A numerical scheme based on local integrated RBFNs and Cartesian grids for solving second-order elliptic problems in two dimensions

N. Mai-Duy¹ and T. Tran-Cong¹

Abstract: This paper reports a new Cartesian-grid computational technique, based on local integrated radialbasis-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. Numerical experiments indicate that the latter outperforms the former regarding accuracy. Moreover, the proposed local IRBFN CV method 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). 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); Wright and Fornberg (2006); Kosec and Sarler (2008); Sanyasiraju and Chandhini (2008)].

To transform a PDE into a set of algebraic equations, one needs to discretise the problem domain. For irregular domain, 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.

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 con-

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

struct 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. 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. 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.

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

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

where the superscript (*i*) is the summation index, **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 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

$$\frac{\partial^2 u(\mathbf{x})}{\partial x_j^2} = \sum_{i=1}^m w_{[x_j]}^{(i)} G^{(i)}(\mathbf{x}), \qquad (2)$$

$$\frac{\partial u(\mathbf{x})}{\partial x_j} = \sum_{i=1}^m w_{[x_j]}^{(i)} H_{[x_j]}^{(i)}(\mathbf{x}) + C_{1[x_j]}(x_k), \qquad (3)$$

$$u_{[x_j]}(\mathbf{x}) = \sum_{i=1}^m w_{[x_j]}^{(i)} \overline{H}_{[x_j]}^{(i)}(\mathbf{x}) + x_j C_{1[x_j]}(x_k) + C_{2[x_j]}(x_k),$$
(4)

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 Tanner (2005); Mai-Duy and Tran-Cong (2005)] for further details.

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). For local grids entirely embedded in the



Figure 1: Local networks in x_1 (left) and x_2 (right) (*: RBF centre and o: interior point).

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, some subscripts are dropped. The function $C(x_k)$ is constructed through

$$\frac{d^2 C(x_k)}{dx_k^2} = \sum_{i=1}^l w^{(i)} g^{(i)}(x_k), \tag{5}$$

$$\frac{dC(x_k)}{dx_k} = \sum_{i=1}^l w^{(i)} h^{(i)}(x_k) + c_1,$$
(6)

$$C(x_k) = \sum_{i=1}^{l} w^{(i)} \bar{h}^{(i)}(x_k) + x_k c_1 + c_2,$$
(7)

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 $\bar{H}^{(i)}$, respectively. Collocating (7) at the local grid points $x_k^{(i)}$ with

$$i = \{1, 2, \cdots, l\}$$
 leads to
 $\widehat{C} = \widehat{\mathscr{T}}\widehat{w},$
(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 $l \times (l+2)$

$$\begin{split} \widehat{C} &= \left(C(x_k^{(1)}), C(x_k^{(2)}), \cdots, C(x_k^{(l)}) \right)^T = \left(C^{(1)}, C^{(2)}, \cdots, C^{(l)} \right)^T, \\ \widehat{w} &= \left(w^{(1)}, w^{(2)}, \cdots, w^{(l)}, c_1, c_2 \right)^T, \\ \widehat{\mathcal{T}} &= \begin{bmatrix} \bar{h}^{(1)}(x_k^{(1)}), & \bar{h}^{(2)}(x_k^{(1)}), & \cdots, & \bar{h}^{(l)}(x_k^{(1)}), & x_k^{(1)}, & 1\\ \bar{h}^{(1)}(x_k^{(2)}), & \bar{h}^{(2)}(x_k^{(2)}), & \cdots, & \bar{h}^{(l)}(x_k^{(2)}), & x_k^{(2)}, & 1\\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots\\ \bar{h}^{(1)}(x_k^{(l)}), & \bar{h}^{(2)}(x_k^{(l)}), & \cdots, & \bar{h}^{(l)}(x_k^{(l)}), & x_k^{(l)}, & 1 \end{bmatrix}. \end{split}$$

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

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

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

where $P^{(i)}(x_k)$ is the product of the first vector on RHS and the *i*th column of $\widehat{\mathscr{T}}^+$, and $\widehat{\mathscr{T}}^+$ is the generalised inverse of $\widehat{\mathscr{T}}$.

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

$$\frac{\partial u(\mathbf{x})}{\partial x_j} = \sum_{i=1}^m w_{[x_j]}^{(i)} H_{[x_j]}^{(i)}(\mathbf{x}) + \sum_{i=1}^l P_{[x_j]}^{(i)}(x_k) C_{1[x_j]}^{(i)}, \qquad (11)$$

$$u_{[x_j]}(\mathbf{x}) = \sum_{i=1}^m w_{[x_j]}^{(i)} \overline{H}_{[x_j]}^{(i)}(\mathbf{x}) + \sum_{i=1}^l x_j P_{[x_j]}^{(i)}(x_k) C_{1[x_j]}^{(i)} + \sum_{i=1}^l P_{[x_j]}^{(i)}(x_k) C_{2[x_j]}^{(i)}.$$
(12)

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

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

$$\frac{\partial u(\mathbf{x})}{\partial x_j} = \sum_{i=1}^{m+2l} w_{[x_j]}^{(i)} H_{[x_j]}^{(i)}(\mathbf{x}), \qquad (14)$$

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

5

where

$$\{ G_{[x_j]}^{(i)}(\mathbf{x}) \}_{i=m+1}^{m+2l} \equiv \{ 0 \}_{i=1}^{2l}, \{ H_{[x_j]}^{(i)}(\mathbf{x}) \}_{i=m+1}^{m+l} \equiv \{ P_{[x_j]}^{(i)}(x_k) \}_{i=1}^{l}, \qquad \{ H_{[x_j]}^{(i)}(\mathbf{x}) \}_{i=m+l+1}^{m+2l} \equiv \{ 0 \}_{i=1}^{l}, \{ \overline{H}_{[x_j]}^{(i)}(\mathbf{x}) \}_{i=m+1}^{m+l} \equiv \{ x_j P_{[x_j]}^{(i)}(x_k) \}_{i=1}^{l}, \qquad \{ \overline{H}_{[x_j]}^{(i)}(\mathbf{x}) \}_{i=m+l+1}^{m+2l} \equiv \{ P_{[x_j]}^{(i)}(x_k) \}_{i=1}^{l}, \{ w_{[x_j]}^{(i)} \}_{i=m+1}^{m+l} \equiv \{ C_{1[x_j]}^{(i)} \}_{i=1}^{l}, \text{ and } \{ w_{[x_j]}^{(i)} \}_{i=m+l+1}^{m+2l} \equiv \{ C_{2[x_j]}^{(i)} \}_{i=1}^{l}.$$

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

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

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

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 $C_{1[x_j]}^{(i)}$ and $C_{2[x_j]}^{(i)}$, 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 $\widetilde{\mathscr{T}}_{[x_j]rs} = \overline{H}_{[x_j]}^{(s)}(\mathbf{x}^{(r)})$, where $1 \le r \le m$ and $1 \le s \le (m+2l)$.

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

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

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

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

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

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

where the function value is computed in an average sense due to numerical error.

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

6

through the middle points between the reference node and its neighbours/appropriate points on the boundary (Figure 1). Integrals can be calculated using 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.

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

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

where $\mathbf{c}^{(i)}$ and $a^{(i)}$ are the centre and width of the *i*th MQ basis function, respectively. The set of centres and the set of collocation points are identical. All MQ centres are associated with the same width that is chosen to be the grid size. 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 $0 \le x_1, x_2 \le 1$ and a circle centered at the origin with radius of 0.5, are considered. The exact solution is given by

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

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

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

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 imes 75	3.2(3)	5.2(3)	3.5(3)
87 imes 87	4.3(3)	7.1(3)	4.7(3)
99 imes 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 imes 135	1.0(4)	1.7(4)	1.1(4)
147×147	1.2(4)	2.0(4)	1.3(4)
159 imes 159	1.4(4)	2.4(4)	1.6(4)
171×171	1.7(4)	2.7(4)	1.8(4)
183 imes 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)

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 $a \times 10^{b}$.

Using the divergence theorem, (25) becomes

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

where Γ_i is the boundary of Ω_i and **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. 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 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 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



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

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, 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.

These numerical experiments 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 flow problem.

4.2 Heat flow

Find the temperature θ such that

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

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

 $\theta = 0$, for $x_2 = -0.5$ and $x_2 = 0.5$, (28)

$$\theta = \cos(\pi x_2)$$
 for $x_1 = 0$, and (29)

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



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



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



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.

11

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 imes 75	2.8(3)	8.3(3)	3.2(3)
87 imes 87	3.8(3)	1.1(4)	4.4(3)
99 imes 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 imes 135	1.0(4)	3.5(4)	1.1(4)
147×147	1.3(4)	4.6(4)	1.6(4)
159 imes 159	1.5(4)	5.8(4)	1.7(4)
171×171	1.8(4)	6.5(4)	2.4(4)
183 imes 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)

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 $a \times 10^{b}$.

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.

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



Figure 6: Heat flow: temperature values 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 $a \times 10^{-b}$.

Standard-CV	IRBFN-CV
2.69(-1)	1.00(-1)
1.83(-2)	3.69(-3)
4.25(-3)	9.36(-4)
1.83(-3)	3.47(-4)
1.01(-3)	1.75(-4)
6.45(-4)	1.11(-4)
4.46(-4)	8.32(-5)
3.27(-4)	6.92(-5)
2.50(-4)	6.15(-5)
	Standard-CV 2.69(-1) 1.83(-2) 4.25(-3) 1.83(-3) 1.01(-3) 6.45(-4) 4.46(-4) 3.27(-4) 2.50(-4)

standard control-volume techniques regarding accuracy for a given grid size, (iii) the local IRBFN controlvolume 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.

Acknowledgement: This work is supported by the Australian Research Council.

References

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, pp. 673–699.

Kosec, G.; Sarler, B. (2008): 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.

Lee, C. K.; Liu, X.; Fan, S. C. (2003): Local multiquadric approximation for solving boundary value problems. *Computational Mechanics*, vol. 30, pp. 396–409.

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

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

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, pp. 8922–8948.

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.

Wright, G. B.; Fornberg, B. (2006): Scattered node compact finite difference-type formulas generated from radial basis functions. *Journal of Computational Physics*, vol. 212, pp. 99–123.