

A High-order Coupled Compact Integrated RBF Approximation Based Domain Decomposition Algorithm for Second-order Differential Problems

C.M.T. Tien¹, N. Pham-Sy¹, N. Mai-Duy¹, C.-D. Tran¹ and T. Tran-Cong¹

Abstract: This paper presents a high-order coupled compact integrated RBF (CCIRBF) approximation based domain decomposition (DD) algorithm for the discretisation of second-order differential problems. Several Schwarz DD algorithms, including one-level additive/ multiplicative and two-level additive/ multiplicative/ hybrid, are employed. The CCIRBF based DD algorithms are analysed with different mesh sizes, numbers of subdomains and overlap sizes for Poisson problems. Our convergence analysis shows that the CCIRBF two-level multiplicative version is the most effective algorithm among various schemes employed here. Especially, the present CCIRBF two-level method converges quite rapidly even when the domain is divided into many subdomains, which shows great promise for either serial or parallel computing. For practical tests, we then incorporate the CCIRBF into serial and parallel two-level multiplicative Schwarz. Several numerical examples, including those governed by Poisson and Navier-Stokes equations are analysed to demonstrate the accuracy and efficiency of the serial and parallel algorithms implemented with the CCIRBF. Numerical results show: (i) the CCIRBF-Serial and -Parallel algorithms have the capability to reach almost the same solution accuracy level of the CCIRBF-Single domain, which is ideal in terms of computational calculations; (ii) the CCIRBF-Serial and -Parallel algorithms are highly accurate in comparison with standard finite difference, compact finite difference and some other schemes; (iii) the proposed CCIRBF-Serial and -Parallel algorithms may be used as alternatives to solve large-size problems which the CCIRBF-Single domain may not be able to deal with. The ability of producing stable and highly accurate results of the proposed serial and parallel schemes is believed to be the contribution of the coarse mesh of the two-level domain decomposition and the CCIRBF approximation. It is noted that the focus of this paper is on the derivation of highly accurate serial and parallel algorithms for second-order differential problems. The scope of this work does not cover a thorough analysis of computational time.

¹ Computational Engineering and Science Research Centre, Faculty of Health, Engineering and Sciences, The University of Southern Queensland, Toowoomba, Queensland 4350, Australia.

Keywords: Coupled compact integrated RBF (CCIRBF), Schwarz, domain decomposition, one-level, two-level, coarse meshes, additive, multiplicative, hybrid, serial, parallel, colouring technique, Poisson equation, Navier-Stokes equation, lid driven cavity.

1 Introduction

Traditional techniques such as the finite difference method (FDM), finite volume method (FVM), finite element method (FEM) and boundary element method (BEM) are among the most popular numerical solution methods for partial differential equations (PDEs) governing many problems in engineering and sciences. These methods are based on some discretisation of a problem domain into small elements. These elements are not overlapping each other. If an element is heavily distorted, approximations on this element are of poor quality, leading to unacceptable accuracy or possibly failed computation. Element-free methods are developed to address the issues associated with element distortions by using different approximation methods over a cluster of scattered nodes. The smooth particle hydrodynamics method (SPH) [Lucy (1977)] is one of the initial and well developed element-free methods. The diffusive element method (DEM) [Nayroles, Touzot, and Villon (1992)] was the first element-free method to employ moving least squares (MLS) approximation [Lancaster and Salkauskas (1981)] in constructing their shape functions over scattered nodes. Several element-free methods have been proposed since then, including the element-free Galerkin method (EFG) [Belytschko, Lu, and Gu (1994)], reproducing kernel particle method (RKPM) [Liu, Chen, Chang, and Belytschko (1996)], partition of unity (PU) method [Babuska and Melenk (1997)] and meshless local Petrov-Galerkin method (MLPG) [Atluri and Zhu (1998)]. For an overview on these element-free methods, readers may find more details in [Belytschko, Krongauz, Organ, Fleming, and Krysl (1996); Chen, Lee, and Eskandarian (2006)] and references therein.

In the last three decades, there has been great interest in using element-free radial basis function (RBF) methods for the numerical solutions of various types of PDEs. Kansa (1990a,b) introduced a new approach for this kind of problems, using radial basis functions (RBFs) (here referred as differential/direct RBF or DRBF) for the approximate solutions of PDEs. Mai-Duy and Tran-Cong (2001a,b, 2003) then proposed an idea of using indirect/integrated radial basis functions (IRBFs) for the solution of PDEs. Numerical examples in Mai-Duy and Tran-Cong (2001a,b, 2003, 2005) show that the IRBF approach achieves a greater accuracy than the DRBF approach. It has been shown that these RBF methods are more accurate than the traditional techniques such as the FDM, FVM and FEM [Zerroukat, Power, and Chen (1998); Li, Cheng, and Chen (2003); Thai-Quang, Mai-Duy, Tran, and Tran-Cong

(2012)]. Furthermore, the RBF approaches can work with simple discretisation based on a Cartesian grid. However, when dealing with large-scale problems, a big obstacle for the global RBF method is that the system matrix is generally ill-conditioned, non-symmetric and dense. Therefore, the RBF method needs to be combined with the domain decomposition (DD) method to reduce the density and ill-conditioning of the matrix for an accurate solution.

The earliest idea of DD was introduced as a classical Schwarz alternating algorithm by Schwarz in 1870. Generally, DD methods can be classified into two major methods: overlapping methods, which are referred to as Schwarz methods, and non-overlapping methods, which are referred to as iterative substructuring or Schur complement methods [Smith, Bjorstad, and Gropp (1996); Quarteroni and Valli (1999); Toselli and Widlund (2005)]. In this work, we will concentrate on iterative Schwarz DD methods using overlapping subdomains. The overlapping DD methods have a simple algorithmic structure because there is no need to solve the continuity problem across subdomain interfaces [Cai (2003)]. The overlapping methods provide parallel, potentially fast and robust algorithms for the solution of linear or nonlinear systems of equations resulting usually from the discretisation of PDEs.

It is noted that the convergence characteristics of the DD based methods are sensitive to the choice of the number of subdomains, mesh sizes and overlap sizes. In particular, having too many subdomains leads to a very large coarse mesh problem, while having too few subdomains requires the solution of large problems for each subdomain. Furthermore, having too small overlaps usually leads to large number of iterations, while having too large overlaps leads to the solution of large problems for each subdomain. In this point of view, efficient DD based methods should stably converge with small number of iterations for a wide range of numbers of subdomains and mesh sizes, using small number of overlaps.

In this paper, we investigate convergence characteristics of the recently developed three-point coupled compact integrated RBF (CCIRBF) approximation scheme proposed in [Tien, Thai-Quang, Mai-Duy, Tran, and Tran-Cong (2015)] when incorporated into the Schwarz DD algorithms for solving Poisson problems. Different types of Schwarz DD algorithms are utilised, including the one-level additive/ multiplicative and two-level additive/ multiplicative/ hybrid Schwarz. In the one-level algorithm, a fine mesh problem on each subdomain is solved and the subdomain solutions are interpolated back to the global grid. The two-level algorithm is formulated by adding the coarse mesh problem to the one-level problem. The use of the coarse mesh generally reduces the number of iterations. The present CCIRBF based Schwarz DD algorithms are investigated with various grid sizes, number of subdomains and overlap sizes. It is found that the present CCIRBF two-level multiplicative version is far better than the other CCIRBF based DD versions in terms

of iteration count. The present CCIRBF two-level multiplicative version shows a great promise for both serial and parallel computing because it is stable with various numbers of subdomains and grid sizes while being able to converge quickly with very small overlap sizes.

Then, we incorporate the CCIRBF into serial and parallel two-level multiplicative Schwarz for practical tests. We parallelise problems which are decomposed by the two-level multiplicative Schwarz with a colouring technique. The serial and parallel algorithm are so called CCIRBF-Serial and -Parallel, respectively. To analyse their accuracy and efficiency, analytical examples including Poisson and Navier-Stokes equations are performed. Lid driven cavity problems, in which Taylor-series type boundary condition for vorticity is first implemented in the context of the CCIRBF, are also analysed as practical applications. Numerical results show: (i) the results produced by CCIRBF-Serial and -Parallel have almost the same solution accuracy with those calculated by the CCIRBF-Single domain, which is computationally ideal; (ii) the CCIRBF-Serial and -Parallel algorithms are highly accurate in comparison with standard FDM, higher-order compact finite difference (HOC) and some other schemes; (iii) the proposed CCIRBF-Serial and -Parallel algorithms can efficiently solve large-size problems. The proposed CCIRBF-Serial and Parallel algorithms may be used effectively for large-scale problems which the single domain algorithms are struggling to handle.

This paper is organised as follows. Section 2 reviews the CCIRBF approximation scheme. In Section 3, we briefly describe the one-level additive/ multiplicative and two-level additive/ multiplicative/ hybrid. In Section 4, the GMRES iterative method is briefly mentioned. Section 5 explains the serial and parallel two-level multiplicative Schwarz DD methods, followed by Section 6 which details the parallel technique. Numerical examples demonstrating the convergence analysis and effectiveness of the algorithms are presented in Section 8. Finally, the concluding remarks of the article are given in Section 9.

2 CCIRBF scheme

The coupled compact integrated radial basis function (CCIRBF) approximation scheme developed by Tien, Thai-Quang, Mai-Duy, Tran, and Tran-Cong (2015) is utilised in this paper. Readers may find more details about CCIRBF scheme in [Tien, Thai-Quang, Mai-Duy, Tran, and Tran-Cong (2015)], which are summarised here for convenience.

The essence of the CCIRBF scheme is to couple extra information of the nodal first and second derivative values via their identity equations, which makes the scheme more accurate, stable and efficient. Hereafter, for brevity, η denotes either x or

y in a generic local stencil $\{\eta_1, \eta_2, \eta_3\}$, where $\eta_1 < \eta_2 < \eta_3$ and $\eta_2 \equiv \eta_{(i,j)}$, are illustrated in Figure 1.

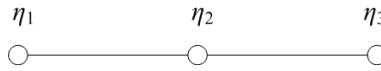


Figure 1: Compact three-point 1D-IRBF stencil for interior nodes.

2.1 First derivatives at interior nodes

For the coupled compact approximation of the first derivatives at interior nodes, nodal derivative values (i.e. extra information) are chosen as not only $\left\{ \frac{du_1}{d\eta}, \frac{du_3}{d\eta} \right\}$ but also $\left\{ \frac{d^2u_1}{d\eta^2}, \frac{d^2u_3}{d\eta^2} \right\}$. At a particular interior node, the approximation is processed through three steps: (i) we first approximate its first derivative over its associated three-point stencil involving $\left\{ \frac{du_1}{d\eta}, \frac{du_3}{d\eta} \right\}$; (ii) we then approximate its first derivative over the same stencil used in step (i) involving $\left\{ \frac{d^2u_1}{d\eta^2}, \frac{d^2u_3}{d\eta^2} \right\}$; (iii) an identity equation of the first derivative is employed to enhance the level of compactness of the stencil. Both $\left\{ \frac{du_1}{d\eta}, \frac{du_3}{d\eta} \right\}$ and $\left\{ \frac{d^2u_1}{d\eta^2}, \frac{d^2u_3}{d\eta^2} \right\}$ are incorporated into the first derivative approximation.

2.1.1 First derivatives at interior nodes involving $\left\{ \frac{du_1}{d\eta}, \frac{du_3}{d\eta} \right\}$

At $\eta = \eta_2$, the approximation formulation of the stencil is expressed in the matrix-vector form as

$$\begin{bmatrix} -\mu_{1F4} & 1 & -\mu_{1F5} \end{bmatrix} \mathbf{u}'' + \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \mathbf{u}''' = \begin{bmatrix} \mu_{1F1} & \mu_{1F2} & \mu_{1F3} \end{bmatrix} \mathbf{u}^n, \quad (1)$$

where $\{\mu_{1Fi}\}_{i=1}^5$ is the set of IRBFs in the physical space, in which 1 and F stand for the 1st derivatives to be approximated and the extra information of the nodal first derivative values chosen, respectively; $\mathbf{u}'' = [u_1'', u_2'', u_3'']^T$; $\mathbf{u}''' = [u_1''', u_2''', u_3''']^T$; and $\mathbf{u}^n = [u_1^n, u_2^n, u_3^n]^T$. It is noted that \mathbf{u}''' is introduced here to produce a general form for the coupling task which is mentioned later on.

2.1.2 First derivatives at interior nodes involving $\left\{ \frac{d^2u_1}{d\eta^2}, \frac{d^2u_3}{d\eta^2} \right\}$

At $\eta = \eta_2$, the approximation formulation of the stencil is expressed in the matrix-vector form as

$$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \mathbf{u}'' + \begin{bmatrix} -\mu_{1S4} & 0 & -\mu_{1S5} \end{bmatrix} \mathbf{u}''' = \begin{bmatrix} \mu_{1S1} & \mu_{1S2} & \mu_{1S3} \end{bmatrix} \mathbf{u}^n, \quad (2)$$

where $\{\mu_{1Si}\}_{i=1}^5$ is the set of IRBFs in the physical space, in which 1 and S stand for the 1st derivatives to be approximated and the extra information of the nodal second derivative values chosen, respectively; $\mathbf{u}^n = [u_1^n, u_2^n, u_3^n]^T$; $\mathbf{u}^{nn} = [u_1^{nn}, u_2^{nn}, u_3^{nn}]^T$; and $\mathbf{u}^n = [u_1^n, u_2^n, u_3^n]^T$.

2.1.3 First derivative couplings at interior nodes

At $\eta = \eta_2$, a coupling equation in matrix-vector form is described as

$$\begin{aligned} \begin{bmatrix} \mu_{1F4} & 0 & \mu_{1F5} \end{bmatrix} \mathbf{u}^n + \begin{bmatrix} -\mu_{1S4} & 0 & -\mu_{1S5} \end{bmatrix} \mathbf{u}^{nn} \\ = \begin{bmatrix} (\mu_{1S1} - \mu_{1F1}) & (\mu_{1S2} - \mu_{1F2}) & (\mu_{1S3} - \mu_{1F3}) \end{bmatrix} \mathbf{u}^n, \end{aligned} \quad (3)$$

where $\mathbf{u}^n = [u_1^n, u_2^n, u_3^n]^T$; $\mathbf{u}^{nn} = [u_1^{nn}, u_2^{nn}, u_3^{nn}]^T$; and $\mathbf{u}^n = [u_1^n, u_2^n, u_3^n]^T$.

2.2 First derivatives at boundary nodes

At the boundary nodes, the first derivatives are approximated in special compact stencils. Consider the boundary node, e.g. η_1 . Its associated stencil is $\{\eta_1, \eta_2, \eta_3, \eta_4\}$ as shown in Figure 2. For the coupled compact approximation of the first derivative at the boundary node η_1 , nodal derivative values (i.e. extra information) are chosen as both $\frac{du_2}{d\eta}$ and $\frac{d^2u_2}{d\eta^2}$. The approximation is processed through three steps: (i) we first approximate its first derivative over its associated four-point stencil involving $\frac{du_2}{d\eta}$; (ii) we then approximate its first derivative over the same stencil used in step (i) involving $\frac{d^2u_2}{d\eta^2}$; (iii) an identity equation of the second derivative is employed to enhance the level of compactness of the stencil. Both $\frac{du_2}{d\eta}$ and $\frac{d^2u_2}{d\eta^2}$ are incorporated into the second derivative approximation.



Figure 2: Special compact four-point 1D-IRBF stencil for boundary nodes.

2.2.1 First derivatives at boundary node η_1 involving $\frac{du_2}{d\eta}$

$$\begin{aligned} \begin{bmatrix} 1 & -\mu_{sp1F5} & 0 & 0 \end{bmatrix} \mathbf{u}^n + \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{u}^{nn} \\ = \begin{bmatrix} \mu_{sp1F1} & \mu_{sp1F2} & \mu_{sp1F3} & \mu_{sp1F4} \end{bmatrix} \mathbf{u}^n, \end{aligned} \quad (4)$$

where $\mathbf{u}^n = [u_1^n, u_2^n, u_3^n, u_4^n]^T$; $\mathbf{u}^{nn} = [u_1^{nn}, u_2^{nn}, u_3^{nn}, u_4^{nn}]^T$; and $\mathbf{u}^n = [u_1^n, u_2^n, u_3^n, u_4^n]^T$.

2.2.2 *First derivatives at boundary node η_1 involving $\frac{d^2u_2}{d\eta^2}$*

$$\begin{aligned} & \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \mathbf{u}^n + \begin{bmatrix} 0 & -\mu_{sp1S5} & 0 & 0 \end{bmatrix} \mathbf{u}''' \\ & = \begin{bmatrix} \mu_{sp1S1} & \mu_{sp1S2} & \mu_{sp1S3} & \mu_{sp1S4} \end{bmatrix} \mathbf{u}^n, \end{aligned} \quad (5)$$

where $\mathbf{u}^n = [u_1^n, u_2^n, u_3^n, u_4^n]^T$; $\mathbf{u}''' = [u_1''', u_2''', u_3''', u_4''']^T$; and $\mathbf{u}^n = [u_1^n, u_2^n, u_3^n, u_4^n]^T$.

2.2.3 *First derivative coupling at boundary node η_1*

$$\begin{aligned} & \begin{bmatrix} 0 & \mu_{sp1F5} & 0 & 0 \end{bmatrix} \mathbf{u}'' + \begin{bmatrix} 0 & -\mu_{sp1S5} & 0 & 0 \end{bmatrix} \mathbf{u}''' \\ & = \begin{bmatrix} (\mu_{sp1S1} - \mu_{sp1F1}) & (\mu_{sp1S2} - \mu_{sp1F2}) & (\mu_{sp1S3} - \mu_{sp1F3}) & (\mu_{sp1S4} - \mu_{sp1F4}) \end{bmatrix} \mathbf{u}^n, \end{aligned} \quad (6)$$

where $\mathbf{u}'' = [u_1'', u_2'', u_3'', u_4'']^T$; $\mathbf{u}''' = [u_1''', u_2''', u_3''', u_4''']^T$; and $\mathbf{u}^n = [u_1^n, u_2^n, u_3^n, u_4^n]^T$. In a similar manner, one is able to calculate the first derivative at the boundary node $\eta_{n\eta}$.

2.3 Second derivatives at interior nodes

In a similar manner as in Section 2.1, one is able to calculate the coupled compact approximation of the second derivatives at interior nodes as follows.

2.3.1 *Second derivatives at interior nodes involving $\left\{ \frac{du_1}{d\eta}, \frac{du_3}{d\eta} \right\}$*

At $\eta = \eta_2$, the approximation formulation of the stencil is expressed in the matrix-vector form as

$$\begin{bmatrix} -v_{2F4} & 0 & -v_{2F5} \end{bmatrix} \mathbf{u}'' + \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \mathbf{u}''' = \begin{bmatrix} v_{2F1} & v_{2F2} & v_{2F3} \end{bmatrix} \mathbf{u}^n, \quad (7)$$

where $\{v_{2Fi}\}_{i=1}^5$ is the set of IRBFs in the physical space, in which 2 and F stand for the 2nd derivatives to be approximated and the extra information of the first derivatives, respectively; $\mathbf{u}'' = [u_1'', u_2'', u_3'']^T$; $\mathbf{u}''' = [u_1''', u_2''', u_3''']^T$; and $\mathbf{u}^n = [u_1^n, u_2^n, u_3^n]^T$.

2.3.2 *Second derivatives at interior nodes involving $\left\{ \frac{d^2u_1}{d\eta^2}, \frac{d^2u_3}{d\eta^2} \right\}$*

At $\eta = \eta_2$, the approximation formulation of the stencil is expressed in the matrix-vector form as

$$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \mathbf{u}'' + \begin{bmatrix} -v_{2S4} & 1 & -v_{2S5} \end{bmatrix} \mathbf{u}''' = \begin{bmatrix} v_{2S1} & v_{2S2} & v_{2S3} \end{bmatrix} \mathbf{u}^n, \quad (8)$$

where $\{v_{2Si}\}_{i=1}^5$ is the set of IRBFs in the physical space, in which 2 and S stand for the 2nd derivatives to be approximated and the extra information of the second derivatives, respectively; $\mathbf{u}^m = [u_1^n, u_2^n, u_3^n]^T$; $\mathbf{u}''^m = [u_1''^m, u_2''^m, u_3''^m]^T$; and $\mathbf{u}^n = [u_1^n, u_2^n, u_3^n]^T$.

2.3.3 Second derivative couplings at interior nodes

At $\eta = \eta_2$, a coupling equation in the matrix-vector form is described as

$$\begin{aligned} \begin{bmatrix} v_{2F4} & 0 & v_{2F5} \end{bmatrix} \mathbf{u}^m + \begin{bmatrix} -v_{2S4} & 0 & -v_{2S5} \end{bmatrix} \mathbf{u}''^m \\ = \begin{bmatrix} (v_{2S1} - v_{2F1}) & (v_{2S2} - v_{2F2}) & (v_{2S3} - v_{2F3}) \end{bmatrix} \mathbf{u}^n, \end{aligned} \quad (9)$$

where $\mathbf{u}^m = [u_1^n, u_2^n, u_3^n]^T$; $\mathbf{u}''^m = [u_1''^m, u_2''^m, u_3''^m]^T$; and $\mathbf{u}^n = [u_1^n, u_2^n, u_3^n]^T$.

2.4 Second derivatives at boundary nodes

In a similar manner as in Section 2.2, one is able to calculate the coupled compact approximation of the second derivatives at boundary nodes as follows.

2.4.1 Second derivatives at boundary node η_1 involving $\frac{du_2}{d\eta}$

$$\begin{aligned} \begin{bmatrix} 0 & -v_{sp2F5} & 0 & 0 \end{bmatrix} \mathbf{u}^m + \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \mathbf{u}''^m \\ = \begin{bmatrix} v_{sp2F1} & v_{sp2F2} & v_{sp2F3} & v_{sp2F4} \end{bmatrix} \mathbf{u}^n, \end{aligned} \quad (10)$$

where $\mathbf{u}^m = [u_1^n, u_2^n, u_3^n, u_4^n]^T$; $\mathbf{u}''^m = [u_1''^m, u_2''^m, u_3''^m, u_4''^m]^T$; and $\mathbf{u}^n = [u_1^n, u_2^n, u_3^n, u_4^n]^T$.

2.4.2 Second derivatives at boundary node η_1 involving $\frac{d^2u_2}{d\eta^2}$

$$\begin{aligned} \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{u}^m + \begin{bmatrix} 1 & -v_{sp2S5} & 0 & 0 \end{bmatrix} \mathbf{u}''^m \\ = \begin{bmatrix} v_{sp2S1} & v_{sp2S2} & v_{sp2S3} & v_{sp2S4} \end{bmatrix} \mathbf{u}^n, \end{aligned} \quad (11)$$

where $\mathbf{u}^m = [u_1^n, u_2^n, u_3^n, u_4^n]^T$; $\mathbf{u}''^m = [u_1''^m, u_2''^m, u_3''^m, u_4''^m]^T$; and $\mathbf{u}^n = [u_1^n, u_2^n, u_3^n, u_4^n]^T$.

2.4.3 Second derivative coupling at boundary node η_1

$$\begin{aligned} \begin{bmatrix} 0 & v_{sp2F5} & 0 & 0 \end{bmatrix} \mathbf{u}^m + \begin{bmatrix} 0 & -v_{sp2S5} & 0 & 0 \end{bmatrix} \mathbf{u}''^m \\ = \begin{bmatrix} (v_{sp2S1} - v_{sp2F1}) & (v_{sp2S2} - v_{sp2F2}) & (v_{sp2S3} - v_{sp2F3}) & (v_{sp2S4} - v_{sp2F4}) \end{bmatrix} \mathbf{u}^n, \end{aligned} \quad (12)$$

where $\mathbf{u}^m = [u_1^n, u_2^n, u_3^n, u_4^n]^T$; $\mathbf{u}''^m = [u_1''^m, u_2''^m, u_3''^m, u_4''^m]^T$; and $\mathbf{u}^n = [u_1^n, u_2^n, u_3^n, u_4^n]^T$.

In a similar manner, one is able to calculate the second derivative at the boundary node $\eta_{n\eta}$.

2.5 Matrix assembly for first and second derivative expressions

The IRBF system on a grid line for the first derivative is obtained by letting the interior node taking value from 2 to $(n_\eta - 1)$ in (1), (2), and (3); and, making use of (4), (5), and (6) for the boundary nodes 1 and n_η . In a similar manner, the IRBF system on a grid line for the second derivative is obtained by letting the interior node taking value from 2 to $(n_\eta - 1)$ in (7), (8), and (9); and, making use of (10), (11), and (12) for the boundary nodes 1 and n_η . The resultant matrix assembly is expressed as

$$\underbrace{\begin{bmatrix} \mathbf{A}_{1F} & \mathbf{0} \\ \mathbf{A}_{1S} & \mathbf{B}_{1S} \\ \mathbf{A}_{1FS} & \mathbf{B}_{1FS} \\ \mathbf{A}_{2F} & \mathbf{B}_{2F} \\ \mathbf{0} & \mathbf{B}_{2S} \\ \mathbf{A}_{2FS} & \mathbf{B}_{2FS} \end{bmatrix}}_{\text{Coefficient matrix}} \begin{bmatrix} \mathbf{u}^m \\ \mathbf{u}''' \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{1F} \\ \mathbf{R}_{1S} \\ \mathbf{R}_{1FS} \\ \mathbf{R}_{2F} \\ \mathbf{R}_{2S} \\ \mathbf{R}_{2FS} \end{bmatrix} \mathbf{u}^n, \tag{13}$$

where \mathbf{A}_{1F} , \mathbf{A}_{1S} , \mathbf{B}_{1S} , \mathbf{A}_{1FS} , \mathbf{B}_{1FS} , \mathbf{A}_{2F} , \mathbf{B}_{2F} , \mathbf{B}_{2S} , \mathbf{A}_{2FS} , \mathbf{B}_{2FS} , and $\mathbf{0}$ are $n_\eta \times n_\eta$ matrices; $\mathbf{u}^m = \{u_1^m, u_2^m, \dots, u_{n_\eta}^m\}^T$; $\mathbf{u}''' = \{u_1''', u_2''', \dots, u_{n_\eta}'''\}^T$; and $\mathbf{u}^n = \{u_1^n, u_2^n, \dots, u_{n_\eta}^n\}^T$. The coefficient matrix is sparse with diagonal, bi-diagonal, and tri-diagonal sub-matrices. Solving (13) yields

$$\mathbf{u}^m = \mathbf{D}_\eta \mathbf{u}^n, \tag{14}$$

$$\mathbf{u}''' = \mathbf{D}_{\eta\eta} \mathbf{u}^n, \tag{15}$$

where \mathbf{D}_η and $\mathbf{D}_{\eta\eta}$ are $n_\eta \times n_\eta$ matrices. The approximations of the first and second derivatives, \mathbf{u}' and \mathbf{u}'' , respectively, are will be used in the following sections.

3 Domain decomposition preconditioners

This paper presents the implementation of the Schwarz domain decomposition (DD) preconditioned GMRES techniques using the CCIRBF approximation scheme for the convergence analysis for Poisson problems. In order to describe the working principles of the Schwarz DD methods, we will first reintroduce the classical alternating and parallel Schwarz algorithms for Poisson problems as below. For more information about the Schwarz DD methods, readers are referred to the literature in [Smith, Bjorstad, and Gropp (1996); Danaila (2007)].

3.1 Classical Schwarz

In one dimension, the Poisson problem is expressed as follows.

$$\begin{cases} -u''(x) = f(x) \text{ for } x \in (a, b), \\ u(a) = u_a, \\ u(b) = u_b, \end{cases} \tag{16}$$

where u'' are the operators of the approximation of the second derivative; f are given right hand side values; u are solutions; and u_a and u_b are boundary values. The computational interval $[a, b]$ is discretised on $n + 2$ points $x_i = a + ih$ for $i = 0, \dots, n + 1$ with a uniform step $h = \frac{b-a}{n+1}$. For simplicity, we decompose the computational interval $[a, b]$ into two subintervals with overlapping: we choose odd value n and two integer values i_l and i_r symmetric with respect to $\frac{n+1}{2}$ such that $i_l < \frac{n+1}{2} < i_r$. We set $x_l = i_l h$ and $x_r = i_r h$, thus defining two intervals $[a, x_r]$ and $[x_l, b]$ with a nonempty overlap $[a, x_r] \cap [x_l, b] = [x_l, x_r] = \Omega_0 \neq \emptyset$. The problem domain and subdomains become $\Omega = (a, b)$, $\Omega_1 = (a, x_r)$, and $\Omega_2 = (x_l, b)$, respectively. It is noted that the domains Ω , Ω_1 and Ω_2 do not include their boundaries and $\cup_{i=1}^2 \Omega_i = \Omega$.

We now compute the solution u to the problem (16) by solving two problems on subintervals $[a, x_r]$ and $[x_l, b]$. The solution u_1 (respectively u_2) is expected to be the restriction on the $[a, x_r]$ (respectively $[x_l, b]$) of the solution u to the problem on the full interval $[a, b]$. The two solutions u_1 and u_2 must therefore be identical within the overlapping region $[x_l, x_r]$, which allows us to define the boundary conditions in x_l and x_r

$$u_1(x_r) = \beta = u_2(x_r) \text{ and } u_1(x_l) = \alpha = u_2(x_l), \tag{17}$$

Initially, the values of α and β are "guessed" by linear interpolation of the global boundary conditions

$$\alpha = \frac{1}{b-a} \{u_a(b-x_l) + u_b(x_l-a)\}, \tag{18}$$

$$\beta = \frac{1}{b-a} \{u_a(b-x_r) + u_b(x_r-a)\}. \tag{19}$$

3.1.1 Classical Alternating Schwarz

In alternating Schwarz method, a sequence (u_1^n, u_2^n) for $n \geq 0$ is built by solving alternatively the same equations (16) in $[a, x_r]$ and $[x_l, b]$ with the values on the boundary defined by the previously computed values in the other subdomain. The

alternating Schwarz method begins by selecting an initial guess $u_2^0(x_r) = \beta$. Then, iteratively for $n = 1, 2, 3, \dots$, one solves the boundary value problem

$$[a, x_r] \begin{cases} -u_1''^n(x) = f(x) \text{ for } x \in \Omega_1 = (a, x_r), \\ u_1^n(a) = u_a, \\ u_1^n(x_r) = u_2^{n-1}(x_r), \end{cases} \quad (20)$$

for solution u_1^n . This is followed by the solution of the boundary value problem

$$[x_l, b] \begin{cases} -u_2''^n(x) = f(x) \text{ for } x \in \Omega_2 = (x_l, b), \\ u_2^n(x_l) = u_1^n(x_l), \\ u_2^n(b) = u_b, \end{cases} \quad (21)$$

for solution u_2^n .

3.1.2 Classical Parallel Schwarz

In parallel Schwarz method, we set $u_1^0(x_l) = \alpha$ and $u_2^0(x_r) = \beta$. The computations in $[a, x_r]$ and $[x_l, b]$ are made in parallel

$$[a, x_r] \begin{cases} -u_1''^n(x) = f(x) \text{ for } x \in (a, x_r), \\ u_1^n(a) = u_a, \\ u_1^n(x_r) = u_2^{n-1}(x_r), \end{cases} \quad (22)$$

and

$$[x_l, b] \begin{cases} -u_2''^n(x) = f(x) \text{ for } x \in (x_l, b), \\ u_2^n(x_l) = u_1^{n-1}(x_l), \\ u_2^n(b) = u_b, \end{cases} \quad (23)$$

for solution u_1^n and u_2^n .

The classical algorithms described above can be modified to get additive, multiplicative and hybrid DD preconditioners used in a Krylov subspace solver such as GMRES. For convenience, we only summarise those DD preconditioners in this paper as below. For more details, readers refer to [Smith, Bjorstad, and Gropp (1996)].

3.2 Additive, multiplicative and hybrid Schwarz preconditioner

Figure 3 illustrates the decomposition of the global domain Ω into two subdomains Ω_i , where Ω_i are overlapping subdomains. We define the restriction map \mathbf{R}_i from global domain Ω to subdomain Ω_i as follows.

$$\mathbf{R}_1 = [\mathbf{I}_{\Omega_1} \quad \mathbf{0}], \quad (24)$$

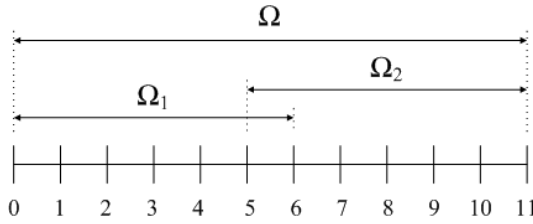


Figure 3: An example of decomposition of a domain into two subdomains.

$$\mathbf{R}_2 = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{\Omega_2} \end{bmatrix}, \tag{25}$$

where \mathbf{I} are identity matrices. Then, subdomain matrix is defined as

$$\mathbf{A}_i = \mathbf{R}_i \mathbf{A} \mathbf{R}_i^T, \tag{26}$$

where \mathbf{A} is the problem system matrix and \mathbf{R}_i^T is the interpolation map from global domain Ω to subdomain Ω_i . For general description, we assume that the global domain Ω is divided into q -subdomains, where $q \geq 2$.

3.2.1 One-level additive Schwarz preconditioner

The one-level additive Schwarz preconditioner is simply formulated as

Algorithm 1 : one-level additive Schwarz preconditioner

$$\mathbf{v} \leftarrow \sum_{i=1}^q \mathbf{B}_i \mathbf{r}, \tag{27}$$

where \mathbf{r} is the residual; $\mathbf{B}_i = \mathbf{R}_i^T \mathbf{A}_i^{-1} \mathbf{R}_i$ restricts the residual \mathbf{r} to subdomain Ω_i ; and \mathbf{v} is Krylov vector in the GMRES algorithm.

3.2.2 One-level multiplicative Schwarz preconditioner

One-level multiplicative Schwarz preconditioner, the sequential version of the one-level additive Schwarz preconditioner, is expressed as follows.

Algorithm 2 : one-level multiplicative Schwarz preconditioner

$$\mathbf{v} \leftarrow \mathbf{B}_1 \mathbf{r}, \tag{28}$$

$$\mathbf{v} \leftarrow \mathbf{v} + \mathbf{B}_2(\mathbf{r} - \mathbf{A}\mathbf{v}), \tag{29}$$

...

$$\mathbf{v} \leftarrow \mathbf{v} + \mathbf{B}_q(\mathbf{r} - \mathbf{A}\mathbf{v}). \tag{30}$$

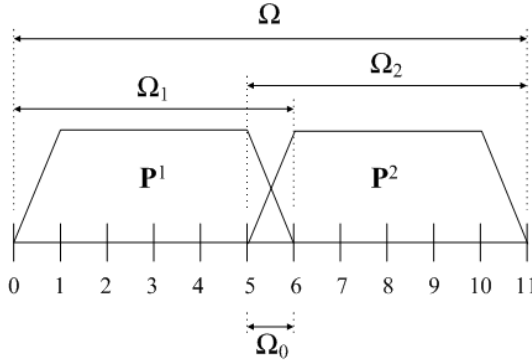


Figure 4: An example of discretisation of two coarse meshes.

Partition of unity coarse meshes for two-level algorithms : In the two-level methods, coarse meshes need to be constructed. We define the coarse mesh on the existing fine mesh [Jenkins, Kees, Kelley, and Miller (2001)]. By this way, we do not need to create the coarse mesh geometry or use the geometric information about subdomains. Figure 4 shows the discretisation of a fine mesh into two coarse meshes with one coarse mesh map per subdomain Ω_i , where $\Omega_0 = \Omega_1 \cap \Omega_2$ is an overlapping region. We use a partition of unity (PU), i.e. to sum up to one everywhere in the domain of calculation. We let \mathbf{P}^i be a PU subordinate to the covering partition Ω_i of Ω with the following conditions

$$\begin{cases} \mathbf{P}^1 + \mathbf{P}^2 = \mathbf{1}, \\ \mathbf{0} \leq \mathbf{P}^1, \mathbf{P}^2 \leq \mathbf{1}, \\ \mathbf{P}^1 \equiv \mathbf{1} \text{ on } \Omega_1 \setminus \Omega_0 \text{ and } \mathbf{P}^1 \equiv \mathbf{0} \text{ on } \Omega_2 \setminus \Omega_0, \\ \mathbf{P}^2 \equiv \mathbf{1} \text{ on } \Omega_2 \setminus \Omega_0 \text{ and } \mathbf{P}^2 \equiv \mathbf{0} \text{ on } \Omega_1 \setminus \Omega_0. \end{cases} \quad (31)$$

Similarly to the one-level, we will choose \mathbf{R}_0 , a coarse mesh restriction map from fine to coarse meshes, such that it has the form

$$\mathbf{R}_0 = \begin{bmatrix} \mathbf{W}^1 \\ \mathbf{W}^2 \end{bmatrix}, \quad (32)$$

where \mathbf{W}^1 and \mathbf{W}^2 are defined as

$$\mathbf{W}_1 = \mathbf{P}^1 / \|\mathbf{P}^1\|_2 \text{ and } \mathbf{W}_2 = \mathbf{P}^2 / \|\mathbf{P}^2\|_2, \quad (33)$$

$\|\cdot\|_2$ denotes Euclidean norm. A Galerkin or variational coarse grid correction uses the fine grid matrix to obtain the coarse mesh matrix as follows.

$$\mathbf{A}_0 = \mathbf{R}_0 \mathbf{A} \mathbf{R}_0^T, \tag{34}$$

where \mathbf{R}_0^T is a coarse mesh interpolation map from fine to coarse meshes. For general description, we again assume that the global domain Ω is divided into q -subdomains, where $q \geq 2$.

3.2.3 Two-level additive Schwarz preconditioner

The two-level additive Schwarz is formed by adding the coarse mesh problem to the one-level additive problem

Algorithm 3 : two-level additive Schwarz preconditioner

$$\mathbf{v} \leftarrow \left(\mathbf{R}_0^T \mathbf{A}_0^{-1} \mathbf{R}_0 + \sum_{i=1}^q \mathbf{B}_i \right) \mathbf{r}. \tag{35}$$

3.2.4 Two-level multiplicative Schwarz preconditioner

The two-level multiplicative Schwarz is formed by adding the coarse mesh problem to the one-level multiplicative as follows. It is noted that the coarse mesh problem is solved only once at the beginning of calculation.

Algorithm 4 : two-level multiplicative Schwarz preconditioner

$$\mathbf{v} \leftarrow \mathbf{R}_0^T \mathbf{A}_0^{-1} \mathbf{R}_0 \mathbf{r}, \tag{36}$$

$$\mathbf{v} \leftarrow \mathbf{v} + \mathbf{B}_1 (\mathbf{r} - \mathbf{A} \mathbf{v}), \tag{37}$$

...

$$\mathbf{v} \leftarrow \mathbf{v} + \mathbf{B}_q (\mathbf{r} - \mathbf{A} \mathbf{v}). \tag{38}$$

3.2.5 Two-level hybrid I Schwarz preconditioner

The two-level hybrid I is formulated on the basis of the one-level multiplicative by adding the coarse mesh problem to its last stage

Algorithm 5 : two-level hybrid I Schwarz preconditioner

$$\mathbf{v} \leftarrow \mathbf{B}_1 \mathbf{r}, \tag{39}$$

$$\mathbf{v} \leftarrow \mathbf{v} + \mathbf{B}_2 (\mathbf{r} - \mathbf{A} \mathbf{v}), \tag{40}$$

...

$$\mathbf{v} \leftarrow \mathbf{v} + \mathbf{B}_q (\mathbf{r} - \mathbf{A} \mathbf{v}). \tag{41}$$

$$\mathbf{v} \leftarrow \mathbf{v} + \mathbf{R}_0^T \mathbf{A}_0^{-1} \mathbf{R}_0 \mathbf{r}. \tag{42}$$

3.2.6 Two-level hybrid II Schwarz preconditioner

The two-level hybrid II is formulated on the basis of the one-level additive by adding the coarse mesh problem to its last stage

Algorithm 6 : two-level hybrid II Schwarz preconditioner

$$\mathbf{v} \leftarrow \sum_{i=1}^q \mathbf{B}_i \mathbf{r}, \tag{43}$$

$$\mathbf{v} \leftarrow \mathbf{v} + \mathbf{R}_0^T \mathbf{A}_0^{-1} \mathbf{R}_0 (\mathbf{r} - \mathbf{A}\mathbf{v}). \tag{44}$$

4 GMRES

We utilise a generalised minimal residual algorithm (GMRES) for solving non symmetric linear systems. More information about GMRES may be found in [Saad and Schultz (1986); Behr (2014); Strang (2007)]. Consider the linear system

$$\mathbf{A}\mathbf{u} = \mathbf{f}, \tag{45}$$

where \mathbf{f} are given right hand side values; \mathbf{u} are unknowns; and \mathbf{A} is the problem system matrix. The GMRES algorithm is outlined in Table 1, where *Algorithm★* is used to represent one of the six preconditioning algorithms mentioned above and ϵ is the GMRES tolerance.

5 Serial and parallel two-level multiplicative Schwarz DD methods

For numerical examples, we incorporate the CCIRBF into serial and parallel two-level multiplicative Schwarz DD methods to solve the linear system (45) as follows.

5.1 Serial two-level multiplicative Schwarz

First, solve the coarse mesh problem once at the beginning

$$\mathbf{u} \leftarrow \mathbf{B}_0 (\mathbf{f} - \mathbf{A}\mathbf{u}), \tag{46}$$

Then, solve the fine mesh problem

$$\mathbf{u} \leftarrow \mathbf{u} + \mathbf{B}_1 (\mathbf{f} - \mathbf{A}\mathbf{u}), \tag{47}$$

...

$$\mathbf{u} \leftarrow \mathbf{u} + \mathbf{B}_q (\mathbf{f} - \mathbf{A}\mathbf{u}), \tag{48}$$

where $\mathbf{B}_i = \mathbf{R}_i^T \mathbf{A}_i^{-1} \mathbf{R}_i$.

Table 1: GMRES Algorithm.

<u>Initialisation</u>	
$\mathbf{r}_0 \leftarrow \mathbf{f} - \mathbf{A}\mathbf{u}_0$	compute initial residual
$\beta \leftarrow \ \mathbf{r}_0\ _2$	compute initial residual norm
$\mathbf{v}_1 \leftarrow \mathbf{r}_0/\beta$	define first Krylov vector
$\mathbf{g} \leftarrow \beta\mathbf{e}_1$	initialise right hand side
<u>Iteration</u>	
1. for $j = 1, 2, \dots, k, \dots$, until satisfied do	
2. $\mathbf{v}_{j+1} \leftarrow \text{Algorithm}\star(\mathbf{v}_j)$	preconditioning step
3. $\mathbf{v}_{j+1} \leftarrow \mathbf{A}\mathbf{v}_{j+1}$	matrix-vector product
4. for $i = 1, 2, \dots, j$	modified Gramm-Schmidt orthogonalisation
5. $h_{i,j} \leftarrow (\mathbf{v}_i, \mathbf{v}_{j+1})$	
6. $\mathbf{v}_{j+1} \leftarrow \mathbf{v}_{j+1} - h_{i,j}\mathbf{v}_i$	
7. $h_{j+1,j} \leftarrow \ \mathbf{v}_{j+1}\ _2$	
8. $\mathbf{v}_{j+1} \leftarrow \mathbf{v}_{j+1}/h_{j+1,j}$	define next Krylov vector
9. for $i = 1, 2, \dots, j$	previous Givens rotation on \mathbf{H}
10. $\begin{cases} h_{i,j} \leftarrow c_i h_{i,j} + s_i h_{i+1,j} \\ h_{i+1,j} \leftarrow -s_i h_{i,j} + c_i h_{i+1,j} \end{cases}$	
11. $\gamma \leftarrow \sqrt{h_{j,j}^2 + h_{j+1,j}^2}$	compute next Givens rotation
12. $c_j \leftarrow h_{j,j}/\gamma$; $s_j \leftarrow h_{j+1,j}/\gamma$	
13. $\begin{cases} h_{j,j} \leftarrow \gamma \\ h_{j+1,j} \leftarrow 0 \end{cases}$	Givens rotation on \mathbf{H}
14. $\begin{cases} g_j \leftarrow c_j g_j \\ g_{j+1} \leftarrow -s_j g_j \end{cases}$	Givens rotation on \mathbf{g}
15. if $ g_{j+1} \leq \varepsilon$ exit loop	loop convergence check
<u>Form approximate solution</u>	
$\begin{Bmatrix} y_1 \\ \vdots \\ y_j \end{Bmatrix} \leftarrow \begin{bmatrix} h_{1,1} & \cdots & h_{1,j} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & h_{j,j} \end{bmatrix}^{-1} \begin{Bmatrix} g_1 \\ \vdots \\ g_j \end{Bmatrix}$	back substitution
$\mathbf{u}_k \leftarrow \mathbf{u}_0 + \mathbf{V}_k \mathbf{y}_k$	form approximate solution
Note: <i>Algorithm</i> ★ is used to represent one of the six preconditioning algorithms mentioned in Section 8.1.	

5.2 Parallel two-level multiplicative Schwarz

The serial two-level multiplicative Schwarz method mentioned above has very little potential for parallelism. This is easily fixed. It is noted that there are often many subdomains which share no common grid points as shown in Figure 5. The numerical solution on these subdomains, therefore, could be updated simultaneously, in parallel.

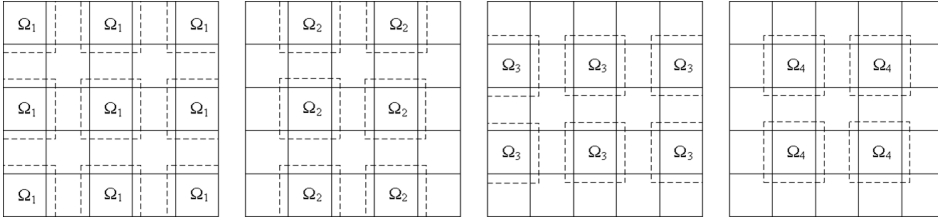


Figure 5: Colouring of 5×5 subdomains into four classes.

Define a colouring of the subdomains in the way described in [Smith, Bjorstad, and Gropp (1996)]. For each subdomain, we associate a colour in the way that no two subdomains sharing common grid points have the same colour. Let i be the number of colours used.

In this paper, we introduce the coarse mesh problem into the original colouring technique of Smith, Bjorstad, and Gropp (1996). We can now generate a i -step method as follows.

First, solve the coarse mesh problem once at the beginning

$$\mathbf{u}^n \leftarrow \mathbf{B}_0 (\mathbf{f} - \mathbf{A}\mathbf{u}^n), \tag{49}$$

Then, solve the fine mesh problem

$$\mathbf{u}^{n+1/i} \leftarrow \mathbf{u}^n + \sum_{i \in \Omega_1} \mathbf{B}_i (\mathbf{f} - \mathbf{A}\mathbf{u}^n), \tag{50}$$

$$\mathbf{u}^{n+2/i} \leftarrow \mathbf{u}^{n+1/i} + \sum_{i \in \Omega_2} \mathbf{B}_i (\mathbf{f} - \mathbf{A}\mathbf{u}^{n+1/i}), \tag{51}$$

...

$$\mathbf{u}^{n+1} \leftarrow \mathbf{u}^{n+(i-1)/i} + \sum_{i \in \Omega_4} \mathbf{B}_i (\mathbf{f} - \mathbf{A}\mathbf{u}^{n+(i-1)/i}), \tag{52}$$

where $(n + 1)$ is the current time level.

6 Parallelism

The parallel implementation is based on the colouring technique explained in Section 5.2. As shown in Figure 5, four colours are used to mark colours of subdomains so that subdomains with the same colour do not overlap each other. In each sub-step from (50) to (52), one colour is considered, each CPU is assigned to solve the problem in each subdomain within that colour. Then, the result from subdomains will be exchanged between themselves in order for each subdomain to obtain a unique copy of the whole domain solution. In next substep, the next colour is considered until the convergence measurement reaches a predefined value.

In this implementation, the broadcast communication is used because each subdomain needs to send information to and receive it from all other subdomains. As whole domain solution is kept in each subdomain and is updated after each substep, the convergence measurement calculated in each subdomain is consistent with all other subdomains. This ensures the concurrent convergence of subdomains and thus alleviate the need of a dedicated termination algorithm for the whole system.

7 Stream function-vorticity formulation

The transient Navier-Stokes equations for an incompressible viscous fluid in the stream function-vorticity formulation are expressed in the dimensionless conservative forms as follows.

$$\frac{\partial \omega}{\partial t} + \frac{\partial(u\omega)}{\partial x} + \frac{\partial(v\omega)}{\partial y} = \frac{1}{Re} \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \omega, \quad (53)$$

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \psi = -\omega. \quad (54)$$

It is well known that equations in conservative form generally produce more accurate results compared to those of non-conservative form [Niyogi, Chakrabarty, and Laha (2009)]. In equations (53) and (54), ψ is the stream function; ω is the vorticity; $Re = Ul/\nu$ is the Reynolds number, in which ν , l and U are the kinematic viscosity, characteristic length and characteristic speed of the flow, respectively; u and v , velocity components in x - and y -direction, are given by

$$u = \frac{\partial \psi}{\partial y} \quad \text{and} \quad v = -\frac{\partial \psi}{\partial x}. \quad (55)$$

At current time level, n , stream function equation is expressed as

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \psi^n = -\omega^{n-1}, \quad (56)$$

And velocities are expressed as

$$u^n = \frac{\partial \psi^n}{\partial y} \quad \text{and} \quad v^n = -\frac{\partial \psi^n}{\partial x}. \tag{57}$$

The temporal discretisation of (53) results in

$$\frac{\omega^n - \omega^{n-1}}{\partial t} = \frac{1}{Re} \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \omega^{n-1} - \frac{\partial(u^n \omega^{n-1})}{\partial x} - \frac{\partial(v^n \omega^{n-1})}{\partial y}. \tag{58}$$

8 Numerical examples

We chose the MQ function as the basis function in the present calculations

$$G_i(x) = \sqrt{(x - c_i)^2 + a_i^2}, \tag{59}$$

where c_i and a_i are the centre and the width of the i -th MQ, respectively. For each stencil, the set of nodal points is taken to be the same as the set of MQ centres. We simply choose the MQ width as $a_i = \beta h_i$, where β is a positive scalar and h_i is the distance between the i -th node and its closest neighbour.

Measurement Criteria: We evaluate the performance of the present schemes through the following measures

- i. the root mean square error (*RMS*) is defined as

$$RMS = \sqrt{\frac{\sum_{i=1}^N (f_i - \bar{f}_i)^2}{N}}, \tag{60}$$

where f_i and \bar{f}_i are the computed and exact values of the solution f at the i -th node, respectively; and, N is the number of nodes over the whole domain.

- ii. the average absolute error (L_1) is defined as

$$L_1 = \frac{1}{N} \sum_{i=1}^N |f_i - \bar{f}_i|. \tag{61}$$

- iii. the global convergence rate with respect to the grid refinement is defined through

$$RMS(h) \approx \gamma h^\alpha = O(h^\alpha), \tag{62}$$

where h is the grid size; and γ and α are exponential model's parameters.

iv. a flow is considered to reach its steady state when

$$\sqrt{\frac{\sum_{i=1}^N (f_i^n - f_i^{n-1})^2}{N}} < 10^{-8}. \quad (63)$$

v. Speed-up, S , and efficiency, E are defined as

$$S = \frac{T_s}{T_p}, \quad (64)$$

$$E = \frac{S}{p} \times 100, \quad (65)$$

where T_s is computation time on single CPU; T_p is computation time on parallel CPUs; and p is number of parallel CPUs. In particular, T_s is defined as the computation time of the CCIRBF-Single domain in this paper.

Subdomain partition: Referring the subdomain partition presented in [Jenkins, Kees, Kelley, and Miller (2001)], we let $h = 2^{-m}$ be the scale of the fine mesh and let the overlap o be the nearest integer larger than

$$2^m o_1, \quad (66)$$

where o_1 is the overlap that depends on the physical subdomain. For examples, overlap of 1% is determined by letting $o_1 = 0.01$. The global grid is an $n \times n$ mesh where $n = 2^m + o$. We will use 2^p subdivisions in each directions so there will be 2^{p+1} subdomains, each of size $m \times m$, where

$$m = 2^{m-p} + o - 1. \quad (67)$$

The scale H of the subdomains is defined as 2^{-p} . This way of partition allows for a perfect split with all intervals having equal length. Figure 6 illustrates an example of decomposition of the 2D domain Ω .

In this work, calculations are done with a Dell computer, Precision T7600. Its specifications are Intel(R) Xeon(R) CPU E5-2687W 0 3.10 GHz 3.10 GHz (2 processors), memory(RAM) of 128GB and 64-bit operating system. The Matlab(R) version 2012 is utilised. In serial and parallel algorithms, the overlapping is chosen between 1% to 25%. In parallel computing, the percentage of communication time is calculated with respect to its total computation time.

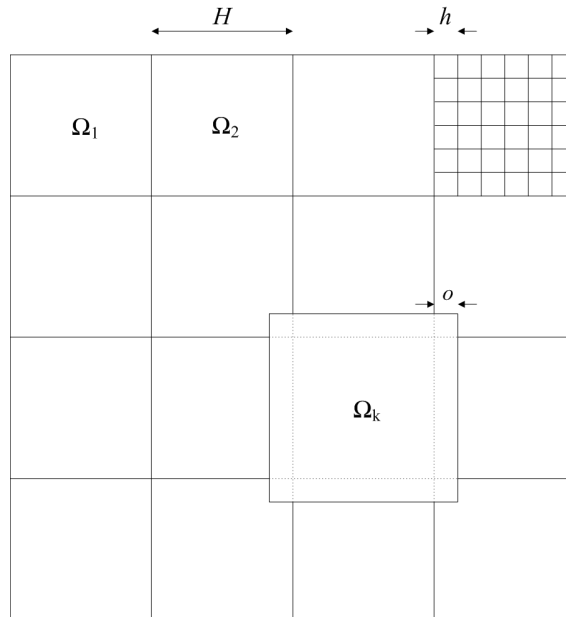


Figure 6: An example of decomposition of Ω into subdomains $\Omega_1, \Omega_2, \dots, \Omega_k, \dots$

8.1 Convergence analysis of CCIRBF based additive / multiplicative / hybrid Schwarz for 2D Poisson

We now apply the GMRES algorithms described in Section 4 for the 2D case. The 2D Poisson problem becomes $-(u''_{xx} + u''_{yy}) = f$. We consider the right hand side f equal to $-2\pi^2 \sin(\pi x) \sin(\pi y)$ and the solution is required to be zero on the boundary of $[0, 1]^2$. Calculations are carried out on coarse meshes of size $H = 1/4, 1/8$ and $1/16$ and fine meshes of size roughly $h = 1/32, 1/64$ and $1/128$. The value of $\beta = 50$ is simply chosen. We terminate calculations when the GMRES residual is smaller than 0.01. We tabulate iteration counts upon termination. In Tables 2-7, H is decreased by a factor of two going down the columns and h is similarly decreased going across the rows. We increase the overlap size as $\{1\%, 5\%, 10\%, 15\%, 20\%, 25\%\}$. We consider calculations having the number of iteration larger than 100 to be not stable and unlikely to converge. For plots of iteration count versus overlap percentage and of GMRES residual versus iteration count, we deliberately choose the case where $H = 1/8$ and $h = 1/128$ to be a representative case for each overlap case because calculations with $H = 1/8$ and $h = 1/128$ are reasonably stable in our experiments.

8.1.1 One-level additive Schwarz preconditioner

Table 2 shows the iteration counts of the one-level additive preconditioned GMRES using the present CCIRBF. More details about convergence characteristics of the present CCIRBF one-level additive algorithm are shown in Figure 7. It appears that the present CCIRBF one-level additive algorithm performs best with an overlap around 20%.

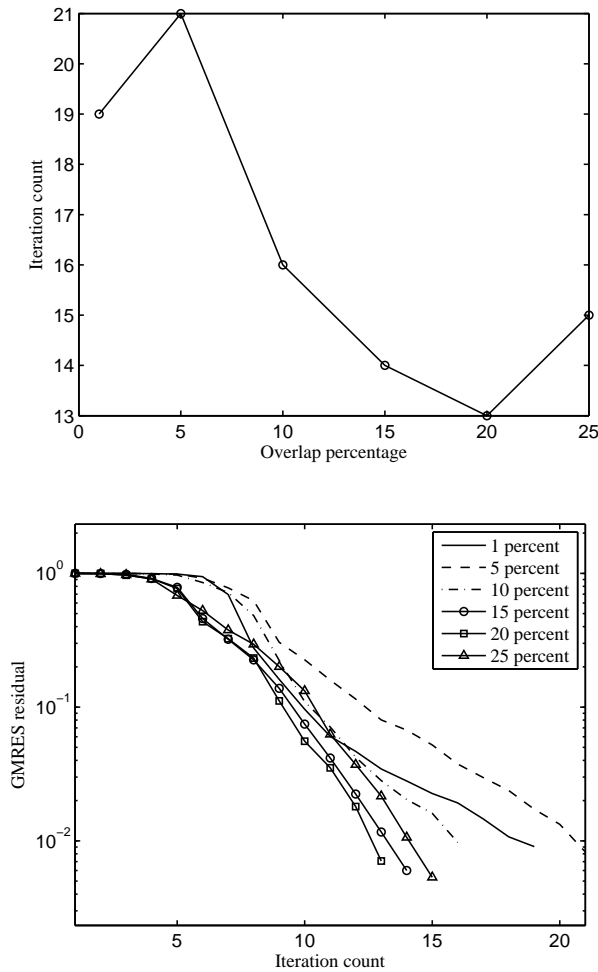


Figure 7: One-level additive preconditioned GMRES using the present CCIRBF: Iteration count versus overlap percentage (top) and GMRES residual versus iteration count (bottom).

Table 2: One-level additive preconditioned GMRES using the present CCIRBF: number of iteration required to achieve convergence.

	present CCIRBF		
$H \setminus h$	1/32	1/64	1/128
	overlap = 1%		
1/4	17	23	20
1/8	23	29	19
1/16	32	39	18
	overlap = 5%		
1/4	12	13	18
1/8	13	13	21
1/16	11	13	26
	overlap = 10%		
1/4	10	14	14
1/8	10	17	16
1/16	13	16	18
	overlap = 15%		
1/4	12	10	10
1/8	15	13	14
1/16	18	16	17
	overlap = 20%		
1/4	12	12	9
1/8	15	16	13
1/16	23	23	19
	overlap = 25%		
1/4	9	10	11
1/8	13	14	15
1/16	20	21	21

8.1.2 One-level multiplicative Schwarz preconditioner

The iteration counts of the one-level multiplicative preconditioned GMRES using the present CCIRBF are provided in Table 3. The iteration counts of the present CCIRBF one-level multiplicative algorithm are much smaller than those of the present CCIRBF one-level additive algorithm tabulated in Tables 2, except the case of 1% overlap. Especially, with cases where $H = 1/16$, the iteration counts of the present CCIRBF one-level multiplicative algorithm are much smaller than those figures of the present CCIRBF one-level additive algorithm in Tables 2. Plots of iteration count versus overlap percentage and GMRES residual versus iteration count are illustrated in Figure 8. It can be seen that the present CCIRBF one-level multiplicative algorithm performs best with the overlap between 15% and 20%.

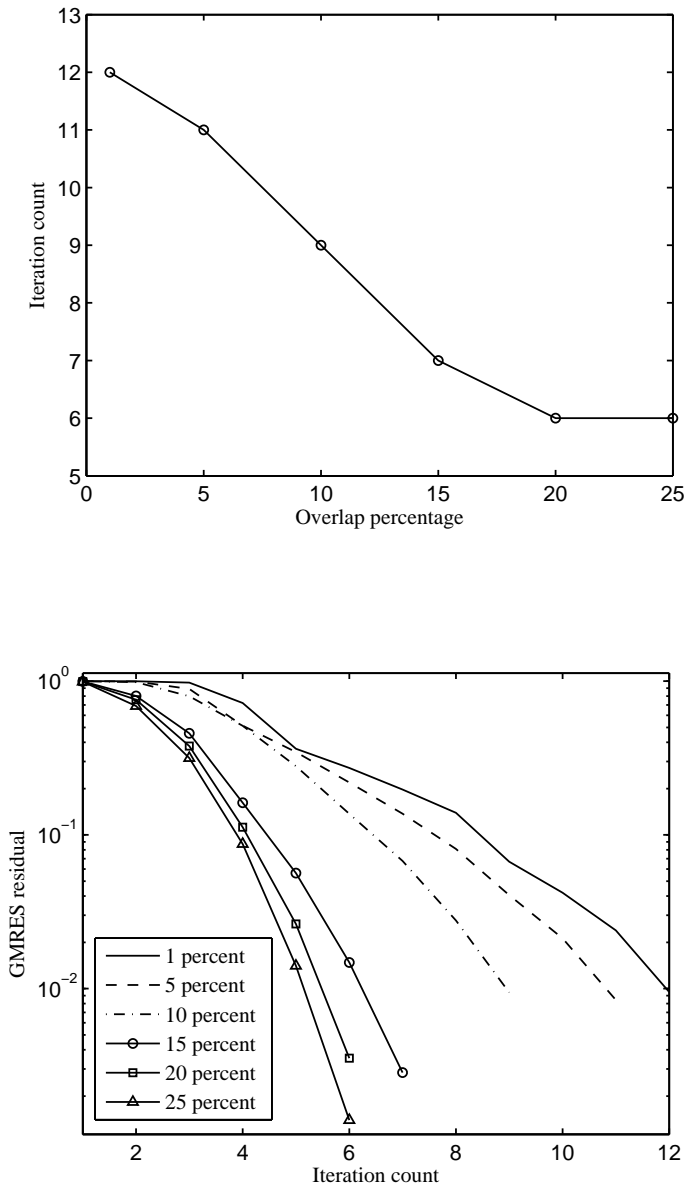


Figure 8: One-level multiplicative preconditioned GMRES using the present CCIRBF: Iteration count versus overlap percentage (top) and GMRES residual versus iteration count (bottom).

Table 3: One-level multiplicative preconditioned GMRES using the present CCIRBF: number of iteration required to achieve convergence.

	present CCIRBF		
$H \setminus h$	1/32	1/64	1/128
	overlap = 1%		
1/4	51	25	11
1/8	19	95	12
1/16	19	34	17
	overlap = 5%		
1/4	8	7	8
1/8	9	9	11
1/16	9	10	16
	overlap = 10%		
1/4	6	7	7
1/8	7	9	9
1/16	7	12	12
	overlap = 15%		
1/4	6	6	6
1/8	8	6	7
1/16	12	7	8
	overlap = 20%		
1/4	6	6	6
1/8	7	7	6
1/16	8	7	7
	overlap = 25%		
1/4	5	5	5
1/8	5	5	6
1/16	5	6	6

8.1.3 Two-level additive Schwarz preconditioner

Table 4 shows the iteration counts of the two-level additive preconditioned GMRES using the present CCIRBF. The present CCIRBF two-level additive scheme is comparable with the present CCIRBF one-level additive scheme shown in Table 2. As shown in Figure 9, the present CCIRBF two-level additive scheme performs best with an overlap of around 10%.

8.1.4 Two-level multiplicative Schwarz preconditioner

Table 5 shows the iteration statistics of the two-level multiplicative preconditioned GMRES using the present CCIRBF. In comparison with the present CCIRBF one-level additive/ multiplicative and two-level additive algorithms shown in Table 2,

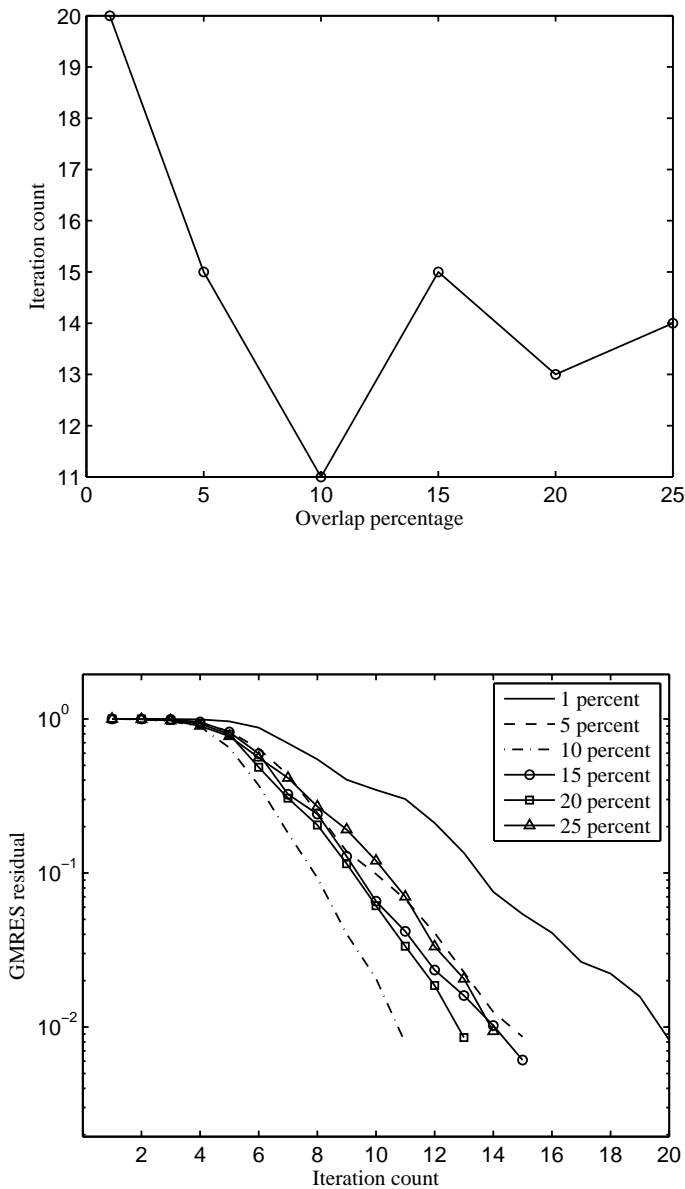


Figure 9: Two-level additive preconditioned GMRES using the present CCIRBF: Iteration count versus overlap percentage (top) and GMRES residual versus iteration count (bottom).

Table 4: Two-level additive preconditioned GMRES using the present CCIRBF: number of iteration required to achieve convergence.

	present CCIRBF		
$H \setminus h$	1/32	1/64	1/128
	overlap = 1%		
1/4	16	21	21
1/8	15	19	20
1/16	11	16	16
	overlap = 5%		
1/4	11	13	17
1/8	11	11	15
1/16	16	17	10
	overlap = 10%		
1/4	9	12	13
1/8	12	10	11
1/16	15	16	18
	overlap = 15%		
1/4	10	9	9
1/8	16	15	15
1/16	15	16	16
	overlap = 20%		
1/4	9	10	8
1/8	14	14	13
1/16	22	21	19
	overlap = 25%		
1/4	11	11	12
1/8	14	14	14
1/16	21	21	21

3 and 4, respectively, the present CCIRBF two-level multiplicative algorithm is superior with much smaller iterations. Figure 10 depicts fast convergence of the present CCIRBF two-level multiplicative algorithm for overlap from 10% up to 25%. Moreover, the smaller the overlap size, the faster the calculation, and therefore, the overlap of 10% is recommended for the present CCIRBF two-level multiplicative.

For comparison purposes, we incorporate the high order compact finite difference (HOC) of [Tian, Liang, and Yu (2011)] into the DD two-level multiplicative Schwarz preconditioned GMRES algorithm. It can be seen that the present CCIRBF two-level multiplicative scheme produces much better results than those of the HOC two-level multiplicative scheme at the overlap of 1% and 25%. For other overlap cases, the present CCIRBF two-level multiplicative algorithm is com-

parable with the HOC two-level multiplicative algorithm.

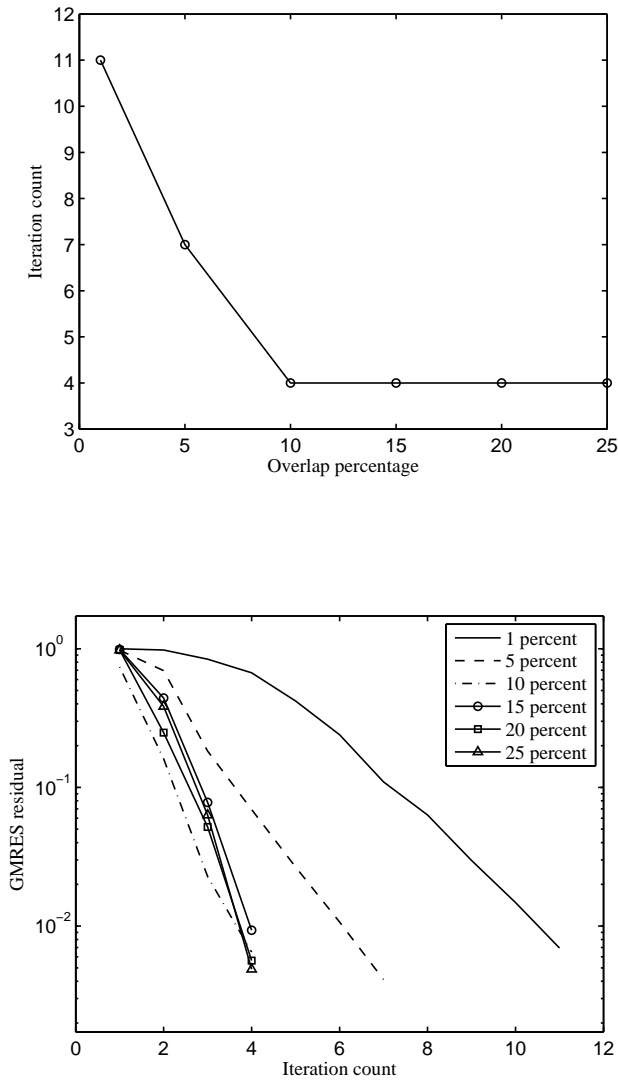


Figure 10: Two-level multiplicative preconditioned GMRES using the present CCIRBF: Iteration count versus overlap percentage (top) and GMRES residual versus iteration count (bottom).

Table 5: Two-level multiplicative preconditioned GMRES using the HOC and present CCIRBF: number of iteration required to achieve convergence.

$H \setminus h$	HOC Tian et al. (2011)			present CCIRBF		
	1/32	1/64	1/128	1/32	1/64	1/128
	overlap = 1%					
1/4	100+	100+	59	10	13	12
1/8	100+	100+	88	9	13	11
1/16	–	100+	100+	5	9	8
	overlap = 5%					
1/4	20	5	6	7	6	7
1/8	8	4	5	5	5	7
1/16	30	6	2	8	7	4
	overlap = 10%					
1/4	3	4	5	4	5	5
1/8	4	2	3	5	5	4
1/16	4	5	5	5	8	8
	overlap = 15%					
1/4	3	3	3	5	4	4
1/8	4	4	4	6	5	4
1/16	2	3	4	9	5	5
	overlap = 20%					
1/4	2	2	3	4	4	3
1/8	4	4	4	5	5	4
1/16	3	3	3	6	6	4
	overlap = 25%					
1/4	5	5	6	4	4	3
1/8	6	6	7	5	4	4
1/16	6	6	6	4	4	4

8.1.5 *Two-level hybrid I Schwarz preconditioner*

The iteration statistics of the two-level hybrid I preconditioned GMRES using the present CCIRBF are presented in Table 6. In comparison with the present CCIRBF two-level multiplicative shown in Table 5, the present CCIRBF two-level hybrid I is less effective with larger iteration counts. Plots of iteration count versus overlap percent and GMRES residual versus iteration count for the present CCIRBF two-level hybrid I are depicted in Figure 11. It can be seen that the present CCIRBF two-level hybrid I performs best around an overlap of 10%.

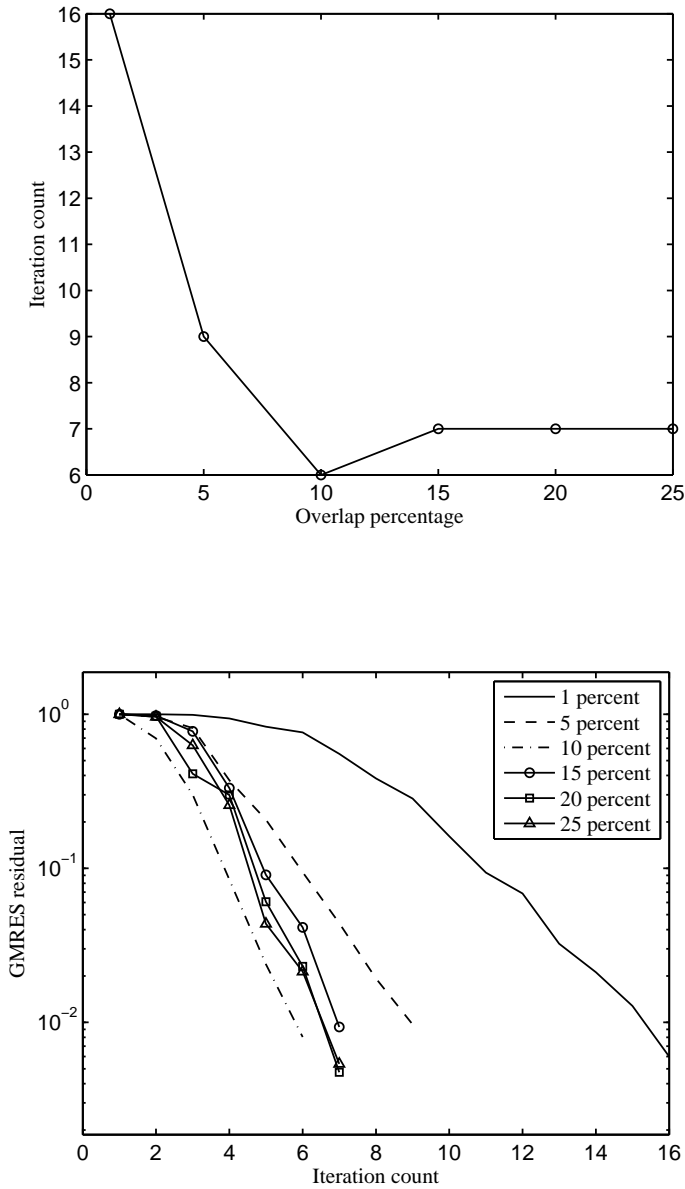


Figure 11: Two-level hybrid I preconditioned GMRES using the present CCIRBF: Iteration count versus overlap percentage (top) and GMRES residual versus iteration count (bottom).

Table 6: Two-level hybrid-I preconditioned GMRES using the present CCIRBF: number of iteration required to achieve convergence.

	present CCIRBF		
$H \setminus h$	1/32	1/64	1/128
	overlap = 1%		
1/4	15	17	14
1/8	13	28	16
1/16	6	15	14
	overlap = 5%		
1/4	9	9	10
1/8	8	8	9
1/16	10	11	6
	overlap = 10%		
1/4	6	7	8
1/8	8	6	6
1/16	8	10	11
	overlap = 15%		
1/4	8	6	6
1/8	8	7	7
1/16	12	8	8
	overlap = 20%		
1/4	7	6	6
1/8	7	7	7
1/16	9	8	7
	overlap = 25%		
1/4	6	6	6
1/8	7	7	7
1/16	7	7	7

8.1.6 Two-level hybrid II Schwarz preconditioner

Table 7 reports the iteration counts of the two-level hybrid-II preconditioned GMRES using the present CCIRBF. At the overlap of 1%, 5%, 10% and 15%, the present CCIRBF two-level hybrid-II shows better results compared to those of the present CCIRBF one- and two-level additive in Tables 2 and 4, respectively. For other overlap cases where $H = 1/16$, the iteration counts of the present CCIRBF two-level hybrid-II algorithm are much smaller than those figures of the present CCIRBF one- and two-level additive algorithm in Tables 2 and 4, respectively. Figure 12 plots the iteration count versus the overlap percent and the GMRES residual versus the iteration count from which it can be observed that the present CCIRBF two-level hybrid-II performs best around an overlap of 10%.

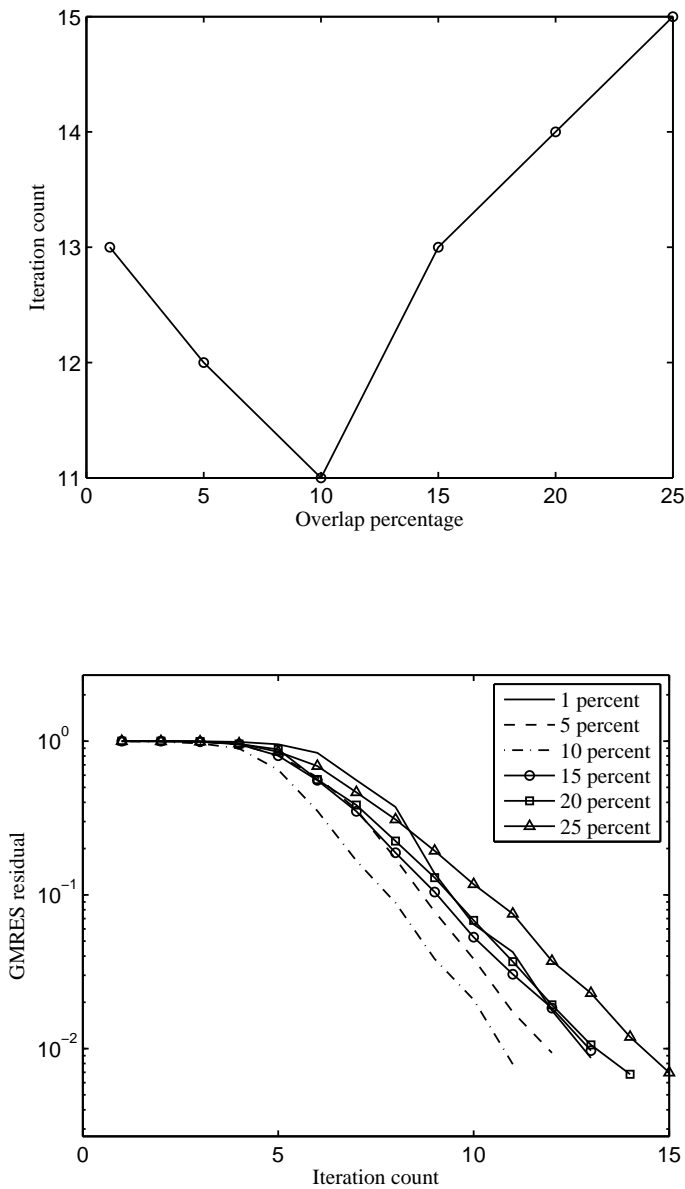


Figure 12: Two-level hybrid II preconditioned GMRES using the present CCIRBF: Iteration count versus overlap percentage (top) and GMRES residual versus iteration count (bottom).

Table 7: Two-level hybrid-II preconditioned GMRES using the present CCIRBF: number of iteration required to achieve convergence.

	present CCIRBF		
$H \setminus h$	1/32	1/64	1/128
	overlap = 1%		
1/4	12	16	16
1/8	12	14	13
1/16	11	12	12
	overlap = 5%		
1/4	10	10	13
1/8	10	10	12
1/16	14	14	10
	overlap = 10%		
1/4	9	11	12
1/8	12	10	11
1/16	15	16	16
	overlap = 15%		
1/4	10	9	9
1/8	14	13	13
1/16	15	16	17
	overlap = 20%		
1/4	10	10	8
1/8	15	15	14
1/16	19	20	18
	overlap = 25%		
1/4	11	11	12
1/8	14	15	15
1/16	18	19	19

8.1.7 Final comparison of the six algorithms using the present CCIRBF

For comparison purpose, at $H = 1/8$ and $h = 1/128$, we finally choose the overlap percentages at which each of the six algorithms using the present CCIRBF gives its best performance. Figure 13 shows the comparison of the six algorithms in terms of the GMRES residual versus the iteration count. It can be seen that the present CCIRBF two-level multiplicative at 10% overlap reaches the prescribed residual with the least iterations of only 4. After that, both the present CCIRBF one-level multiplicative at 20% overlap and the present CCIRBF two-level hybrid I at 10% overlap require 6 iterations to reach the prescribed GMRES residual. Then, both the present CCIRBF two-level additive at 10% overlap and the present CCIRBF two-level hybrid II at 10% overlap take 11 iterations to get to the target GMRES

residual.

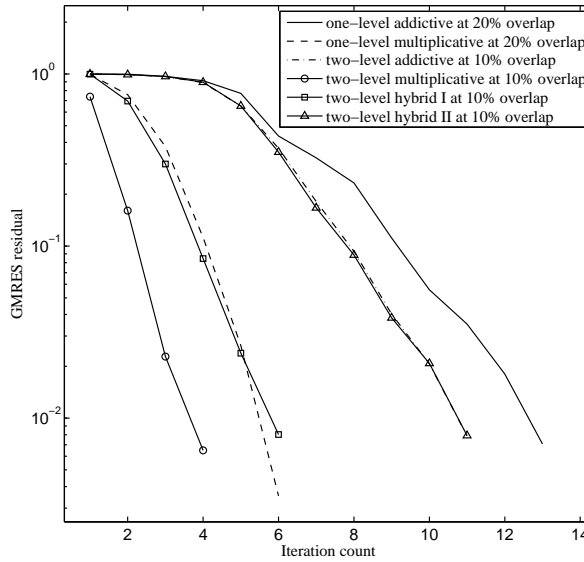


Figure 13: Comparison of the six algorithms using the present CCIRBF: GMRES residual versus iteration count.

8.2 Poisson equation

In order to study the spatial accuracy of the present CCIRBF-Serial and -Parallel algorithms, we consider the following Poisson equation

$$\frac{d^2u}{dx_1^2} + \frac{d^2u}{dx_2^2} = -18\pi^2 \sin(3\pi x_1) \sin(3\pi x_2), \tag{68}$$

subject to Dirichlet boundary condition derived from the following exact solution

$$\bar{u} = \sin(3\pi x_1) \sin(3\pi x_2), \tag{69}$$

on a square domain $[0, 1]^2$. The calculations are carried out on a set of uniform grids of $\{21 \times 21, 32 \times 32, 42 \times 42, 53 \times 53, 63 \times 63, 74 \times 74, 84 \times 84, 95 \times 95, 105 \times 105\}$. The CCIRBF-Serial and -Parallel are considered to reach its steady state when its RMS is smaller than 10^{-9} . The value of $\beta = 50$ is chosen for calculations. Table 8 illustrates the proposed CCIRBF-Serial using 2×2 subdomains and CCIRBF-Parallel using 4×4 subdomains are able to produce the same solution accuracy

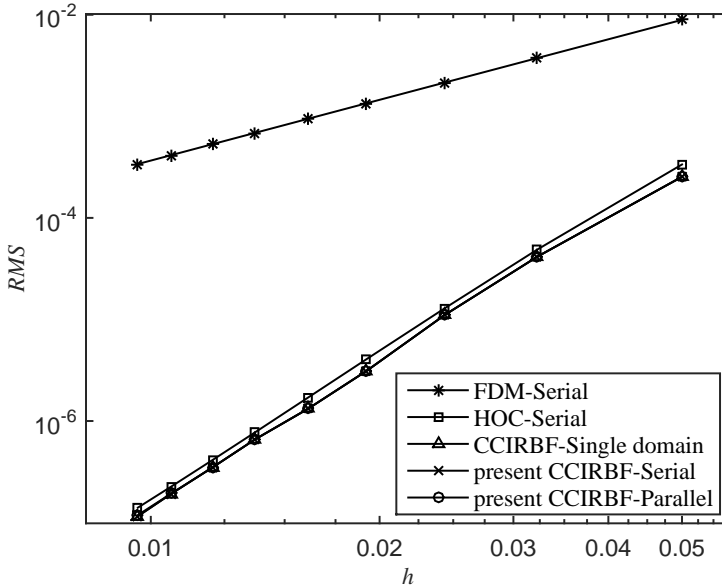


Figure 14: Poisson equation, $\beta = 50$: The effect of grid size h on the solution accuracy RMS . It is noted that the results for the CCIRBF-Single domain, CCIRBF-Serial and CCIRBF-Parallel are indistinguishable.

to those of the CCIRBF-Single domain. For comparison purposes, we incorporate the standard central FDM and the HOC of Tian, Liang, and Yu (2011) into the two-level DD. Figure 14 shows that the proposed CCIRBF-Serial and -Parallel outperform the FDM-Serial and HOC-Serial in terms of solution accuracy. The solutions converge as $O(h^{4.7})$ for the CCIRBF-Single domain, -Serial and -Parallel, $O(h^{4.8})$ for the HOC-Serial, and $O(h^{2.0})$ for the FDM-Serial.

An analysis of computational efficiency of the three algorithms, CCIRBF-Single domain, -Serial and -Parallel are illustrated in Table 9. Table 9 shows that the CCIRBF-Serial and -Parallel are generally much more efficient than the CCIRBF-Single domain. In term of computation time, the CCIRBF-Parallel generally uses less time to reach the same accuracy than the CCIRBF-Serial does. In term of efficiency, the CCIRBF-Serial is much more efficient than the CCIRBF-Parallel at low numbers of grids. However, the efficiency of the CCIRBF-Parallel increases and becomes better than that of the CCIRBF-Serial as the number of grids increases.

Table 8: Poisson equation, $\beta = 50$: The effect of grid size h on the solution accuracy RMS . LCR stands for "Local Convergence Rate".

Grid ($n_x \times n_y$)	CCIRBF-Single domain		present CCIRBF-Serial		present CCIRBF-Parallel	
	RMS	LCR	RMS	LCR	RMS	LCR
21 × 21	2.5405E-04	—	2.5405E-04	—	2.5405E-04	—
32 × 32	4.1526E-05	4.30	4.1526E-05	4.30	4.1526E-05	4.30
42 × 42	1.1130E-05	4.51	1.1130E-05	4.51	1.1130E-05	4.51
53 × 53	3.1203E-06	4.75	3.1203E-06	4.75	3.1203E-06	4.75
63 × 63	1.3301E-06	4.78	1.3301E-06	4.78	1.3301E-06	4.78
74 × 74	6.5607E-07	4.73	6.5607E-07	4.73	6.5607E-07	4.73
84 × 84	3.4805E-07	4.76	3.4805E-07	4.76	3.4805E-07	4.76
95 × 95	1.9210E-07	4.76	1.9210E-07	4.76	1.9210E-07	4.76
105 × 105	1.1548E-07	4.78	1.1548E-07	4.78	1.1548E-07	4.78

LCR= $-\log[RMS(n_x)]/RMS(21)]/\log[n_x/21]$.

Table 9: Poisson equation, $\beta = 50$: computational efficiency of the three algorithms, CCIRBF-Serial domain, -Serial and -Parallel. It is noted that "Comm." is the abbreviation of "Communication".

Grid ($n_x \times n_y$)	Methods	Subdomains	CPUs	RMS	Iterations	Comm. time Second(%)	Total time Second	Speed-up, S	Efficiency, E %
53 × 53	CCIRBF-Serial domain	—	1	3.1203e-06	—	—	6.7	—	—
	present CCIRBF-Serial	2 × 2	1	3.1203e-06	71	—	1.1	6.1	610
	present CCIRBF-Serial	4 × 4	1	3.1203e-06	125	—	3.5	1.9	190
	present CCIRBF-Parallel	4 × 4	4	3.1203e-06	321	0.4(22)	1.8	3.7	93
84 × 84	present CCIRBF-Parallel	6 × 6	9	3.1203e-06	391	1.1(23)	4.7	1.4	16
	CCIRBF-Serial domain	—	1	3.4805E-07	—	—	19.2	—	—
	present CCIRBF-Serial	2 × 2	1	3.4805E-07	73	—	6.3	3.0	300
	present CCIRBF-Serial	4 × 4	1	3.4805E-07	23	—	6.5	3.0	300
95 × 95	present CCIRBF-Parallel	4 × 4	4	3.4805E-07	75	0.3(16)	1.9	10.1	253
	present CCIRBF-Parallel	6 × 6	9	3.4805E-07	370	1.1(10)	10.5	1.8	20
	CCIRBF-Serial domain	—	1	1.9210E-07	—	—	36.0	—	—
	present CCIRBF-Serial	2 × 2	1	1.9210E-07	73	—	9.6	3.8	380
105 × 105	present CCIRBF-Serial	4 × 4	1	1.9210E-07	29	—	11.2	3.2	320
	present CCIRBF-Parallel	4 × 4	4	1.9210E-07	101	0.6(17)	3.5	10.3	258
	present CCIRBF-Parallel	6 × 6	9	1.9210E-07	287	1.2(10)	12.1	3.0	33
	CCIRBF-Serial domain	—	1	1.1548E-7	—	—	59.4	—	—
105 × 105	present CCIRBF-Serial	2 × 2	1	1.1548E-7	96	—	16.3	3.6	360
	present CCIRBF-Serial	4 × 4	1	1.1548E-7	43	—	21.7	2.7	270
	present CCIRBF-Parallel	4 × 4	4	1.1548E-7	143	1.3(19)	6.7	8.9	223
	present CCIRBF-Parallel	6 × 6	9	1.1548E-7	201	1.0(9)	11.6	5.1	57
205 × 205	CCIRBF-Serial domain	—	1	7.8865E-08	—	—	2469.8	—	—
	present CCIRBF-Serial	2 × 2	1	7.8865E-08	67	—	177.4	13.9	139
	present CCIRBF-Serial	4 × 4	1	7.8865E-08	51	—	243.1	10.2	102
	present CCIRBF-Parallel	4 × 4	4	7.8865E-08	230	16.2(17)	93.8	26.3	658
305 × 305	present CCIRBF-Parallel	6 × 6	9	7.8865E-08	468	13.8(6)	230.4	10.7	119
	CCIRBF-Serial domain	—	—	—	—	—	—	—	—
	present CCIRBF-Serial	2 × 2	1	1.3785E-07	111	—	1102.5	—	—
	present CCIRBF-Serial	4 × 4	1	1.3785E-07	117	—	2386.2	—	—
305 × 305	present CCIRBF-Parallel	4 × 4	4	1.3785E-07	620	151.1(17)	876.5	—	—
	present CCIRBF-Parallel	6 × 6	9	1.3785E-07	786	101.8(7)	1468.3	—	—

8.3 Navier-Stokes equation

To construct a test problem of the stream function-vorticity formulation with known solution, we specify the stream function described in [Richards and Crane (1979)]

$$\psi = \frac{(x^2 + y^2)}{4} (\ln(x^2 + y^2) - 2), \tag{70}$$

on the unit square. The corresponding vorticity function, derived from (54), results in

$$\omega = \ln(x^2 + y^2), \tag{71}$$

The calculations are carried out on a uniform grid of 21×21 . A wide range of Reynold numbers, $Re = [10, 30, 50, 70, 90, 120, 150, 200]$, is employed. The value of $\beta = 50$ is chosen for calculations. Starting values of ω are analytic values of (71). To solve the steady vorticity equation, we utilise the vorticity equation (53), where $\frac{\partial u}{\partial t}$ is a pseudo time-derivative term. The vorticity equation (53) is subjected to Dirichlet boundary condition derived from the exact solution of (71). We deliberately employ a small time step, $\Delta t = 10^{-6}$, to minimise the effect of the approximate error in time. The criterion to be satisfied for termination of the iteration scheme is given

$$\left| \frac{\omega^n - \omega^{n-1}}{\omega^{n-1}} \right| < 10^{-6}. \tag{72}$$

Figure 15 shows numerical results produced by CCIRBF-Single domain, -Serial and -Parallel are much more accurate than those computed by the standard central FDM in Richards and Crane (1979). Table 10 shows the CCIRBF-Serial and -Parallel produce the same results with those of the CCIRBF-Single domain.

8.4 Lid driven cavity

The classical lid driven cavity has been considered as the test problem for the valuation of numerical methods and the validation of fluid flow solvers for the past decades. Figure 16 shows the problem definition and boundary conditions. The discretisation of the cavity domain is shown in Figure 17.

To derive the boundary conditions of the vorticity, the grid arrangement close to the bottom wall ($j = 1$) is illustrated in Figure 18. Apply Taylor series up to second order for $\psi_{i,j=2}$ [Biringen and Chow (2011)]

$$\psi_{i,j=2} = \psi_{i,j=1} + \frac{\partial \psi_{i,j=1}}{\partial y} h + \frac{\partial^2 \psi_{i,j=1}}{\partial y^2} \frac{h^2}{2}, \tag{73}$$

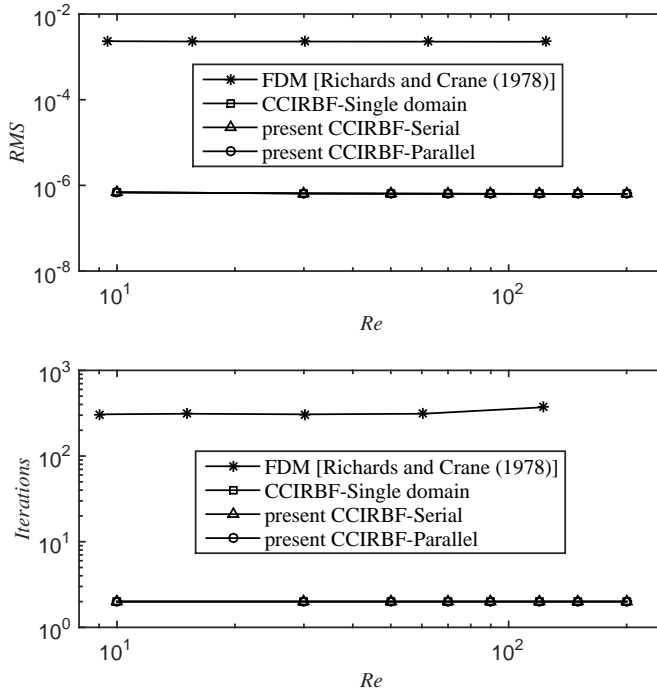


Figure 15: Navier Stokes problem with analytic solution, numerical solution using a grid of 21×21 , $\beta = 50$: The effect of Reynold number Re on the solution accuracy L_1 of vorticity (top) and on iteration number (bottom). It is noted that the results for the CCIRBF-Single domain, CCIRBF-Serial and CCIRBF-Parallel are indistinguishable.

Table 10: Navier Stokes analytic solution, $\beta = 50$: The effect of Reynold number Re on the solution accuracy RMS of vorticity.

Re	CCIRBF-Single domain	present CCIRBF-Serial	present CCIRBF-Parallel
	L_1	L_1	L_1
10	6.8802E-07	6.8802E-07	6.8802E-07
30	6.4819E-07	6.4819E-07	6.4819E-07
50	6.4047E-07	6.4047E-07	6.4047E-07
70	6.3717E-07	6.3717E-07	6.3717E-07
90	6.3534E-07	6.3534E-07	6.3534E-07
120	6.3374E-07	6.3374E-07	6.3374E-07
150	6.3278E-07	6.3278E-07	6.3278E-07
200	6.3182E-07	6.3182E-07	6.3182E-07

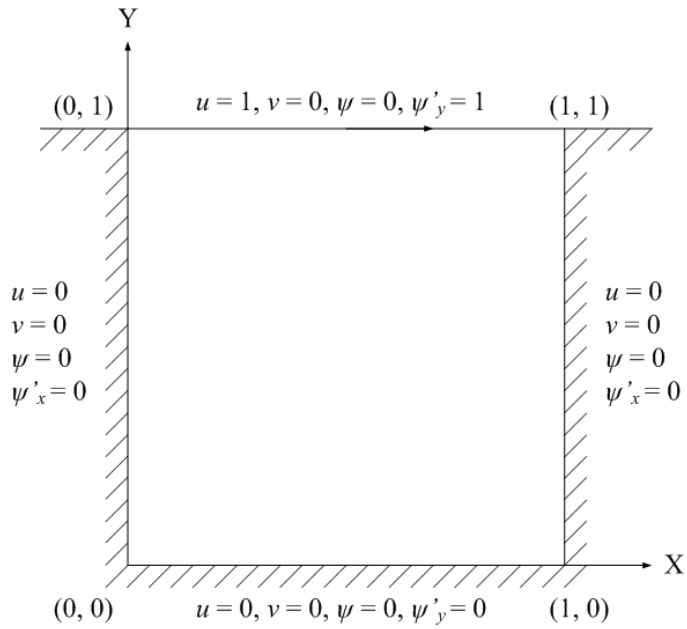


Figure 16: Lid driven cavity: problem configuration and boundary conditions in terms of the stream function.

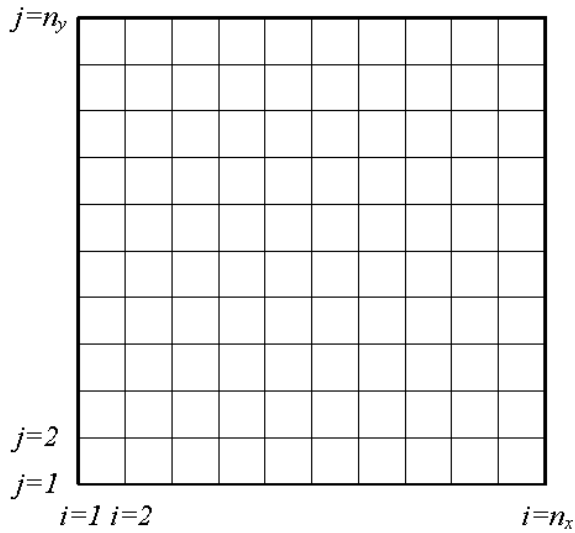


Figure 17: Lid driven cavity: domain discretisation.

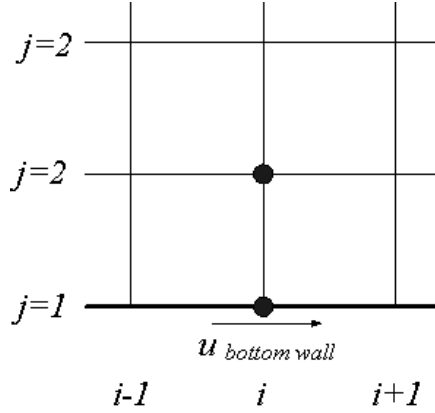


Figure 18: Lid driven cavity: Grid arrangement close to the bottom wall.

Using

$$-\frac{\partial^2 \psi_{i,j=1}}{\partial y^2} = \omega_{i,j=1} = \omega_{\text{bottom wall}}; \text{ and } \frac{\partial \psi_{i,j=1}}{\partial y} = u_{i,j=1} = u_{\text{bottom wall}}, \quad (74)$$

Equation (73) becomes

$$\psi_{i,j=2} = \psi_{i,j=1} + u_{\text{bottom wall}} h - \omega_{\text{bottom wall}} \frac{h^2}{2}, \quad (75)$$

or

$$\omega_{\text{bottom wall}} = (\psi_{i,j=1} - \psi_{i,j=2}) \frac{2}{h^2} + u_{\text{bottom wall}} \frac{2}{h}. \quad (76)$$

Similarly, at the top wall ($j = n_y$)

$$\omega_{\text{top wall}} = (\psi_{i,j=n_y} - \psi_{i,j=n_y-1}) \frac{2}{h^2} - u_{\text{top wall}} \frac{2}{h}. \quad (77)$$

At the left wall ($i = 1$)

$$\omega_{\text{left wall}} = (\psi_{i=1,j} - \psi_{i=2,j}) \frac{2}{h^2} - u_{\text{left wall}} \frac{2}{h}. \quad (78)$$

At the right wall ($i = n_x$)

$$\omega_{\text{right wall}} = (\psi_{i=n_x,j} - \psi_{i=n_x-1,j}) \frac{2}{h^2} + u_{\text{right wall}} \frac{2}{h}. \quad (79)$$

The numerical integration is done according to the following steps.

1. Set initial conditions at $t = 0$ (e.g., at all interior points set $\omega_{i,j}^{n-1} = 0$).

2. Obtain interior values of $\psi_{i,j}^n$ by solving

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \psi_{i,j}^n = -\omega_{i,j}^{n-1}. \tag{80}$$

3. Compute interior points of velocities by calculating

$$u_{i,j}^n = \frac{\partial \psi_{i,j}^n}{\partial y} \quad \text{and} \quad v_{i,j}^n = -\frac{\partial \psi_{i,j}^n}{\partial x}. \tag{81}$$

4. Calculate (76) to (79) for boundary values of $\omega_{i,j}^n$ using $\psi_{i,j}^n$.

5. Find right hand side (RHS) of vorticity equation (58)

$$RHS_{i,j}^n = \frac{1}{Re} \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \omega_{i,j}^{n-1} - \frac{\partial(u_{i,j}^n \omega_{i,j}^{n-1})}{\partial x} - \frac{\partial(v_{i,j}^n \omega_{i,j}^{n-1})}{\partial y}. \tag{82}$$

6. Compute interior values of $\omega_{i,j}^n$ using (58)

$$\omega_{i,j}^n = \omega_{i,j}^{n-1} + \Delta t RHS_{i,j}^n. \tag{83}$$

7. If a prescribed convergence criterion is reached, terminate the calculation; otherwise, go back to step 2.

Uniform grids of $\{11 \times 11, 31 \times 31, 41 \times 41, 51 \times 51\}$ and a range of $Re \in \{100, 400, 1000\}$ are employed in the simulation. A fixed time step is chosen to be $\Delta t = 0.0001$. Results of the present schemes are compared with some others [Ghia, Ghia, and Shin (1982); Gresho, Chan, Lee, and Upson (1984); Bruneau and Jouron (1990); Deng, Piquet, Queutey, and Visonneau (1994b); Botella and Peyret (1998); Sahin and Owens (2003); Thai-Quang, Le-Cao, Mai-Duy, and Tran-Cong (2012)]. From the literature, FDM results using very dense grids presented by Ghia, Ghia, and Shin (1982) and pseudo-spectral results presented by Botella and Peyret (1998) have been referred as "Benchmark" results for comparison purposes.

Tables 11, 12 and 13 show the present results for the extrema of the vertical and horizontal velocity profiles along the horizontal and vertical centrelines of the cavity for several Reynolds numbers. For $Re = 100$ (Table 11) and $Re = 1000$ (Table 13), the "Errors" are evaluated relative to "Benchmark" results of Botella and Peyret (1998). The results obtained by the present schemes are very comparable with others.

Table 11: Lid driven cavity, $\beta = 10$, $Re = 100$: Extrema of the vertical and horizontal velocity profiles along the horizontal and vertical centrelines of the cavity, respectively. "Errors" are relative to the "Benchmark" data.

Method	Grid	u_{min}	Error (%)	y_{min}	v_{max}	Error (%)	x_{max}	v_{min}	Error (%)	x_{min}
CCIRBF-Single domain	11×11	-0.1856399	13.27	0.4590	0.1486491	17.22	0.2574	-0.2393795	5.68	0.8379
CCIRBF-Single domain	31×31	-0.2115965	1.14	0.4596	0.1772869	1.27	0.2385	-0.2505319	1.29	0.8109
CCIRBF-Single domain	51×51	-0.2132844	0.35	0.4588	0.1789250	0.36	0.2372	-0.2526019	0.47	0.8105
Present CCIRBF-Serial	31×31	-0.2115965	1.14	0.4596	0.1772869	1.27	0.2385	-0.2505319	1.29	0.8109
Present CCIRBF-Serial	51×51	-0.2132844	0.35	0.4588	0.1789250	0.36	0.2372	-0.2526019	0.47	0.8105
Present CCIRBF-Parallel	31×31	-0.2115980	1.14	0.4596	0.1772881	1.27	0.2385	-0.2505346	1.29	0.8109
Present CCIRBF-Parallel	51×51	-0.2132822	0.36	0.4588	0.1789225	0.36	0.2372	-0.2525999	0.47	0.8105
CIRBF (u, v, p), [Thai-Quang (2012b)]	31×31	-0.2102259	1.78	0.4578	0.1768808	1.50	0.2370	-0.2501843	1.43	0.8107
CIRBF (u, v, p), [Thai-Quang (2012b)]	51×51	-0.2121503	0.88	0.4579	0.1781849	0.77	0.2372	-0.2520400	0.69	0.8107
FVM (u, v, p), [Deng (1994b)]	64×64	-0.21315	0.42	—	0.17896	0.34	—	-0.25339	0.16	—
FDM (ψ, ω), [Ghia (1982)]	129×129	-0.21090	1.47	0.4531	0.17527	2.40	0.2344	-0.24533	3.34	0.8047
FDM (u, v, p), [Bruneau (1990)]	129×129	-0.2106	1.61	0.4531	0.1786	0.54	0.2344	-0.2521	0.67	0.8125
FVM (u, v, p), [Sahin (2003)]	257×257	-0.213924	0.06	0.4598	0.180888	0.73	0.2354	-0.256603	1.10	0.8127
Benchmark, [Botella (1998)]		-0.2140424		0.4581	0.1795728		0.2370	-0.2538030		0.8104

Table 12: Lid driven cavity, $\beta = 2$, $Re = 400$: Extrema of the vertical and horizontal velocity profiles along the horizontal and vertical centrelines of the cavity, respectively.

Method	Grid	u_{min}	y_{min}	v_{max}	x_{max}	v_{min}	x_{min}
CCIRBF-Single domain	31×31	-0.3047737	0.2873	0.2771614	0.2255	-0.4220126	0.8568
CCIRBF-Single domain	41×41	-0.3201692	0.2814	0.2938922	0.2222	-0.4433914	0.8601
CCIRBF-Single domain	51×51	-0.3296499	0.2799	0.3030723	0.2265	-0.4537653	0.8621
Present CCIRBF-Serial	31×31	-0.3047737	0.2873	0.2771614	0.2255	-0.4220126	0.8568
Present CCIRBF-Serial	41×41	-0.3201424	0.2811	0.2939847	0.2219	-0.4434279	0.8601
Present CCIRBF-Serial	51×51	-0.3296499	0.2799	0.3030723	0.2265	-0.4537653	0.8621
Present CCIRBF-Parallel	31×31	-0.3047736	0.2873	0.2771612	0.2255	-0.4220121	0.8568
Present CCIRBF-Parallel	41×41	-0.3201424	0.2811	0.2939846	0.2219	-0.4434277	0.8601
Present CCIRBF-Parallel	51×51	-0.3296498	0.2799	0.3030722	0.2265	-0.4537652	0.8621
CIRBF (u, v, p), [Thai-Quang (2012b)]	31×31	-0.316205	0.2833	0.293696	0.2236	-0.435578	0.8583
CIRBF (u, v, p), [Thai-Quang (2012b)]	51×51	-0.323158	0.2814	0.297493	0.2248	-0.442770	0.8605
CIRBF (u, v, p), [Thai-Quang (2012b)]	71×71	-0.325168	0.2804	0.300818	0.2252	-0.449146	0.8620
FVM (u, v, p), [Deng (1994b)]	128×128	-0.32751	—	0.30271	—	-0.45274	—
FDM (ψ, ω), [Ghia (1982)]	129×129	-0.32726	0.2813	0.30203	0.2266	-0.44993	0.8594
FVM (u, v, p), [Sahin (2003)]	257×257	-0.328375	0.2815	0.304447	0.2253	-0.456316	0.8621

Table 13: Lid driven cavity, $\beta = 1$, $Re = 1000$: Extrema of the vertical and horizontal velocity profiles along the horizontal and vertical centrelines of the cavity, respectively. "Errors" are relative to the "Benchmark" data.

Method	Grid	u_{min}	Error (%)	y_{min}	v_{max}	Error (%)	x_{max}	v_{min}	Error (%)	x_{min}
CCIRBF-Single domain	31×31	-0.3602892	7.28	0.2135	0.3398357	9.84	0.1697	-0.4642615	11.92	0.8879
CCIRBF-Single domain	41×41	-0.3824678	1.57	0.2015	0.3654459	3.05	0.1646	-0.4973683	5.64	0.8913
CCIRBF-Single domain	51×51	-0.3915410	0.76	0.1925	0.3770969	0.04	0.1618	-0.5162784	2.05	0.8955
Present CCIRBF-Serial	41×41	-0.3823441	1.60	0.2015	0.3653190	3.08	0.1646	-0.4972672	5.66	0.8913
Present CCIRBF-Serial	51×51	-0.3915274	0.76	0.1925	0.3770822	0.04	0.1618	-0.5162669	2.05	0.8955
Present CCIRBF-Parallel	41×41	-0.3823440	1.60	0.2015	0.3653189	3.08	0.1646	-0.4972672	5.66	0.8913
Present CCIRBF-Parallel	51×51	-0.3915273	0.76	0.1925	0.3770822	0.04	0.1618	-0.5162668	2.05	0.8955
CIRBF (u, v, p), [Thai-Quang (2012b)]	71×71	-0.3755225	3.36	0.1753	0.3637009	3.51	0.1608	-0.5086961	3.49	0.9078
CIRBF (u, v, p), [Thai-Quang (2012b)]	91×91	-0.3815923	1.80	0.1735	0.3698053	1.89	0.1594	-0.5174658	1.82	0.9085
CIRBF (u, v, p), [Thai-Quang (2012b)]	111×111	-0.3840354	1.17	0.1728	0.3722634	1.24	0.1588	-0.5209683	1.16	0.9088
CIRBF (u, v, p), [Thai-Quang (2012b)]	129×129	-0.3848064	0.97	0.1724	0.3729119	1.07	0.1586	-0.5223350	0.90	0.9089
FVM (u, v, p), [Deng (1994b)]	128×128	-0.38511	0.89	—	0.37369	0.86	—	-0.5228	0.81	—
FDM (ψ, ω), [Ghia (1982)]	129×129	-0.38289	1.46	0.1719	0.37095	1.59	0.1563	-0.5155	2.20	0.9063
FEM (u, v, p), [Gresho (1984)]	129×129	-0.375	3.49	0.160	0.362	3.96	0.160	-0.516	2.10	0.906
FDM (u, v, p), [Bruneau (1990)]	256×256	-0.3764	3.13	0.1602	0.3665	2.77	0.1523	-0.5208	1.19	0.9102
FVM (u, v, p), [Sahin (2003)]	257×257	-0.388103	0.12	0.1727	0.376910	0.01	0.1573	-0.528447	0.26	0.9087
Benchmark, [Botella (1998)]		-0.3885698		0.1717	0.3769447		0.1578	-0.5270771		0.9092

Table 14: Lid driven cavity, $\beta = 10$, $Re = 100$: computational efficiency of the three algorithms, CCIRBF-Single domain, -Serial and -Parallel. It is noted that "Comm." is the abbreviation of "Communication".

Grid ($n_x \times n_y$)	Methods	Subdomains	CPUs	Iterations	Comm. time		Speed-up, S	Efficiency, E %
					Minute(%)	Minute		
51 × 51	CCIRBF-Single domain	—	1	169	—	19.7	—	—
	present CCIRBF-Serial	2 × 2	1	502	—	16.3	1.2	120
	present CCIRBF-Serial	4 × 4	1	820	—	30.4	0.6	—
101 × 101	present CCIRBF-Parallel	4 × 4	4	1972	2.7(13)	20.4	1.0	25
	present CCIRBF-Parallel	6 × 6	9	2384	4.3(11)	37.5	0.5	—
	CCIRBF-Single domain	—	1	1067	—	238.1	—	—
151 × 151	present CCIRBF-Serial	2 × 2	1	2153	—	92.2	2.6	260
	present CCIRBF-Serial	4 × 4	1	1475	—	103.0	2.3	230
	present CCIRBF-Parallel	4 × 4	4	4181	5.6(15)	36.5	6.5	163
201 × 201	present CCIRBF-Parallel	6 × 6	9	3936	4.9(9)	55.2	4.3	48
	CCIRBF-Single domain	—	1	969	—	285.9	—	—
	present CCIRBF-Serial	2 × 2	1	2049	—	105.3	2.7	270
51 × 51	present CCIRBF-Serial	4 × 4	1	1341	—	112.4	2.5	250
	present CCIRBF-Parallel	4 × 4	4	4562	6.1(16)	38.2	7.5	188
	present CCIRBF-Parallel	6 × 6	9	4256	5.2(9)	58.2	4.9	54
101 × 101	CCIRBF-Single domain	—	1	3509	—	2476.3	—	—
	present CCIRBF-Serial	2 × 2	1	1444	—	639.2	3.9	390
	present CCIRBF-Serial	4 × 4	1	1307	—	714.2	3.5	350
151 × 151	present CCIRBF-Parallel	4 × 4	4	2942	21.9(14)	151.7	16.3	408
	present CCIRBF-Parallel	6 × 6	9	2831	16.7(8)	215.2	11.5	128

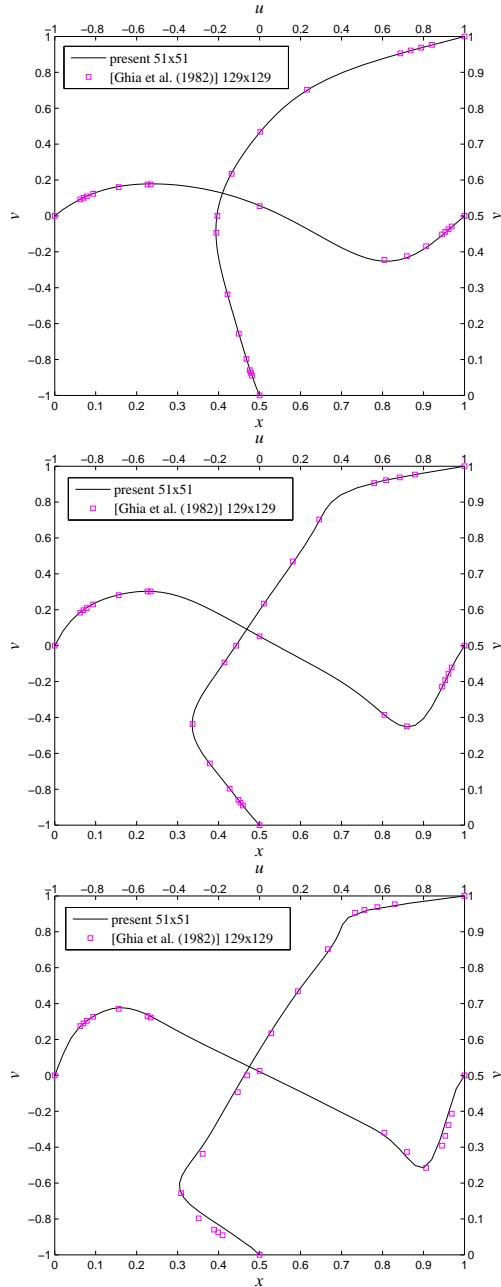


Figure 19: Lid driven cavity: Profiles of the u -velocity along the vertical centreline and the v -velocity along the horizontal centreline for $Re = 100$ (top), $Re = 400$ (middle) and $Re = 1000$ (bottom) with the grid of 51×51 .

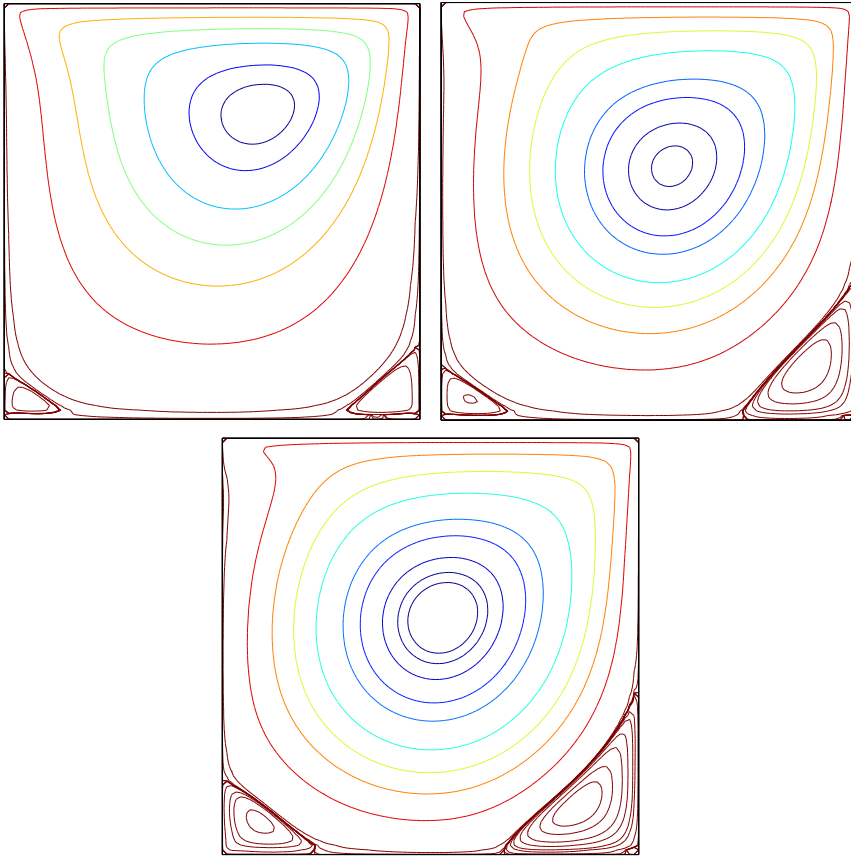


Figure 20: Lid driven cavity: Streamlines of the flow for $Re = 100$ (top), $Re = 400$ (middle) and $Re = 1000$ (bottom) with the grid of 91×91 . The contour values used here are taken to be the same at those in [Ghia, Ghia, and Shin (1982)].

From Tables 11, 12 and 13, we can observe the present scheme effectively achieves the benchmark results with fewer grids in comparison with the grids of some other methods used to obtain the benchmark results. In addition, those velocity profiles at $Re \in \{100, 400, 1000\}$ with the grid of 51×51 , are displayed in Figure 19, where the present solutions match the benchmark ones very well. The present scheme effectively achieves the benchmark results with fewer numbers of grids of 51×51 in comparison with the grid of 129×129 used to obtain the benchmark results in [Ghia, Ghia, and Shin (1982)].

To exhibit contour plots of the flow, a range of $Re \in \{100, 400, 1000\}$ and the grid of 91×91 are employed. Figures 20 and 21 show streamlines and iso-vorticity lines,

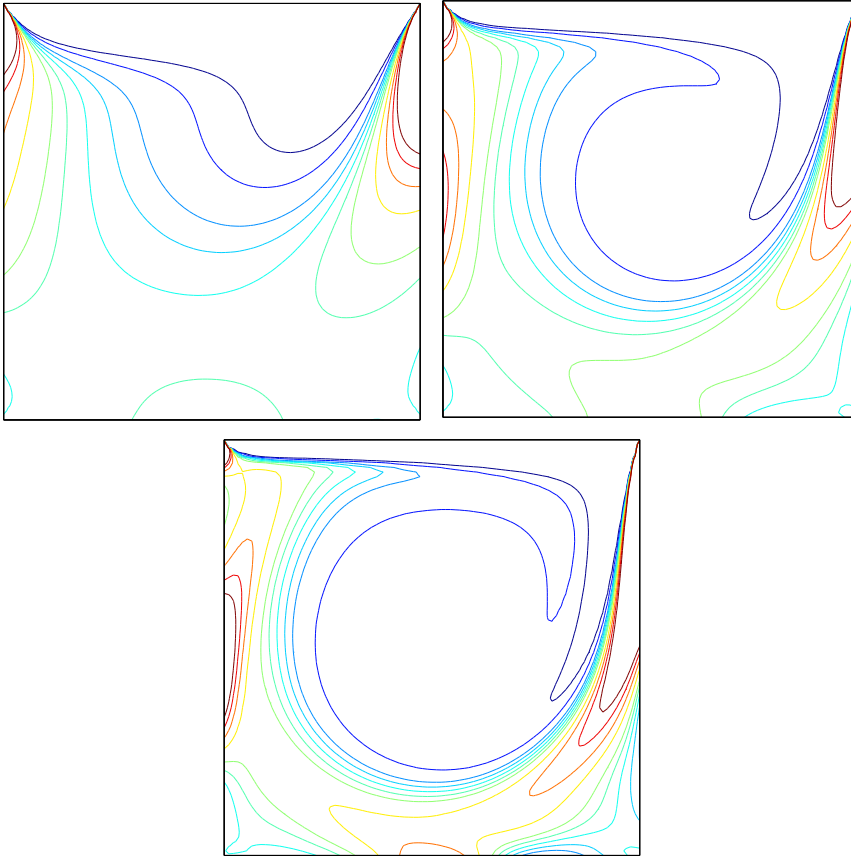


Figure 21: Lid driven cavity: Iso-vorticity lines of the flow for $Re = 100$ (top), $Re = 400$ (middle) and $Re = 1000$ (bottom) with the grid of 91×91 . The contour values used here are taken to be the same at those in [Ghia, Ghia, and Shin (1982)].

which are derived from the velocity field. These plots are also in good agreement with those reported in the literature.

For simplicity, the results of CCIRBF-Parallel are chosen to be plotted in Figures 19, 20 and 21. It is noted that the results of CCIRBF-Serial and -Parallel are indistinguishable.

Table 14 shows the indicative comparison of computational efficiency of the CCIRBF-Single domain, -Serial and -Parallel for the case of $Re = 100$ with various numbers of grids. The CCIRBF-Serial and -Parallel are more efficient to be compared with the CCIRBF-Single domain.

9 Concluding remarks

In this work, we carry out a convergence analysis for different types of domain decomposition (DD) preconditioners implemented with the coupled compact integrated RBF (CCIRBF). The performance of the present CCIRBF based algorithms are analysed in terms of the iteration count with different mesh sizes, number of subdomains and overlap sizes. The numerical results show that

- i. the present CCIRBF two-level multiplicative algorithm is the best one compared with the other present CCIRBF based algorithms.
- ii. the present CCIRBF two-level multiplicative algorithm is better than the HOC two-level multiplicative algorithm for the case of 1% and 25% overlaps. For other overlap cases, the present CCIRBF two-level multiplicative algorithm is comparable with the HOC two-level multiplicative algorithm.

In the implementation of the present CCIRBF in the DD preconditioners, we found that the incorporation of a coarse mesh problem into the multiplicative preconditioner is necessary to obtain a significant reduction in the computational iteration count. The present CCIRBF two-level multiplicative method yields small iteration counts over a wide range of numbers of subdomains, grid sizes and overlap sizes in our examples.

The present work introduces highly accurate serial and parallel algorithms using the CCIRBF for heat and fluid flow problems. The beauty of the proposed serial and parallel schemes is that they are able to produce almost the same level of accuracy as that of the single domain scheme. In computational examples, the results produced by serial and parallel algorithms are very compatible with other methods such as finite element method (FEM) and finite difference method (FDM). The capability of producing the stable and highly accurate results of the proposed algorithms is due to the utilisation of the coarse mesh of the two-level DD and the CCIRBF approximation. The serial and parallel algorithms offer a divide-and-conquer solution for large-scale partial differential equation (PDE) problems. Therefore, the proposed algorithms may be used as alternatives to the single domain scheme to solve large-scale problems which the single domain scheme is generally struggling to solve due to its ill-conditioned or fully populated companion matrix.

Acknowledgement: The first author would like to thank USQ for an International Postgraduate Research Scholarship.

References

- Atluri, S. N.; Zhu, T.** (1998): A new meshless local petrov-galerkin (mlpg) approach in computational mechanics. *Computational Mechanics*, vol. 22, no. 2, pp. 117–127.
- Babuska, I.; Melenk, J. M.** (1997): The partition of unity method. *International Journal for Numerical Methods in Engineering*, vol. 40, no. 4, pp. 727–758.
- Behr, M.** (2014): Gmres usage, chair for computational analysis of technical systems. <http://www.cats.rwth-aachen.de/library/research/technotes>.
- Belytschko, T.; Krongauz, Y.; Organ, D.; Fleming, M.; Krysl, P.** (1996): Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, vol. 139, no. 1-4, pp. 3–47.
- Belytschko, T.; Lu, Y. Y.; Gu, L.** (1994): Element-free galerkin methods. *International Journal for Numerical Methods in Engineering*, vol. 37, pp. 229–256.
- Biringen, S.; Chow, C.** (2011): *An introduction to computational fluid mechanics by example*. John Wiley and Sons, Hoboken, New Jersey, US.
- Botella, O.; Peyret, R.** (1998): Benchmark spectral results on the lid-driven cavity flow. *Comput. Fluids*, vol. 27, no. 4, pp. 421–433.
- Bruneau, C.; Jouron, C.** (1990): An efficient scheme for solving steady incompressible navier-stokes equations. *J. Comput. Phys.*, vol. 89, pp. 389–413.
- Cai, X.** (2003): *Advanced Topics in Computational Partial Differential Equations*, volume 33 of *Lecture Notes in Computational Science and Engineering*. Springer, Berlin, Germany.
- Chen, Y.; Lee, J.; Eskandarian, A.** (2006): *Meshless Methods in Solid Mechanics*. Springer, NY 10013, USA.
- Danaila, I.** (2007): *An introduction to scientific computing: twelve computational projects solved with MATLAB*. Springer, NY 10013, USA.
- Deng, G.; Piquet, J.; Queutey, P.; Visonneau, M.** (1994b): Incompressible-flow calculations with a consistent physical interpolation finite-volume approach. *Comput. Fluids*, vol. 23, no. 8, pp. 1029–1047.
- Ghia, U.; Ghia, K.; Shin, C.** (1982): High-resolutions for incompressible flows using navier-stokes equations and a multigrid method. *J. Comput. Phys.*, vol. 48, pp. 387–411.
- Gresho, P.; Chan, S.; Lee, R.; Upson, C.** (1984): A modified finite element method for solving the time-dependent, incompressible navier-stokes equations. part 2: Applications. *Int. J. Numer. Meth. Fluids*, vol. 4, pp. 619–640.

Jenkins, E. W.; Kees, C. E.; Kelley, C. T.; Miller, C. T. (2001): An aggregation-based domain decomposition preconditioner for groundwater flow. *SIAM Journal on Scientific Computing*, vol. 23, no. 2, pp. 430–441.

Kansa, E. (1990a): Multiquadrics - a scattered data approximation scheme with applications to computational fluid-dynamics - i. surface approximations and partial derivative estimates. *Computers and Mathematics with Applications*, vol. 19, no. 8/9, pp. 127–145.

Kansa, E. (1990b): Multiquadrics - a scattered data approximation scheme with applications to computational fluid-dynamics - ii. solutions to parabolic, hyperbolic and elliptic partial differential equations. *Computers and Mathematics with Applications*, vol. 19, no. 8/9, pp. 147–161.

Lancaster, P.; Salkauskas, K. (1981): Surfaces generated by moving least squares methods. *Mathematics of Computation*, vol. 37, no. 155, pp. 141–158.

Li, J.; Cheng, A. H. D.; Chen, C. S. (2003): A comparison of efficiency and error convergence of multiquadric collocation method and finite element method. *Engineering Analysis with Boundary Elements*, vol. 27, pp. 251–257.

Liu, W. K.; Chen, Y.; Chang, C. T.; Belytschko, T. (1996): Advances in multiple scale kernel particle methods. *Computational Mechanics*, vol. 18, no. 2, pp. 73–111.

Lucy, L. B. (1977): A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, vol. 82, no. 12, pp. 1013–1024.

Mai-Duy, N.; Tran-Cong, T. (2001a): 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. (2001b): Numerical solution of navier-stokes equations using multiquadric radial basis function networks. *International Journal for Numerical Methods in Fluids*, vol. 37, no. 1, pp. 65–86.

Mai-Duy, N.; Tran-Cong, T. (2003): Approximation of function and its derivatives using radial basis function networks. *Applied Mathematical Modelling*, vol. 27, no. 3, pp. 197–220.

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

Nayroles, B.; Touzot, G.; Villon, P. (1992): Generalizing the finite element method: Diffuse approximation and diffuse elements. *Computational Mechanics*, vol. 10, no. 5, pp. 307–318.

- Niyogi, P.; Chakrabartty, S.; Laha, M.** (2009): *Introduction to computational fluid dynamics*. Dorling Kindersley, 482, F.I.E., Patparganj, Delhi 110 092, India.
- Quarteroni, A.; Valli, A.** (1999): *Domain decomposition methods for partial differential equations*. Clarendon Press, Oxford.
- Richards, C.; Crane, C.** (1979): The accuracy of finite difference schemes for the numerical solution of the navier-stokes equations. *Applied Mathematical Modelling*, vol. 3.
- Saad, Y.; Schultz, M.** (1986): Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, vol. 7, pp. 856–869.
- Sahin, M.; Owens, R.** (2003): A novel fully implicit finite volume method applied to the lid-driven cavity problem - part i: High reynolds number flow calculations. *Int. J. Numer. Meth. Fluids*, vol. 42, pp. 57–77.
- Smith, B. F.; Bjorstad, P. E.; Gropp, W. D.** (1996): *Domain Decomposition Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, NY 10011-4211, USA.
- Strang, G.** (2007): *Computational science and engineering*. Wellesley-Cambridge Press, MA 02482, USA.
- Thai-Quang, N.; Le-Cao, K.; Mai-Duy, N.; Tran-Cong, T.** (2012): A high-order compact local integrated-rbf scheme for steady-state incompressible viscous flows in the primitive variables. *CMES: Computer Modeling in Engineering and Sciences*, vol. 84, no. 6, pp. 528–557.
- Thai-Quang, N.; Mai-Duy, N.; Tran, C.-D.; Tran-Cong, T.** (2012): High-order alternating direction implicit method based on compact integrated-rbf approximations for unsteady/steady convection-diffusion equations. *CMES: Computer Modeling in Engineering and Sciences*, vol. 89, no. 3, pp. 189–220.
- Tian, Z.; Liang, X.; Yu, P.** (2011): A higher order compact finite difference algorithm for solving the incompressible navier stockes equations. *International Journal for Numerical Methods in Engineering*, vol. 88, pp. 511–532.
- Tien, C. M. T.; Thai-Quang, N.; Mai-Duy, N.; Tran, C.-D.; Tran-Cong, T.** (2015): A three-point coupled compact integrated rbf scheme for second-order differential problems. *CMES: Computer Modeling in Engineering and Sciences (Accepted)*.
- Toselli, A.; Widlund, O.** (2005): *Domain Decomposition Methods - Algorithms and Theory*. Springer, Berlin, Germany.

Zerroukat, M.; Power, H.; Chen, C. S. (1998): A numerical method for heat transfer problems using collocation and radial basis functions. *International Journal for Numerical Methods in Engineering*, vol. 42, pp. 1263–1278.