use (34) directly for the first case but may need to derive a suitable formula from (34), which requires information about ψ on one grid line only, for the second case. The latter will be handled here by applying formulae reported in our previous work (e.g. [Le-Cao, Mai-Duy, and Tran-Cong (2009)]), namely

$$\omega_b = \left[1 + \left(\frac{t_1}{t_2}\right)\right] \frac{\partial^2 \psi_b}{\partial x_1^2}, \tag{36}$$

for a x_1-grid line, and

$$\omega_b = \left[1 + \left(\frac{t_2}{t_1}\right)\right] \frac{\partial^2 \psi_b}{\partial x_2^2}, \tag{37}$$

for a x_2-grid line. In (36) and (37), t_1 and t_2 are the x_1- and x_2-components of the unit vector tangential to the boundary.

As mentioned earlier, one has to incorporate $\partial \psi_b / \partial x_1 = 0$ and $\partial \psi_b / \partial x_2 = 0$ into (35), (34), (36) and (37). The incorporation process is similar to that in [Ho-Minh, Mai-Duy, and Tran-Cong (2009)]. It will be briefly reproduced here for the sake of completeness. Consider a x_j grid line. On the line, one has

$$\begin{aligned}
\frac{\partial^2 \psi(x)}{\partial x_j^2} &= \sum_{i=1}^{n_j} w^{(i)} g^{(i)}(x_j) + 0c_1 + 0c_2, && (38)\\
\frac{\partial \psi(x_j)}{\partial x_j} &= \sum_{i=1}^{n_j} w^{(i)} h^{(i)}(x_j) + c_1 + 0c_2, && (39)\\
\psi(x_j) &= \sum_{i=1}^{n_j} w^{(i)} \bar{h}^{(i)}(x_j) + x_j c_1 + c_2, && (40)
\end{aligned}$$

where n_j is the number of nodal points on the line and other notations are defined as before. There are two extra coefficients c_1 and c_2 in (40). As a result, two extra equations representing boundary derivative values $\partial \psi(x_j^{(1)}) / \partial x_j$ and $\partial \psi(x_j^{(n_j)}) / \partial x_j$ can be added to the transformation system

$$\begin{pmatrix} \widehat{\psi} \\ \frac{\partial \psi(x_j^{(1)})}{\partial x_j} \\ \frac{\partial \psi(x_j^{(n_j)})}{\partial x_j} \end{pmatrix} = \widehat{\mathscr{T}} \widehat{w} \tag{41}$$

where $\widehat{\psi}$ is the vector of nodal variable values of length n_j, $\widehat{w}$ the coefficient vector of length (n_j+2) and $\widehat{\mathscr{T}}$ is the transformation matrix of dimensions $(n_j+2)\times(n_j+2)$

$$\widehat{\psi}=\left(\psi(x_j^{(1)}),\psi(x_j^{(2)}),\cdots,\psi(x_j^{(n_j)})\right)^T, \tag{42}$$

$$\widehat{w}=\left(w^{(1)},w^{(2)},\cdots,w^{(n_j)},c_1,c_2\right)^T, \tag{43}$$

$$\widehat{\mathscr{T}}=\begin{bmatrix} \bar{h}^{(1)}(x_j^{(1)}), & \bar{h}^{(2)}(x_j^{(1)}), & \cdots, & \bar{h}^{(n_j)}(x_j^{(1)}), & x_j^{(1)}, & 1 \\ \bar{h}^{(1)}(x_j^{(2)}), & \bar{h}^{(2)}(x_j^{(2)}), & \cdots, & \bar{h}^{(n_j)}(x_j^{(2)}), & x_j^{(2)}, & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \bar{h}^{(1)}(x_j^{(n_j)}), & \bar{h}^{(2)}(x_j^{(n_j)}), & \cdots, & \bar{h}^{(n_j)}(x_j^{(n_j)}), & x_j^{(n_j)}, & 1 \\ h^{(1)}(x_j^{(1)}), & h^{(2)}(x_j^{(1)}), & \cdots, & h^{(n_j)}(x_j^{(1)}), & 1, & 0 \\ h^{(1)}(x_j^{(n_j)}), & h^{(2)}(x_j^{(n_j)}), & \cdots, & h^{(n_j)}(x_j^{(n_j)}), & 1, & 0 \end{bmatrix}.$$

The values of $\partial^2\psi/\partial x_j^2$ at the two boundary points can be computed by

$$\begin{pmatrix} \dfrac{\partial^2\psi(x_j^{(1)})}{\partial x_j^2} \\ \dfrac{\partial^2\psi(x_j^{(n_j)})}{\partial x_j^2} \end{pmatrix}= \tag{44}$$

$$\begin{bmatrix} g^{(1)}(x_j^{(1)}), & g^{(2)}(x_j^{(1)}), & \cdots, & g^{(n_j)}(x_j^{(1)}), & 0, & 0 \\ g^{(1)}(x_j^{(n_j)}), & g^{(2)}(x_j^{(n_j)}), & \cdots, & g^{(n_j)}(x_j^{(n_j)}), & 0, & 0 \end{bmatrix}\widehat{\mathscr{T}}^{-1}\begin{pmatrix} \widehat{\psi} \\ \dfrac{\partial\psi(x_j^{(1)})}{\partial x_j} \\ \dfrac{\partial\psi(x_j^{(n_j)})}{\partial x_j} \end{pmatrix}, \tag{45}$$

where $\widehat{\mathscr{T}}^{-1}$ is an inverse of $\widehat{\mathscr{T}}$.

By means of point collocation and integration constants, derivative boundary values are forced to be satisfied exactly. Moreover, all grid points on the associated grid lines are used to compute ω_b. The present boundary schemes thus have a global property.

Four Cartesian grids, namely Grid 1 (2352 interior points), Grid 2 (5402 points), Grid 3 (9702 points) and Grid 4 (15252 points), are employed to study the convergence of the solution. The flow is simulated at the Reynolds number of 0, 100, 200 and 500. A time-marching approach is applied here to solve the present system of non-linear equations. For the vorticity transport equation (33), the diffusive and convective terms are treated implicitly and explicitly, respectively. We choose the initial solution to be the solution at a lower *Re*. For the case of $Re = 0$, the flow starts from rest.

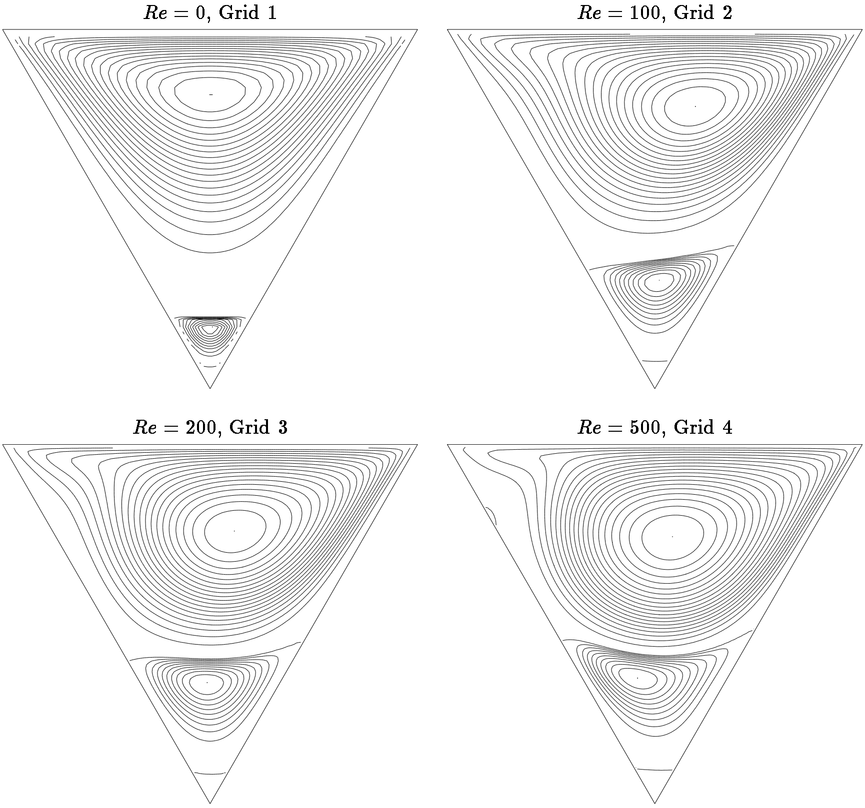


Figure 9: Cavity flow: Streamlines which are drawn using 21 uniform lines between the minimum and zero values, and 11 uniform lines between the zero and maximum values.

Figures 9 and 10 present contour plots of the streamfunction and vorticity variables, which look reasonable when compared with those available in the literature (e.g. [Ribbens, Watson, and Wang (1994); Kohno and Bathe (2006)]).

Figure 11 presents variations of the x_1 component of the velocity vector on the vertical centreline $x_1 = 0$ and the x_2 component of velocity on the horizontal line $x_2 = 2$. Results obtained by [Kohno and Bathe (2006)] are also included for comparison purposes. It can be seen that the present results agree well with those by the flow-conditioned-based interpolation FEM for all values of *Re*.

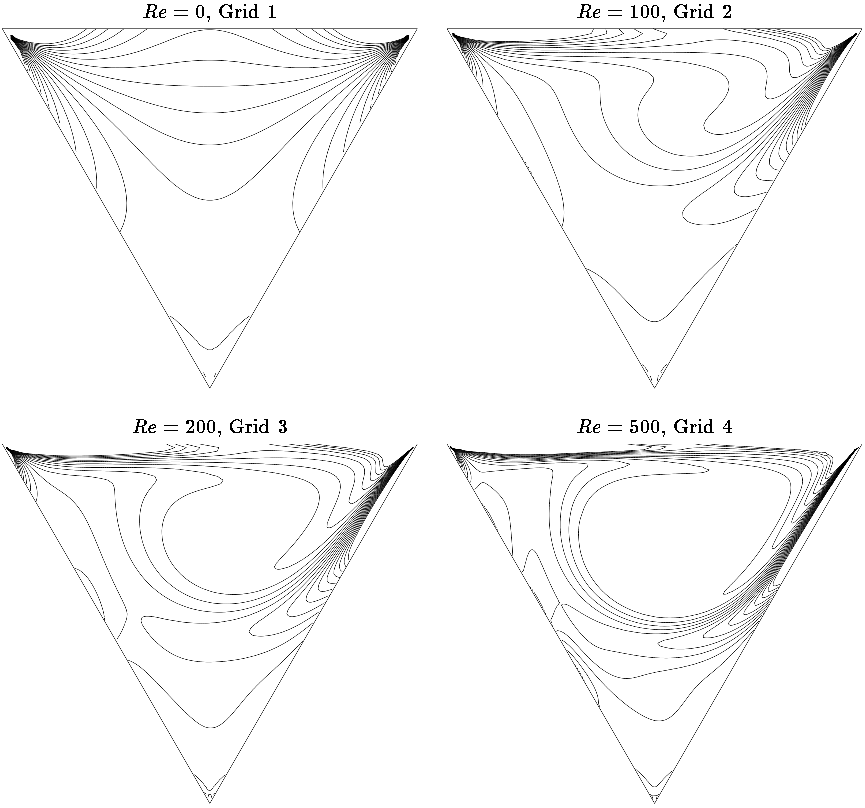


Figure 10: Cavity flow: Iso vorticity lines whose values are the same as those in [Kohno and Bathe (2006)]

5 Concluding remarks

This paper is concerned with the use of local integrated RBFNs and Cartesian grids in the point-collocation and control-volume frameworks. Two main advantages of the present local techniques are that (i) their matrices are sparse and (ii) their preprocessing is simple. Numerical results show that (i) both local IRBFN methods result in the system matrix with a much lower condition number than global RBF techniques, (ii) they outperform standard control-volume techniques regarding accuracy for a given grid size, (iii) the local IRBFN control-volume technique is much more accurate than the local IRBFN collocation technique, (iv) the local IRBFN control-volume technique has the capability to produce accurate results for the simulation of flow problems having steep gradients and complex patterns.

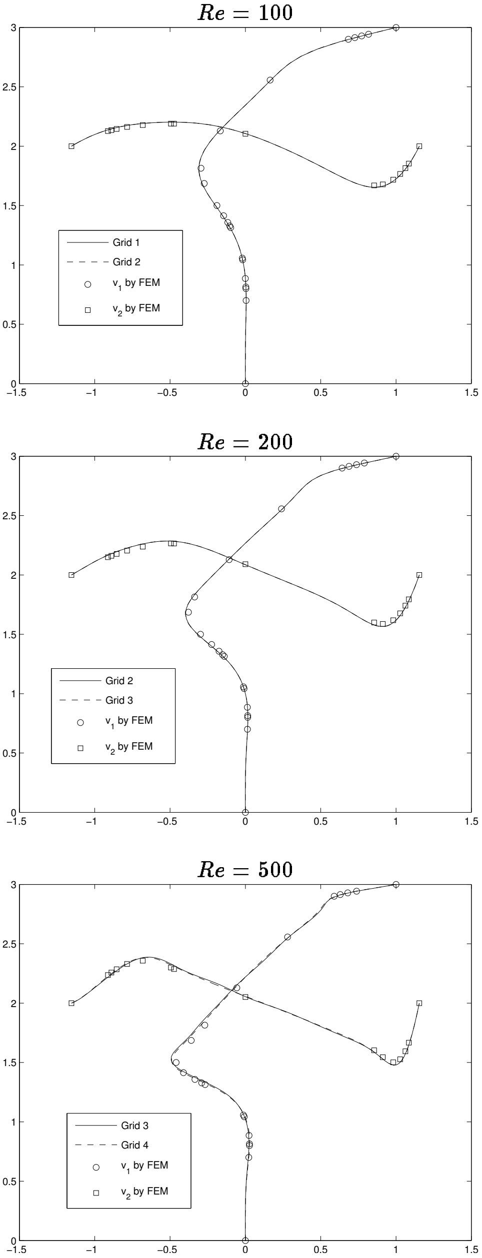


Figure 11: Cavity flow: Vertical and horizontal velocity profiles along the centre line ($x_1 = 0$) and the horizontal line ($x_2 = 2$) for three Reynolds numbers. Results by FEM [Kohno and Bathe (2006)] are also included for comparison purposes. It is noted that the obtained results on the two grids used are indiscernible.

Acknowledgement: This work is supported by the Australian Research Council. We would like to thank the referees for their helpful comments.

References

Bourantas, G. C.; Skouras, E. D.; Nikiforidis, G. C. (2009): Adaptive support domain implementation on the moving least squares approximation for mfree methods applied on elliptic and parabolic pde problems using strong-form description. *CMES: Computer Modeling in Engineering & Sciences*, vol. 43, no. 1, pp. 1–26.

Chantasiriwan, S. (2004): Investigation of the use of radial basis functions in local collocation method for solving diffusion problems. *International Communications in Heat and Mass Transfer*, vol. 31, no. 8, pp. 1095–1104.

Fasshauer, G. E. (2007): *Meshfree Approximation Methods With Matlab (Interdisciplinary Mathematical Sciences - Vol. 6).* World Scientific Publishers.

Ho-Minh, D.; Mai-Duy, N.; Tran-Cong, T. (2009): A Galerkin-RBF approach for the streamfunction-vorticity-temperature formulation of natural convection in 2D enclosured domains. *CMES: Computer Modeling in Engineering & Sciences*, vol. 44, pp. 219–248.

Hu, S. P.; Young, D. L.; Fan, C. M. (2008): FDMFS for diffusion equation with unsteady forcing function. *CMES: Computer Modeling in Engineering & Sciences*, vol. 24, no. 1, pp. 1–20.

Johansen, H.; Colella, P. (1998): A Cartesian grid embedded boundary method for Poisson's equation on irregular domains. *Journal of Computational Physics*, vol. 147, no. 1, pp. 60–85.

Jomaa, Z.; Macaskill, C. (2005): The embedded finite difference method for the Poisson equation in a domain with an irregular boundary and Dirichlet boundary conditions. *Journal of Computational Physics*, vol. 202, no. 2, pp. 488–506.

Jyotsna, R.; Vanka, S. P. (1995): Multigrid calculation of steady, viscous flow in a triangular cavity. *Journal of Computational Physics*, vol. 122, no. 1, pp. 107–117.

Kohno, H.; Bathe, K.-J. (2006): A flow-condition-based interpolation finite element procedure for triangular grids. *International Journal for Numerical Methods in Fluids*, vol. 51, no. 6, pp. 673–699.

Kosec, G.; Sarler, B. (2008a): Solution of thermo-fluid problems by collocation with local pressure correction. *International Journal of Numerical Methods for Heat & Fluid Flow*, vol. 18, pp. 868–882.

Kosec, G.; Sarler, B. (2008b): Local RBF Collocation Method for Darcy Flow. *CMES: Computer Modeling in Engineering & Sciences*, vol. 25, pp. 197–208.

Le-Cao, K.; Mai-Duy, N.; Tran-Cong, T. (2009): An effective integrated-RBFN Cartesian-grid discretisation for the stream function-vorticity-temperature formulation in non-rectangular domains. *Numerical Heat Transfer, Part B*, vol. 55, pp. 480–502.

Lee, C. K.; Liu, X.; Fan, S. C. (2003): Local multiquadric approximation for solving boundary value problems. *Computational Mechanics*, vol. 30, no. 5-6, pp. 396–409.

Mai-Duy, N. (2005): Solving high order ordinary differential equations with radial basis function networks. *International Journal for Numerical Methods in Engineering*, vol. 62, pp. 824–852.

Mai-Duy, N.; Tanner, R. I. (2005): An effective high order interpolation scheme in BIEM for biharmonic boundary value problems. *Engineering Analysis with Boundary Elements*, vol. 29, pp. 210–223.

Mai-Duy, N.; Tran-Cong, T. (2001): Numerical solution of differential equations using multiquadric radial basis function networks. *Neural Networks*, vol. 14, no. 2, pp. 185–199.

Mai-Duy, N.; Tran-Cong, T. (2005): An efficient indirect RBFN-based method for numerical solution of PDEs. *Numerical Methods for Partial Differential Equations*, vol. 21, pp. 770–790.

Mai-Duy, N.; Tran-Cong, T. (2006): Solving biharmonic problems with scattered-point discretisation using indirect radial-basis-function networks. *Engineering Analysis with Boundary Elements*, vol. 30, no. 2, pp. 77–87.

Mai-Duy, N.; Tran-Cong, T. (2009a): A control volume technique based on integrated RBFNs for the convection-diffusion equation. *Numerical Methods for Partial Differential Equations (accepted).*

Mai-Duy, N.; Tran-Cong, T. (2009b): A numerical study of 2D integrated RBFNs incorporating Cartesian grids for solving 2D elliptic differential problems. *Numerical Methods for Partial Differential Equations(accepted).*

Orsini, P.; Power, H.; Morvan, H. (2008): Improving volume element methods by meshless radial basis function techniques. *CMES: Computer Modeling in Engineering & Sciences*, vol. 23, pp. 187–208.

Parussini, L.; Pediroda, V. (2008): Investigation of multi geometric uncertainties by different polynomial chaos methodologies using a fictitious domain solver. *CMES: Computer Modeling in Engineering & Sciences*, vol. 23, no. 1, pp. 29–52.

Pasquim, B. M.; Mariani, V. C. (2008): Solutions for incompressible viscous flow in a triangular cavity using cartesian grid method. *CMES: Computer Modeling in Engineering & Sciences*, vol. 35, no. 2, pp. 113–132.

Patankar, S. V. (1980): *Numerical Heat Transfer and Fluid Flow*. McGraw-Hill.

Ribbens, C. J.; Watson, L. T.; Wang, C.-Y. (1994): Steady viscous flow in a triangular cavity. *Journal of Computational Physics*, vol. 112, no. 1, pp. 173–181.

Sanyasiraju, Y. V. S. S.; Chandhini, G. (2008): Local radial basis function based gridfree scheme for unsteady incompressible viscous flows. *Journal of Computational Physics*, vol. 227, no. 20, pp. 8922–8948.

Sarra, S. A. (2006): Integrated multiquadric radial basis function approximation methods. *Computers & Mathematics with Applications*, vol. 51, no. 8, pp. 1283–1296.

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.

Shu, C.; Ding, H.; Yeo, K. S. (2005): Computation of incompressible navier-stokes equations by local RBF-based differential quadrature method. *CMES: Computer Modeling in Engineering & Sciences*, vol. 7, pp. 195–206.

Tolstykh, A. I.; Shirobokov, D. A. (2003): On using radial basis functions in a "finite difference mode" with applications to elasticity problems. *Computational Mechanics*, vol. 33, no. 1, pp. 68–79.

Tolstykh, A. I.; Shirobokov, D. A. (2005): Using radial basis functions in a "finite difference mode". *CMES: Computer Modeling in Engineering & Sciences*, vol. 7, pp. 207–222.

Vertnik, R.; Sarler, B. (2009): Solution of incompressible turbulent flow by a mesh-free method. *CMES: Computer Modeling in Engineering & Sciences*, vol. 44, pp. 65–96.

Wright, G. B.; Fornberg, B. (2006): Scattered node compact finite difference-type formulas generated from radial basis functions. *Journal of Computational Physics*, vol. 212, no. 1, pp. 99–123.