

Simulation of Fluid Flows at High Reynolds/Rayleigh Numbers Using Integrated Radial Basis Functions

D. Ho-Minh, K. Le-Cao, N. Mai-Duy and T. Tran-Cong

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

Abstract

In this paper, we discuss the use of integrated radial basis functions as an interpolation tool in the point-collocation, Galerkin and control-volume schemes for the discretisation of the Navier-Stokes equations. Accurate solutions for flows at high Reynolds/Rayleigh numbers are obtained using relatively-coarse grids.

Introduction

Finding numerical solutions to the Navier-Stokes equations at high values of the Reynolds/Rayleigh (Re/Ra) number is still a major challenge in CFD.

Radial basis functions (RBFs) have emerged as a powerful interpolation tool in solving differential equations [1]. Integrated RBFs (IRBFs) have the ability to avoid the problem of reduced convergence rate caused by differentiation and to provide effective ways of incorporating extra information such as multiple boundary conditions into the discrete system [4, 9, 10]. In this paper, IRBFs are incorporated into several discretisation schemes, namely point collocation, Galerkin and sub-region collocation/control-volume, to represent the field variables. We consider a Newtonian fluid and employ the governing Navier-Stokes equations in the streamfunction formulation and the streamfunction-vorticity formulation. The pressure does not have to be considered, which results in computational efficiency and ease of implementation. However, there is the need to derive boundary conditions for the vorticity variable if one uses the streamfunction-vorticity formulation, and to implement double boundary conditions if one chooses the streamfunction formulation. Since the structure of a flow is usually complex, a sufficiently large number of nodes is typically required for an accurate simulation. Local approximations have the advantage of low computational cost and they thus appear to be a preferred option. Only one-dimensional IRBFs ("local" approximations) employed with Cartesian grids are discussed here. This discussion is based on our previous works reported in [2, 3, 5, 6, 7, 8, 11].

The remainder of the paper is organised as follows. Section 2 briefly outlines 1D-IRBFs. In Section 3, three 1D-IRBF-based discretisation schemes are described. Section 4 presents some 1D-IRBF simulation results at high Re/Ra values. Section 5 concludes the paper.

One-dimensional IRBFs

Consider a function $u(x)$. The basic idea of the integral RBF scheme [9, 10] is to decompose a p th-order derivative of the function u into RBFs

$$\frac{d^p u(x)}{dx^p} = \sum_{i=1}^{N_x} w_i \phi_i(x) = \sum_{i=1}^{N_x} w_i I_i^{(p)}(x), \quad (1)$$

where $\{w_i\}_{i=1}^{N_x}$ is the set of network weights, and $\{\phi_i(x)\}_{i=1}^{N_x} \equiv \{I_i^{(p)}(x)\}_{i=1}^{N_x}$ is the set of RBFs. Lower-order derivatives and the function itself are then obtained through integration

$$\frac{d^{p-1} u(x)}{dx^{p-1}} = \sum_{i=1}^{N_x} w_i I_i^{(p-1)}(x) + c_1, \quad (2)$$

$$\frac{d^{p-2} u(x)}{dx^{p-2}} = \sum_{i=1}^{N_x} w_i I_i^{(p-2)}(x) + c_1 x + c_2, \quad (3)$$

$$\dots \dots \dots \dots \dots$$

$$u(x) = \sum_{i=1}^{N_x} w_i I_i^{(0)}(x) + c_1 \frac{x^{p-1}}{(p-1)!} + \dots + c_p, \quad (4)$$

where $I_i^{(p-1)}(x) = \int I_i^{(p)}(x) dx$, $I_i^{(p-2)}(x) = \int I_i^{(p-1)}(x) dx, \dots, I_i^{(0)}(x) = \int I_i^{(1)}(x) dx$, and (c_1, c_2, \dots, c_p) are the constants of integration.

Unlike conventional differential schemes, the starting point of the integral scheme can vary in use, depending on the particular application under consideration. The scheme is said to be of order p , denoted by 1D-IRBF- p , if the p th-order derivative is taken as the starting point.

Evaluation of (1)-(4) at a set of collocation points $\{x_i\}_{i=1}^{N_x}$ leads to

$$\widehat{\frac{d^p u}{dx^p}} = \widehat{I}_{[p]}^{(p)} \widehat{\alpha}, \quad (5)$$

$$\widehat{\frac{d^{p-1} u}{dx^{p-1}}} = \widehat{I}_{[p]}^{(p-1)} \widehat{\alpha}, \quad (6)$$

$$\dots \dots \dots$$

$$\widehat{u} = \widehat{I}_{[p]}^{(0)} \widehat{\alpha}, \quad (7)$$

where the subscript $[.]$ and superscript $(.)$ are used to denote the order of an 1D-IRBF scheme and the order of a derivative function, respectively;

$$\widehat{I}_{[p]}^{(p)} = \begin{bmatrix} I_1^{(p)}(x_1), & \dots, & I_{N_x}^{(p)}(x_1), & 0, & \dots, & 0 \\ I_1^{(p)}(x_2), & \dots, & I_{N_x}^{(p)}(x_2), & 0, & \dots, & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ I_1^{(p)}(x_{N_x}), & \dots, & I_{N_x}^{(p)}(x_{N_x}), & 0, & \dots, & 0 \end{bmatrix},$$

$$\widehat{I}_{[p]}^{(p-1)} = \begin{bmatrix} I_1^{(p-1)}(x_1), & \dots, & I_{N_x}^{(p-1)}(x_1), & 1, & \dots, & 0 \\ I_1^{(p-1)}(x_2), & \dots, & I_{N_x}^{(p-1)}(x_2), & 1, & \dots, & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ I_1^{(p-1)}(x_{N_x}), & \dots, & I_{N_x}^{(p-1)}(x_{N_x}), & 1, & \dots, & 0 \end{bmatrix},$$

.....,

$$\widehat{I}_{[p]}^{(0)} = \begin{bmatrix} I_1^{(0)}(x_1), & \dots, & I_{N_x}^{(0)}(x_1), & \frac{x_1^{p-1}}{(p-1)!}, & \dots, & 1 \\ I_1^{(0)}(x_2), & \dots, & I_{N_x}^{(0)}(x_2), & \frac{x_2^{p-1}}{(p-1)!}, & \dots, & 1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ I_1^{(0)}(x_{N_x}), & \dots, & I_{N_x}^{(0)}(x_{N_x}), & \frac{x_{N_x}^{p-1}}{(p-1)!}, & \dots, & 1 \end{bmatrix};$$

$$\widehat{\alpha} = (w_1, w_2, \dots, w_{N_x}, c_1, c_2, \dots, c_p)^T;$$

and

$$\frac{d^k \widehat{u}}{dx^k} = \left(\frac{d^k u_1}{dx^k}, \frac{d^k u_2}{dx^k}, \dots, \frac{d^k u_{N_x}}{dx^k} \right)^T, \quad k = (1, 2, \dots, p),$$

$$\widehat{u} = (u_1, u_2, \dots, u_{N_x})^T,$$

in which $d^k u_i / dx^k = d^k u(x_i) / dx^k$ and $u_i = u(x_i)$ with $i = (1, 2, \dots, N_x)$.

1D-IRBF-based discretisation schemes

A problem domain, which can be regular or irregular, is embedded in a rectangular domain that is then discretised using a Cartesian grid. The interior points are defined as grid points inside the problem domain, while the boundary points are generated by the intersection of the grid lines and the boundary. Grid nodes outside the problem domain are removed. It can be seen that this preprocessing is economical.

Consider an x grid line. The system used for the conversion of the RBF coefficient space into the physical space can be described as

$$\begin{pmatrix} \widehat{u} \\ \widehat{e} \end{pmatrix} = \begin{bmatrix} \widehat{I}_{[p]}^{(0)} \\ \widehat{\mathcal{K}} \end{bmatrix} \widehat{\alpha} = \widehat{C} \widehat{\alpha}, \quad (8)$$

where \widehat{e} , whose length can be up to p , is a vector representing extra information (e.g. normal derivative boundary conditions); $\widehat{e} = \widehat{\mathcal{K}} \widehat{\alpha}$; \widehat{u} , $\widehat{I}_{[p]}^{(0)}$ and $\widehat{\alpha}$ defined as before; and \widehat{C} the conversion matrix. It can be seen from (8) that the approximate solution u is collocated at the whole set of nodes. Solving (8) for $\widehat{\alpha}$ yields

$$\widehat{\alpha} = \widehat{C}^{-1} \begin{pmatrix} \widehat{u} \\ \widehat{e} \end{pmatrix}, \quad (9)$$

where \widehat{C}^{-1} is the inverse or pseudo-inverse of \widehat{C} , depending on its dimension. Substitution of (9) into (1)-(4) leads to

$$u(x) = (I_1^{(0)}(x), I_2^{(0)}(x), \dots) \widehat{C}^{-1} \begin{pmatrix} \widehat{u} \\ \widehat{e} \end{pmatrix}, \quad (10)$$

$$\frac{\partial^p u(x)}{\partial x^p} = (I_1^{(p)}(x), I_2^{(p)}(x), \dots) \widehat{C}^{-1} \begin{pmatrix} \widehat{u} \\ \widehat{e} \end{pmatrix}. \quad (11)$$

They can be rewritten in the form

$$u(x) = \sum_{i=1}^{N_x} \varphi_i(x) u_i + \varphi_{N_x+1}(x) e_1 + \varphi_{N_x+2}(x) e_2 + \dots, \quad (12)$$

$$\frac{\partial^p u(x)}{\partial x^p} = \sum_{i=1}^{N_x} \frac{d^p \varphi_i(x)}{dx^p} u_i + \frac{d^p \varphi_{N_x+1}(x)}{dx^p} e_1 + \frac{d^p \varphi_{N_x+2}(x)}{dx^p} e_2 + \dots \quad (13)$$

Three discretisation schemes, namely point collocation, Galerkin, and control volume, are considered. For illustration purposes, consider a differential problem

$$L(u) = 0, \quad (14)$$

defined on a rectangular domain Ω , where L is an elliptic operator. Assume that the variable u is prescribed on the boundary of Ω (Dirichlet boundary condition). To find its interior values, a number of algebraic equations that is equal to a number of unknowns needs to be generated.

Point collocation scheme

The algebraic equation set is generated by collocating the governing differential equation at the interior points

$$L(u(x_i, y_j)) = 0, \quad (15)$$

where $2 \leq i \leq (N_x - 1)$ and $2 \leq j \leq (N_y - 1)$.

Galerkin scheme

The Galerkin weighting process applied to (14) produces the following results

$$\int_{\Omega} \varphi_i \varphi_j L(u) d\Omega = 0, \quad (16)$$

where $2 \leq i \leq (N_x - 1)$ and $2 \leq j \leq (N_y - 1)$. The above volume integrals can be evaluated using repeated integrals, for which Gauss quadrature points are employed along the grid lines.

Control-volume scheme

For each grid point (x_i, y_j) , one can construct a control volume $\Omega_{i,j}$ that is non-overlapping with each other. By means of the control-volume statement, one has

$$\int_{\Omega_{i,j}} L(u) = 0, \quad (17)$$

where $2 \leq i \leq (N_x - 1)$ and $2 \leq j \leq (N_y - 1)$. The Gauss divergence theorem can be applied to (17) to transform the volume integral into surface integrals.

1D-IRBFs are incorporated into (15), (16) and (17) to represent the field variables on each grid line.

Numerical results

The Navier-Stokes equations are taken in the streamfunction formulation and the streamfunction-vorticity formulation. We employ 1D-IRBF-4s and 1D-IRBF-2s with the multiquadric (MQ) basis function for the discretisation of the streamfunction and the streamfunction-vorticity formulations, respectively. The grid size is chosen as the MQ width. The set of nodal points is taken as the set of MQ centres. The extra information vector \widehat{e} in (8) is set (i) to null in solving the PDEs, (ii) to derivative boundary values of the streamfunction in deriving computational boundary conditions for the vorticity, and (iii) to derivative boundary values in imposing Neumann and double boundary conditions. The performance of 1D-IRBF-based discretisation schemes is numerically investigated through the steady-state solution of the following three problems.

Example 1: Natural convection heat transfer from a heated inner circular cylinder to a cooled square closure, streamfunction formulation, point collocation scheme (Table 1 and Figure 1)

Example 2: Natural convection in a square cavity,

streamfunction-vorticity formulation, Galerkin scheme (Table 2 and Figure 2)

Example 3: Lid-driven cavity flow, streamfunction-vorticity formulation, control-volume scheme (Figure 3)

Table 1: Example 1: the average Nusselt number.

Ra	10^4	10^5	10^6
Present	3.22	4.90	8.72
[12]	3.24	4.86	8.90

Table 2: Example 2: the average Nusselt number throughout the cavity (N_u) and on the middle cross section ($N_{u1/2}$).

	N_u	$N_{u1/2}$
Present	30.548	30.525
[13]	30.225	30.225

The obtained results agree very well with the benchmark solutions available in the literature.

Concluding remarks

In this paper, trial functions are implemented using 1D-IRBFs rather than the usual low-order polynomials for the solution of the streamfunction and the streamfunction-vorticity formulations. Attractive features of 1D-IRBFs include (i) to provide effective treatments of irregular boundary geometries for Cartesian-grid collocation methods, (ii) to offer a proper way of implementing Neumann and double boundary conditions (through integration constants) and (iii) to provide an effective way of deriving a boundary condition for the vorticity (through integration constants). Numerical examples show that 1D-IRBF-based discretisation schemes are able to achieve accurate simulations at high Re/Ra values.

Acknowledgements

This work is supported by the Australian Research Council.

References

- [1] Fasshauer, G.E., *Meshfree Approximation Methods With Matlab*, World Scientific Publishers, 2007.
- [2] Ho-Minh, D., Mai-Duy, N. and Tran-Cong, T., A Galerkin-RBF approach for the streamfunction-vorticity-temperature formulation of natural convection in 2D enclosed domains, *Computer Modeling in Engineering & Sciences*, **44**, 2009, 219–248.
- [3] Le-Cao, K., Mai-Duy, N. and Tran-Cong, T., An effective integrated-RBFN Cartesian-grid discretisation for the stream function-vorticity-temperature formulation in non-rectangular domains, *Numerical Heat Transfer, Part B*, **55**, 2009, 480–502.
- [4] Mai-Duy, N., Solving high order ordinary differential equations with radial basis function networks, *International Journal for Numerical Methods in Engineering*, **62**, 2005, 824–852.
- [5] Mai-Duy, N., Ho-Minh, D. and Tran-Cong, T., A Galerkin approach incorporating integrated radial basis function networks for the solution of biharmonic equations in two dimensions, *International Journal of Computer Mathematics*, **86**, 2009, 1746–1759.
- [6] Mai-Duy, N., Le-Cao, K. and Tran-Cong, T., A Cartesian grid technique based on one dimensional integrated radial basis function networks for natural convection in concentric annuli, *International Journal for Numerical Methods in Fluids*, **57**, 2008, 1709–1730.

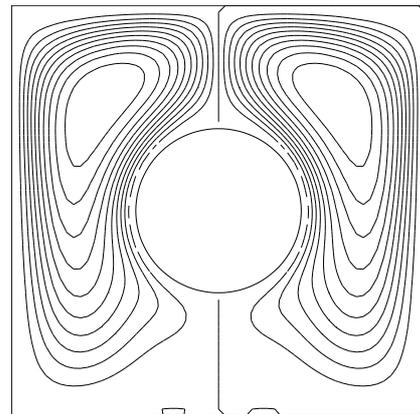
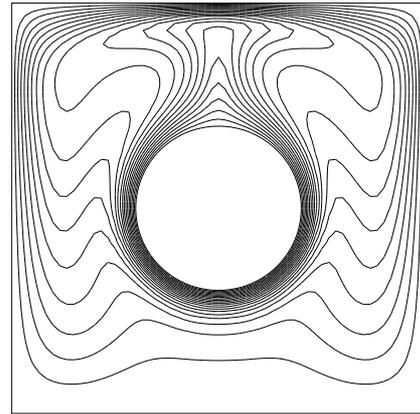
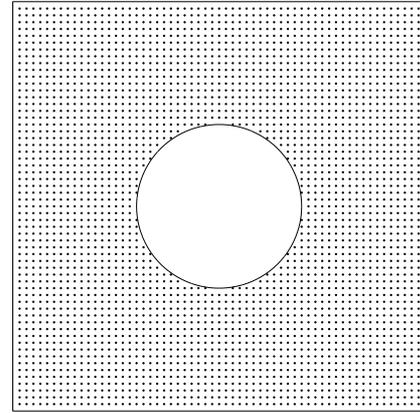


Figure 1: Example 1, natural convection heat transfer from a heated inner circular cylinder to a cooled square closure, $Ra = 10^6$, 61×61 : Computational domain and discretisation (top), isotherms (middle) and streamlines (bottom).

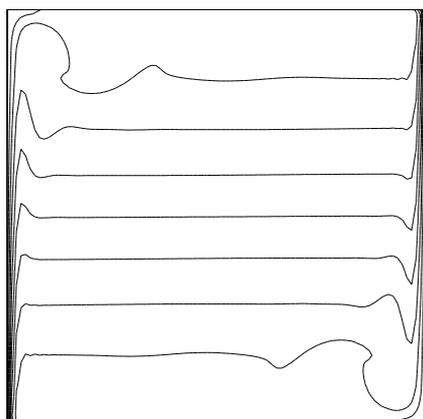
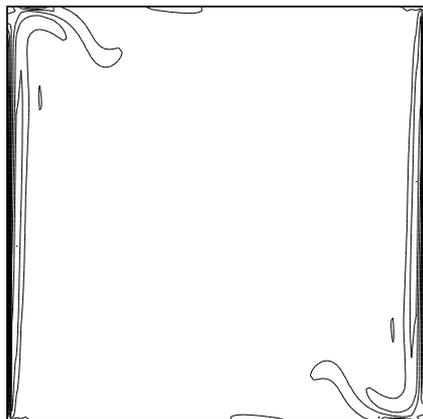
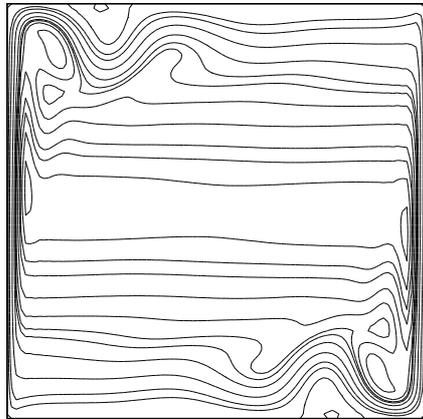


Figure 2: Example 2, natural convection heat transfer from a heated left wall to a cooled right wall (adiabatic along the bottom and to walls), $Ra = 10^8$, 91×91 : Contour plots for the streamfunction (top), vorticity (middle) and temperature (bottom).

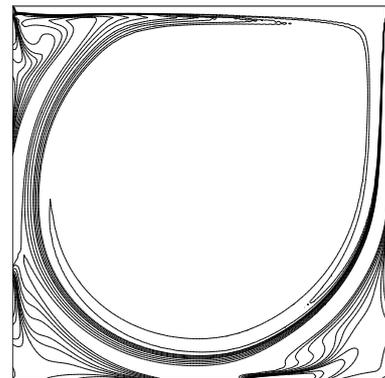
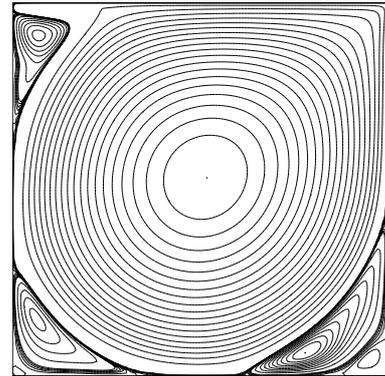


Figure 3: Example 3, lid-driven cavity flow, $Re = 7500$, 131×131 : Contour plots for the streamfunction (top) and vorticity (bottom). Top lid moves from left to right.

- [7] Mai-Duy, N., Mai-Cao, L. and Tran-Cong, T., Computation of transient viscous flows using indirect radial basis function networks, *Computer Modeling in Engineering & Sciences*, **18**, 2007, 59–77.
- [8] Mai-Duy, N. and Tanner, R., Solving high order partial differential equations with radial basis function networks, *International Journal for Numerical Methods in Engineering*, **63**, 2005, 1636–1654.
- [9] Mai-Duy, N. and Tran-Cong, T., Numerical solution of differential equations using multiquadric radial basis function networks, *Neural Networks*, **14**, 2001, 185–199.
- [10] Mai-Duy, N. and Tran-Cong, T., Approximation of function and its derivatives using radial basis function networks, *Applied Mathematical Modelling*, **27**, 2003, 197–220.
- [11] Mai-Duy, N. and Tran-Cong, T., A control volume technique based on integrated RBFNs for the convection-diffusion equation, *Numerical Methods for Partial Differential Equations*, **26**, 2010, 426–447.
- [12] Shu, C. and Zhu, Y.D., Efficient computation of natural convection in a concentric annulus between an outer square cylinder and an inner circular cylinder, *International Journal for Numerical Methods in Fluids*, **38**, 2002, 429–445.
- [13] Quéré, P.L., Accurate solutions to the square thermally driven cavity at high Rayleigh number, *Computers & Fluids*, **20**, 1991, 29–41.