A new integrated-rbf-based domain-embedding scheme for solving fluid-flow problems

2010 IOP Conf. Ser.: Mater. Sci. Eng. 10 012021

(http://iopscience.iop.org/1757-899X/10/1/012021)

IOP Conf. Series: Materials Science and Engineering **10** (2010) 012021 doi:10.1088/1757-899X/10/1/012021

A new integrated-rbf-based domain-embedding scheme for solving fluid-flow problems

K. Le-Cao[1], N. Mai-Duy[1], C.-D. Tran[2] and T. Tran-Cong[1]

[1]Computational Engineering and Science Research Centre
Faculty of Engineering and Surveying,
The University of Southern Queensland, Toowoomba, QLD 4350, Australia
[2] CSIRO, Geelong, VIC, Australia

E-mail: KhoaCao.Le@usq.edu.au

Abstract. This paper presents a new domain embedding numerical scheme for the simulation of flows of a Newtonian fluid in multiply-connected domains. The governing equations are taken from the stream function-vorticity formulation. The problem domain is converted into a simply-connected domain that is then discretised using a Cartesian grid. Radial-basis-function networks, which are constructed through integration rather than the usual differentiation, are employed on grid lines to approximate the field variables. Each field variable is assumed to vary over interior holes according to appropriate polynomials that satisfy the boundary conditions. Point collocation is applied to discretise the governing equations. Several linear and nonlinear problems, including natural convection in the annulus between square and circular cylinders are simulated to verify the proposed technique.

1. Introduction

Solving the Navier-Stokes equations in irregularly shaped domains presents a challenge in CFD. The concept of domain embedding or fictitious domain is known to provide an efficient way to handle complex geometries.

The basic idea of domain embedding methods/fictitious domain methods is to extend the problem defined on a geometrically-complex domain to that on a larger, but simpler shape domain. The obtained fictitious domain allows the use of a regular grid/mesh that can be fixed to represent the computational domain, and one can thus use fast direct solvers for the resultant algebraic system. All given boundary conditions must be imposed in order to match the solution on the fictitious domain with that on the original domain. Fictitious-domain techniques have been very successful in solving complicated engineering problems. Glowinski et al. [2] have presented a family of fictitious-domain techniques which are based on the explicit use of Lagrange multipliers defined on the actual boundary and associated with the boundary conditions for Dirichlet elliptic problems. Since then, the Lagrange multiplier/fictitious-domain methods have become increasingly popular. Typical examples include incompressible viscous flows (e.g. [5]), fluid/rigid-body interactions (e.g. [3]) and fluid/flexible-body interactions (e.g. [13]).

The basic equations governing the motion of a fluid can be written in different dependent variables, e.g. the velocity - pressure, stream function - vorticity and stream function formulations. Each formulation has some advantages over the others for certain classes of

problems. For the stream function - vorticity formulation, one has to derive boundary conditions for the vorticity whose accuracy strongly affects the overall solution.

There are many discretisation methods, including those based on a finite-element mesh, a finite-volume mesh, a Cartesian grid or a set of unstructured points, to reduce the PDEs to sets of algebraic equations. Among them, generating a Cartesian grid can be seen to be the most straight forward task. The use of Cartesian grids for solving problems defined on irregular domains has received much increased attention in recent decades.

In this study, we report a numerical collocation technique incorporating 1D-IRBFNs on grid lines for the simulation of heat transfers and fluid flows in multiply-connected domains. The technique combines strengths of the three approaches, namely 1D-IRBFNs, Cartesian grids and fictitious domains. It should be emphasised that conventional RBFN methods lead to fully populated matrices that tend to become ill-conditioned quickly with increasing numbers of RBFs. Instead of using conventional schemes, 1D-IRBFN approximation schemes [7,8] are utilised in the present work. In the case of natural convection, an effective formula for computing the vorticity boundary condition on a Cartesian grid is derived. First derivatives of the stream function along the boundaries are incorporated into the computational vorticity boundary values by means of integration constants. The present IRBFN approximations are constructed to satisfy all boundary conditions identically.

Through fictitious domains, the proposed technique is able to work for domains of different shapes in a similar fashion. Unlike Glowinski et al. [2], the field variables at interior holes are presently replaced by appropriate polynomials that satisfy the boundary conditions. Results obtained are compared well with available numerical data in the literature.

The remainder of the paper is organised as follows. Section 2 gives a brief review of the governing equations. In Section 3, we describe the proposed domain embedding technique. A formula for handling vorticity boundary conditions at boundary points that are not grid nodes is given in Section 4. Numerical results are presented in Section 5. Section 6 concludes the paper.

2. Governing equations

The stream function - vorticity - temperature formulation is used here. The non-dimensional basic equations for natural convection under the Boussinesq approximation in the Cartesian $x-y$ coordinate system can be written as (e.g. [9])

$$\frac{\partial^2\psi}{\partial x^2}+\frac{\partial^2\psi}{\partial y^2}=\omega, \tag{1}$$

$$\frac{\partial\omega}{\partial t}+u\frac{\partial\omega}{\partial x}+v\frac{\partial\omega}{\partial y}=\sqrt{\frac{Pr}{Ra}}\left(\frac{\partial^2\omega}{\partial x^2}+\frac{\partial^2\omega}{\partial y^2}\right)-\frac{\partial T}{\partial x}, \tag{2}$$

$$\frac{\partial T}{\partial t}+u\frac{\partial T}{\partial x}+v\frac{\partial T}{\partial y}=\frac{1}{\sqrt{RaPr}}\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}\right), \tag{3}$$

where ψ is the stream function, ω the vorticity, T the temperature, t the time, u and v the velocity components, and Pr and Ra the Prandtl and Rayleigh numbers defined as $Pr=\nu/\alpha$ and $Ra=\beta g\Delta T L^3/\alpha\nu$, in which ν is the kinematic viscosity, α the thermal diffusivity, β the thermal expansion coefficient and g the gravity, respectively. In this dimensionless scheme, L, ΔT (temperature difference), $U=\sqrt{gL\beta\Delta T}$ and (L/U), are taken as scale factors for length, temperature, velocity and time, respectively. It is noted that the velocity scale is chosen here in a way that the buoyancy and inertial forces are balanced (e.g. [9]).

The velocity components are defined in terms of the stream function as $u=\partial\psi/\partial y$ and $v=-\partial\psi/\partial x$. The given velocity boundary conditions, u and v, can be transformed into two boundary conditions on the stream function and its normal derivative $\psi=\gamma$ and $\partial\psi/\partial n=\xi$,

where n is the direction normal to the boundary, and γ and ξ prescribed functions. In the case of fixed concentric cylinders, non-slip boundary conditions usually lead to $\gamma = 0$ and $\xi = 0$.

3. One dimensional IRBFN-based domain embedding technique

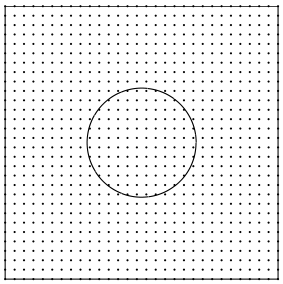

Figure 1. Computational domains and discretisations. It is noted that the real domain is the region between inner circular cylinder and outer square cylinder.

Consider a square domain with a circular hole located at the center. This physical domain is extended to a square one that can be then conveniently represented by a Cartesian grid of $m \times m$ (Figure 1). It can be seen that 1D-IRBFN expressions on the $x-$ and $y-$ grid lines have similar forms. In the following, only a horizontal grid line is considered. The second-order derivative of the field variable f along a grid line can be decomposed into RBFs

$$\frac{\partial^2 f(x)}{\partial x^2} = \sum_{i=1}^{m} w_i g_i(x) = \sum_{i=1}^{m} w_i I_i^{(2)}(x), \tag{4}$$

where m is the number of RBFs, $\{g_i(x)\}_{i=1}^m \equiv \left\{I_i^{(2)}(x)\right\}_{i=1}^m$ the set of RBFs, $\{w_i\}_{i=1}^m$ the set of weights to be found and f represents ψ, ω and T. Approximate expressions for the first-order derivative and the field variable are then obtained through integration

$$\frac{\partial f(x)}{\partial x} = \sum_{i=1}^{m} w_i I_i^{(1)}(x) + c_1, \tag{5}$$

$$f(x) = \sum_{i=1}^{m} w_i I_i^{(0)}(x) + c_1 x + c_2, \tag{6}$$

where $I_i^{(1)}(x) = \int I_i^{(2)}(x)dx$ and $I_i^{(0)}(x) = \int I_i^{(1)}(x)dx$ and (c_1, c_2) are the constants of integration. Collocating (6) at the nodal points yields

$$\widehat{f} = \widehat{\mathcal{I}}^{(0)} \begin{pmatrix} \widehat{w} \\ c_1 \\ c_2 \end{pmatrix}, \tag{7}$$

where

$$\widehat{f} = (f(x_1), f(x_2), \cdots, f(x_m))^T,$$
$$\widehat{w} = (w_1, w_2, \cdots, w_m)^T,$$
$$\widehat{\mathcal{I}}^{(0)} = \begin{bmatrix} I_1^{(0)}(x_1) & I_2^{(0)}(x_1) & \cdots & I_m^{(0)}(x_1) & x_1 & 1 \\ I_1^{(0)}(x_2) & I_2^{(0)}(x_2) & \cdots & I_m^{(0)}(x_2) & x_2 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ I_1^{(0)}(x_m) & I_2^{(0)}(x_m) & \cdots & I_m^{(0)}(x_m) & x_m & 1 \end{bmatrix}.$$

Solving (7) for the coefficient vector including the two integration constants results in

$$\begin{pmatrix} \widehat{w} \\ c_1 \\ c_2 \end{pmatrix} = \left(\widehat{\mathcal{I}}^{(0)}\right)^{-1} \widehat{f}, \tag{8}$$

where $\left(\widehat{\mathcal{I}}^{(0)}\right)^{-1}$ is the generalised inverse. The values of the first and second derivatives of f with respect to x at the nodal points are thus computed in terms of nodal variable values

$$\widehat{\frac{\partial f}{\partial x}} = \widehat{\mathcal{I}}^{(1)} \left(\widehat{\mathcal{I}}^{(0)}\right)^{-1} \widehat{f} = \widehat{\mathcal{D}}_{1x} \widehat{f}, \tag{9}$$

and

$$\widehat{\frac{\partial^2 f}{\partial x^2}} = \widehat{\mathcal{I}}^{(2)} \left(\widehat{\mathcal{I}}^{(0)}\right)^{-1} \widehat{f} = \widehat{\mathcal{D}}_{2x} \widehat{f}, \tag{10}$$

where

$$\widehat{\frac{\partial f}{\partial x}} = \left(\frac{\partial f(x_1)}{\partial x}, \frac{\partial f(x_2)}{\partial x}, \cdots, \frac{\partial f(x_m)}{\partial x}\right)^T,$$
$$\widehat{\frac{\partial^2 f}{\partial x^2}} = \left(\frac{\partial^2 f(x_1)}{\partial x^2}, \frac{\partial^2 f(x_2)}{\partial x^2}, \cdots, \frac{\partial^2 f(x_m)}{\partial x^2}\right)^T,$$
$$\widehat{\mathcal{I}}^{(1)} = \begin{bmatrix} I_1^{(1)}(x_1) & I_2^{(1)}(x_1) & \cdots & I_m^{(1)}(x_1) & 1 & 0 \\ I_1^{(1)}(x_2) & I_2^{(1)}(x_2) & \cdots & I_m^{(1)}(x_2) & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ I_1^{(1)}(x_m) & I_2^{(1)}(x_m) & \cdots & I_m^{(1)}(x_m) & 1 & 0 \end{bmatrix},$$
$$\widehat{\mathcal{I}}^{(2)} = \begin{bmatrix} g_1(x_1) & g_2(x_1) & \cdots & g_m(x_1) & 0 & 0 \\ g_1(x_2) & g_2(x_2) & \cdots & g_m(x_2) & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ g_1(x_m) & g_2(x_m) & \cdots & g_m(x_m) & 0 & 0 \end{bmatrix},$$

and $\widehat{\mathcal{D}}_{1x}$, $\widehat{\mathcal{D}}_{2x}$ are the first- and second-order differentiation matrices in the physical space.

In the case that a horizontal grid line crosses the inner hole (Figure 2), there are two interfaces as shown at x_{b2} and x_{b3}. The domain can be divided into two different parts: the region betweens the two interfaces (extended domain) and the remaining regions (real domain). The extended domain thus represents a hole inside a square cylinder. The solutions in the extended and real domains are denoted as f_f and f_r, respectively. Because of the continuity of the solution, one has $f_f = f_r$ on the interfaces. We assume that the solution on the extended domain is

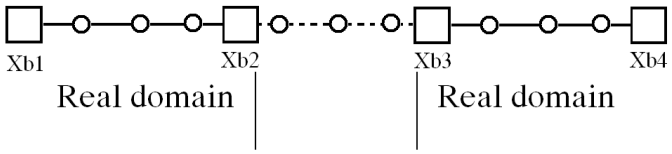

Figure 2. Points on a grid line consist of interior points x_i ($\circ$) and boundary points x_{bi} ($\square$).

known and can be described by a polynomial. This polynomial can be constructed as follows. Since there are four boundary points on the grid line, one can use a polynomial of third order, $ax^3 + bx^2 + cx + d$, whose coefficients are determined as

$$\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \mathcal{P}^{-1} \begin{pmatrix} f_{b1} \\ f_{b2} \\ f_{b3} \\ f_{b4} \end{pmatrix} \tag{11}$$

where

$$\mathcal{P} = \begin{bmatrix} x_{b1}^3 & x_{b1}^2 & x_{b1} & 1 \\ x_{b2}^3 & x_{b2}^2 & x_{b2} & 1 \\ x_{b3}^3 & x_{b3}^2 & x_{b3} & 1 \\ x_{b4}^3 & x_{b4}^2 & x_{b4} & 1 \end{bmatrix}.$$

The solution in the extended domain is thus computed as $f_f = ax^3 + bx^2 + cx + d$, where $x_{b2} \leq x \leq x_{b3}$.

For illustration purposes, the formulation is presented in detail for the Poisson equation $\nabla^2 f = b$ subject to Dirichlet boundaty conditions. Using (10) and tensor products, the PDE reduces to

$$\mathcal{A}\widehat{f} = \widehat{b}, \tag{12}$$

where $\mathcal{A} = \widehat{\mathcal{D}}_{2x} \otimes \widetilde{\mathcal{I}} + \widetilde{\mathcal{I}} \otimes \widehat{\mathcal{D}}_{2y}$. In which, $\widetilde{\mathcal{I}}$ is the identity matrix of dimensions of $m \times m$. In (12), the grid nodes are numbered from left to right and bottom to top. This system can be rearranged for the unknown values of f in the real domain as

$$\mathcal{A}(idr, idr)\widehat{f}(idr) = \widehat{b}(idr) - \mathcal{A}(idr, idb)\widehat{f}(idb) - \mathcal{A}(idr, idf)\widehat{f}(idf), \tag{13}$$

where idr, idb and idf are the sets whose elements are the indices of nodes in the real domain, on the outer boundary and in the extended domain, respectively.

4. A new formula for computing vorticity boundary conditions

It can be seen that boundary conditions are over-prescribed for the stream-function equation (1) and under-prescribed for the vorticity equation (2). We use normal derivative boundary conditions for the stream function to derive boundary conditions for the vorticity. The values of the vorticity on the boundaries can be computed via

$$\omega_b = \frac{\partial^2 \psi_b}{\partial x^2} + \frac{\partial^2 \psi_b}{\partial y^2}, \tag{14}$$

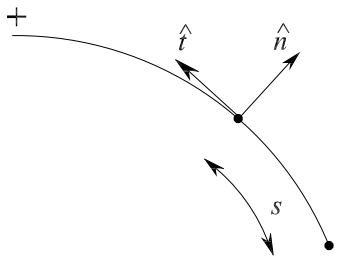

Figure 3. A curved boundary.

where the subscript b is used to indicate the boundary quantities. The handling of ω_b thus involves the evaluation of second-order derivatives of the stream function in both x and y directions.

For regular boundary points (also grid nodes), one can apply (14) directly. The x and y grid lines passing through those points can be used for computing $\partial^2\psi_b/\partial x^2$ and $\partial^2\psi_b/\partial y^2$, respectively. However, in general, the boundary points do not coincide with the grid nodes and hence they lie on either x or y grid lines. Information about ψ is thus given in one coordinate direction only. A great challenge here is how to compute second derivatives of ψ in (14) with respect to the direction without a grid line. A new formula to overcome this difficulty is proposed below.

Consider a curved boundary, along which the values for ψ and $\partial\psi/\partial n$ are prescribed (Figure 3). It can be seen that the values of $\partial\psi/\partial x$ and $\partial\psi/\partial y$ on the boundary can then be obtained in a straightforward manner. Let s be the arclength of the boundary. By introducing an interpolating scheme (e.g. 1D-IRBFNs), one is able to derive derivatives of $\partial\psi/\partial x$ and $\partial\psi/\partial y$ with respect to s such as $\partial^2\psi/\partial x\partial s$ and $\partial^2\psi/\partial y\partial s$.

A tangential derivative of a function F at a boundary point can be computed using the following formula

$$\frac{\partial F}{\partial s} = \frac{\partial F}{\partial x}t_x + \frac{\partial F}{\partial y}t_y \tag{15}$$

where t_x and t_y are the two x and y components of the unit vector $\widehat{t}$ tangential to the curve ($t_x = \partial x/\partial s$ and $t_y = \partial y/\partial s$).

Replacing F with $\partial\psi_b/\partial x$, one has

$$\frac{\partial^2\psi_b}{\partial x\partial s} = \frac{\partial^2\psi_b}{\partial x^2}t_x + \frac{\partial^2\psi_b}{\partial x\partial y}t_y, \tag{16}$$

or

$$\frac{\partial^2\psi_b}{\partial x\partial y} = \frac{1}{t_y}(\frac{\partial^2\psi_b}{\partial x\partial s} - \frac{\partial^2\psi_b}{\partial x^2}t_x), \tag{17}$$

where $\partial^2\psi_b/\partial x\partial s$ is considered as a known quantity.

Similarly, taking F as $\partial\psi_b/\partial y$ results in

$$\frac{\partial^2\psi_b}{\partial x\partial y} = \frac{1}{t_x}(\frac{\partial^2\psi_b}{\partial y\partial s} - \frac{\partial^2\psi_b}{\partial y^2}t_y), \tag{18}$$

where $\partial^2\psi_b/\partial y\partial s$ is a known value.

From (17) and (18), one can derive the relationship between $\partial^2\psi/\partial x^2$ and $\partial^2\psi/\partial y^2$ at a boundary point

$$\frac{1}{t_y}(\frac{\partial^2\psi_b}{\partial x\partial s} - \frac{\partial^2\psi_b}{\partial x^2}t_x) = \frac{1}{t_x}(\frac{\partial^2\psi_b}{\partial y\partial s} - \frac{\partial^2\psi_b}{\partial y^2}t_y). \tag{19}$$

Consider a x grid line. The interpolating scheme employed along this line does not facilitate the computation of second-order derivative of ψ with respect to the y coordinate. However, such a derivative at a boundary point can be found by using (19)

$$\frac{\partial^2 \psi_b}{\partial y^2} = (\frac{t_x}{t_y})^2 \frac{\partial^2 \psi_b}{\partial x^2} + q_y, \tag{20}$$

where q_y is a known quantity defined by

$$q_y = -\frac{t_x}{t_y^2} \frac{\partial^2 \psi_b}{\partial x \partial s} + \frac{1}{t_y} \frac{\partial^2 \psi_b}{\partial y \partial s}. \tag{21}$$

By substituting (20) into (14), a boundary condition for the vorticity at a boundary point on a horizontal grid line will be computed by

$$\omega_b = \left[1 + \left(\frac{t_x}{t_y} \right)^2 \right] \frac{\partial^2 \psi_b}{\partial x^2} + q_y, \tag{22}$$

where only the approximations in the x direction are needed.

In the same manner, on a vertical grid line, a boundary condition for the vorticity at a boundary point will be computed by

$$\omega_b = \left[1 + \left(\frac{t_y}{t_x} \right)^2 \right] \frac{\partial^2 \psi_b}{\partial y^2} + q_x, \tag{23}$$

where q_x is a known quantity defined by

$$q_x = -\frac{t_y}{t_x^2} \frac{\partial^2 \psi_b}{\partial y \partial s} + \frac{1}{t_x} \frac{\partial^2 \psi_b}{\partial x \partial s}. \tag{24}$$

The boundary conditions for the vorticity are thus written in terms of second derivative of the stream function with respect to x or y only.

5. Numerical examples

All multiply-connected domains are extended to rectangular domains. Calculations are performed on uniform Cartesian grids. The IRBFN approximations are implemented with the multiquadric function, where the RBF width is chosen to be a grid size. Three examples are employed to study the performance of the present technique.

5.1. Example 1: Poisson equation with analytic solution

This example problem is governed by

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = b(x, y), \tag{25}$$

on a domain as shown in Figure 1 with Dirichlet boundary conditions. The exact solution of this problem is taken as

$$f_e = \frac{1}{\pi^2} \sin(\pi x) \sin(\pi y), \tag{26}$$

from which the driving function $b(x, y)$ and the boundary conditions can be derived analytically. Five grids from 10×10 to 50×50 are used. The accuracy of an approximation scheme is measured by means of the discrete relative L_2 error defined as

$$Ne = \frac{\sqrt{\sum_{i=1}^{M}(f_e^i - f^i)^2}}{\sqrt{\sum_{i=1}^{M}(f_e^i)^2}} \tag{27}$$

where M is the number of unknown nodal values of f, and f_e and f are the exact and approximate solutions, respectively. Results of Ne and the condition number of the system matrix are displayed in Table 1.

Table 1. Example 1: Errors and condition numbers of the system matrix.

Grid	Error	Cond(A)
10×10	1.1e-3	1e1
20×20	5.2e-4	8e1
30×30	3.7e-4	2e2
40×40	2.3e-4	4e2
50×50	1.8e-4	6e2

5.2. Example 2: Heat transfer in a multi-hole domain

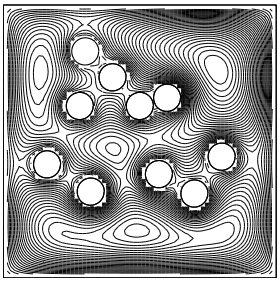

Figure 4. Contour plot.

This example is chosen to illustrate the capability of the proposed technique in handling geometrically-complex problems using a Cartesian grid. Consider the Poisson equation $\nabla^2 f = -1$ defined on a unit square with 10 holes of the same radius 0.01 and subject to Dirichlet boundary conditions $f = 0$. Figure 4 presents a contour plot of the solution f using a grid of 60×60.

5.3. Example 3: Concentric annulus between a square outer cylinder and a circular inner cylinder

This example is concerned with natural convection between a heated inner circular cylinder and a cooled square enclosure (Figure 1). An aspect ratio of $L/2R = 0.26$ (L: the side length of

Table 2. Example 3: Comparison of ψ_{max} for $Ra = 10^4, 10^5, 10^6$ between the present technique and other methods.

	ψ_{max}		
Ra	Present	[12]	[1]
$1e4$	0.96	0.82	1.0
$1e5$	8.11	7.53	8.3
$1e6$	24.6	25.54	24.13

the outer square and R: the radius of the inner circle), $Pr = 0.71$ and $Ra = \{10^4, 10^5, 10^6\}$ are employed here. Numerical results are obtained for two grids of 32×32 and 40×40.

The solution procedure involve the following main steps.

(i) Guess the distributions of T, ω and ψ.

(ii) Discretise the governing equations in time using a first-order accurate finite-difference scheme.

(iii) Discretise the governing equations in space using 1D-IRBFNs.

(iv) Solve the energy equation (3) for T.

(v) Derive computational boundary conditions for ω.

(vi) Solve the vorticity equation (1) for ω.

(vii) Solve the stream-function equation (2) for ψ.

(viii) Check to see whether the solution has reached a steady state using the following convergence measure (CM)

$$CM = \frac{\sqrt{\sum_{i=1}^{n_{ip}} \left(\psi_i^{(k)} - \psi_i^{(k-1)}\right)^2}}{\sqrt{\sum_{i=1}^{n_{ip}} \left(\psi_i^{(k)}\right)^2}} < \epsilon, \tag{28}$$

where n_{ip} is the number of interior points in the real domain, k the time level and ϵ the tolerance (in this study, ϵ is taken to be 10^{-12}).

(ix) If it is not satisfied, advance time step and repeat from step 4. Otherwise, stop the computation and output the results.

The obtained results are shown in Table 2 and Figure 5. Table 2 presents the comparison of the maximum value of the stream function between the proposed and other techniques ([1,12]), showing a good agreement. Figure 5 displays the velocity fields and isotherms for $Ra = 10^4$, $Ra = 10^5$ and $Ra = 10^6$, whose behaviours are qualitatively similar to those in [1].

6. Concluding remarks

In this article, a new domain embedding scheme for the stream function - vorticity formulation using Cartesian grids and 1D-IRBFNs is reported. Attractive features of the proposed technique include (i) The preprocessing is simple as the multiple connected domain is converted into a rectangular one, (ii) The boundary conditions for the vorticity are implemented in an effective manner and (iii) Numerical results show that the matrix condition number is relatively small. The technique is verified successfully through several test problems.

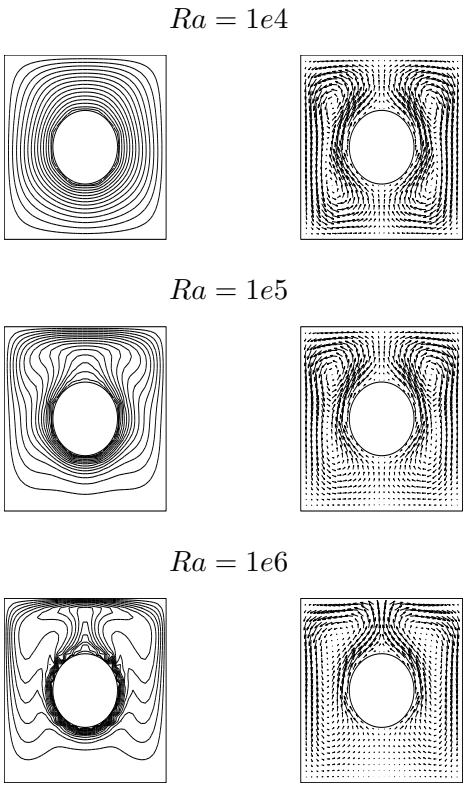

Figure 5. Square-circular cylinders: temperature (left) and velocity vector (right) fields

Acknowledgements
This research is supported by the CESRC, Faculty of Engineering and Surveying, University of Southern Queensland and Australia's Commonwealth Scientific and Industrial Research Organisation.

References
[1] Ding, H and Shu, C 2005 *Int. J. Computation and Methodology*, **47**,271-313.
[2] Glowinski, R. and Kuznetsov, Yu. 2007 *Computer Methods in Applied Mechanics and Engineering*, **96**, 14981506.
[3] Glowinski, R., Pan, T.-W., Hesla, T.I., Joseph, D.D. and Priaux, J. 2001 *J. of Comp. Phys.*, **169**, 363426.
[4] Glowinski, R., Pan, T.-W. and Priaux, J. 1994 *Comp. M. in Applied Mechanics and Engineering*, **111**, 283303.
[5] Glowinski, R., Pan, T.-W. and Priaux, J. 1994, *Computer Methods in Applied Mechanics and Engineering*, **112**, 133148.
[6] Le Cao, K., Mai-Duy, N. and Tran-Cong, T. 2009 *Num. Heat Transfer, Part B*, **55**, 480502
[7] Mai-Duy, N., Le-Cao, K. and Tran-Cong, T. 2008 *Int. J. for Numerical Methods in fluids*, **57**, 1709-30.
[8] Mai-Duy, N. and Tran-Cong, T. 2007 *Num. M. for Partial Differential Equations*,**23**(5), 1192–1210.
[9] Ostrach, S. 1988 *J. of Heat Transfer*, **110**, 1175-90.
[10] Sarler, B. 2005 *Computer Modeling in Engineering and Sciences*, **7**(2),185-194.
[11] Shu, C. and Zhu, Y.D. 2002 *Int. J. for Num. M. in Fluids*, **38**, 429-45.
[12] Wu, Y.L., Liu, G.R. and Gu, Y.T. 2005 *Num. Heat Transfer, Part B: Fundamentals*, **48** 5,459 - 475.
[13] Yu, Z. 2005 *J. of Comp. Phys.*, **207**, 127.