New approximations for one-dimensional 3-point and two-dimensional 5-point compact integrated RBF stencils

N. Mai-Duy[1,*] and D. Strunin[2]
[1] School of Mechanical and Electrical Engineering,
[2] School of Sciences,
University of Southern Queensland, Toowoomba, QLD 4350, Australia

Submitted to *Engineering Analysis with Boundary Elements*, June 2020;
Revised, December 2020

Abstract This paper presents some new compact approximation stencils based on integrated radial basis functions (IRBFs) for numerically solving second-order elliptic differential problems on Cartesian grids. Higher-order IRBF schemes are employed to approximate the field/dependent variable. The IRBF approximations in each direction are based on 3 points and constructed independently, where derivatives of the second, third, fourth, fifth and sixth orders along the grid line are enforced at the two end-points. The imposed nodal derivative values are simply acquired through a Picard-type iteration scheme. The stencil is made up of 3 points and 5 points for 1D and 2D discretisations, respectively. Numerical results show that the proposed stencils yield a high rate of convergence with respect to grid refinement, e.g. up to the 13th order for 1D problems and to the 9th order for 2D problems.

Keywords: compact approximation, local approximation, integrated radial basis function, 3-point stencil, 5-point stencil

*Corresponding author E-mail: nam.mai-duy@usq.edu.au, Telephone +61-7-46312748, Fax +61-7-46312529

1 Introduction

Ordinary/partial differential equations (ODEs/PDEs) arise in the mathematical modelling of physical phenomena. For most problems, their governing equations can only be solved by means of numerical methods. Conventional techniques such as the finite-element and finite-volume methods require a mesh to support the interpolation of the field variable. The generation of meshes for problems defined on complex geometries can be very time-consuming. As a result, much effort has been devoted to develop meshless methods and Cartesian-grid methods. Meshless methods have attracted a great deal of attention in recent decades [1]. The domain of interest is represented by a set of unstructured points, and the interpolation of the field variable is simply based on overlapping cells that do not require any fixed topology. Cartesian-grid methods have a long history with the early works ([2]) dated back to the 1930's. The domain is discretised by using Cartesian grids that can be generated in a straightforward manner. The methods have been increasingly used for solving problems with complexly-shaped domains due to their economical preprocessing. For most of the reported Cartesian-grid methods, their discretisations are based on finite-difference/finite-volume formulations, which typically lead to schemes that are second-order accurate [3,4].

Radial basis functions (RBFs) have been widely used in solving ODEs and PDEs (see, for example, [5,6,7,8,9,10, 11,12,13]). In the RBF collocation methods [14], the field variable, which is typically a nonlinear function, is expressed as a linear combination of RBFs, of which the accuracy can be enhanced quickly with an increase in the number of RBFs (i.e. the number of nodes). For several types of RBFs such as the Gaussian and multiquadric functions, they have the ability to offer an exponential rate of convergence [15,16,17]. With all nodes taken into account, the RBF approximations lead to system matrices that are fully populated. In addition, their condition numbers grow rapidly with the number of nodes which can depend exponentially on the number of dimensions. Utilisation of extended-precision floating point arithmetic to bypass the solution of the ill-conditioned linear system was shown to be effective [18,19]. The computational costs for solving global RBF systems become prohibitive when a large number of nodes are employed. For local RBF approximations [20], only a few

nodes are activated for the approximation at a point. It leads to sparse algebraic systems, which can be solved in an efficient way. A drawback here is that their accuracies are significantly reduced (i.e. algebraic convergence rates only). In [21], for equispaced nodes, the error was analytically estimated as $\epsilon \approx O((h/a)^p)$, where h is the nodal distance and a the RBF width/shape-parameter. The exponent p takes value of 2 when the number of supporting nodes is in range of 5 to 12. One effective way to overcome accuracy reductions is to develop compact local RBF stencils, where the RBF approximations are expressed in terms of nodal values of not only the field variable but also its derivatives [22,23,24,25]. They can be implemented with differentiated RBFs or integrated RBFs on Cartesian grids [24,25,26] or unstructured nodes [23,27]. Nodal derivative values can be included by using the Hermite interpolation approach [23,27] or by means of integration constants [26]. An advantage of the former over the latter is that its interpolation matrix is symmetric and guaranteed to be invertible. However, the former involves the computation of fourth-order derivatives including the cross/mixed ones. For the latter, one needs to compute pure second-order derivatives. Improved rates of convergence are acquired with compact approximations. In the case of using Cartesian grids, 1D-approximation-based stencils are typically employed with 3 nodes in each direction. If more than 3 nodes are employed, special treatments will be required for the stencils associated with interior nodes near the boundaries. In addition, the level of sparseness of the system matrix is also reduced (i.e. larger matrix bandwidths). For one-dimensional 3-point stencils and two-dimensional 5-point stencils, fourth-order accurate solutions were achieved [23,25]. In this work, we propose new IRBF approximations on the 3- and 5-point stencils to further improve their rates of convergence. The error can decrease at the rate of much higher than 4 on regular and irregular 3-point/5-point stencils. We note some recent efforts in the development of robust reduced-order models (ROMs) (e.g. [28]) to solve large-scale applications.

The structure of the paper is as follows. In Section 2, compact IRBF stencils including the proposed ones are presented. In Section 3, the implementation of the proposed stencils is discussed. In Section 4, verification is conducted in linear and nonlinear problems with rectangular and non-rectangular domains, and uniform and nonuniform grids. Section 5

gives some concluding remarks.

2 Compact IRBF stencils

2.1 One dimension

Approximation stencils based on IRBFs including the proposed ones are presented in the solution of

$$-\frac{d^2u}{dx^2}+\lambda\frac{du}{dx}=f(x),\quad x_A\le x\le x_B, \tag{1}$$

with Dirichlet boundary conditions only, and with Dirichlet and Neumann boundary conditions. In (1), $f(x)$ is a smooth function and λ is a constant. The domain $[x_A, x_B]$ is disrectised by a set of N points: $(x_1 = x_A, x_2, \cdots, x_N = x_B)$.

Consider a 3-point stencil $[x_{i-1}, x_i, x_{i+1}]$, over which the field variable can be represented by an IRBF approximation scheme

$$\frac{d^qu(x)}{dx^q}=\sum_{i=1}^{3}w_iG_i(x)=\sum_{i=1}^{3}w_iI_i^{(q)}(x), \tag{2}$$

$$\frac{d^{q-1}u(x)}{dx^{q-1}}=\sum_{i=1}^{3}w_iI_i^{(q-1)}(x)+c_1, \tag{3}$$

$$\frac{d^{q-2}u(x)}{dx^{q-2}}=\sum_{i=1}^{3}w_iI_i^{(q-2)}(x)+c_1x+c_2, \tag{4}$$

$$\cdots\quad\cdots\quad\cdots$$

$$\frac{du(x)}{dx}=\sum_{i=1}^{3}w_iI_i^{(1)}(x)+c_1\frac{x^{q-2}}{(q-2)!}+c_2\frac{x^{q-3}}{(q-3)!}+\cdots+c_{q-1}, \tag{5}$$

$$u(x)=\sum_{i=1}^{3}w_iI_i^{(0)}(x)+c_1\frac{x^{q-1}}{(q-1)!}+c_2\frac{x^{q-2}}{(q-2)!}+\cdots+c_{q-1}x+c_q, \tag{6}$$

where q is the order of an IRBF scheme, $G_i(x)$ the RBF, $I_i^{(q-1)}(x)=\int I_i^{(q)}(x)dx$, $I_i^{(q-2)}(x)=\int I_i^{(q-1)}(x)dx$, $\cdots$, $I_i^{(0)}(x)=\int I_i^{(1)}(x)dx$, (w_1, w_2, w_3) the coefficients, and $(c_1, c_2, \cdots, c_q)$ the integration constants. For the multiquadric function, $G_i(x)=\sqrt{(x-x_i)^2+a_i^2}$, where a is

the width/shape-parameter.

Classical stencils

For second-order differential problems, $q = 2$ is typically employed. A mapping from the RBF space to the physical space is then formed by collocating expression (6) (function u) at the three nodes of the stencil:

$$\widehat{u} = \mathcal{C}\widehat{w}, \tag{7}$$

where $\mathcal{C}$ is called the conversion matrix, which are of dimensions 3×5 here,

$$\mathcal{C} = \begin{bmatrix} I_1^{(0)}(x_{i-1}), & I_2^{(0)}(x_{i-1}), & I_3^{(0)}(x_{i-1}), & x_{i-1}, & 1 \\ I_1^{(0)}(x_i), & I_2^{(0)}(x_i), & I_3^{(0)}(x_i), & x_i, & 1 \\ I_1^{(0)}(x_{i+1}), & I_2^{(0)}(x_{i+1}), & I_3^{(0)}(x_{i+1}), & x_{i+1}, & 1 \end{bmatrix},$$

$$\widehat{u} = (u_{i-1}, u_i, u_{i+1})',$$

$$\widehat{w} = (w_1, w_2, w_3, c_1, c_2)'.$$

Solving (7) yields

$$\widehat{w} = \mathcal{C}^{-1}\widehat{u}, \tag{8}$$

where $\mathcal{C}^{-1}$ is the Moore-Penrose pseudoinverse of matrix $\mathcal{C}$.

Making use of (8), the second derivative of u at $x = x_i$ is computed as

$$\frac{d^2u_i}{dx^2} = \left[I_1^{(2)}(x_i), I_2^{(2)}(x_i), I_3^{(2)}(x_i), 0, 0\right] \mathcal{C}^{-1} (u_{i-1}, u_i, u_{i+1})', \tag{9}$$

or

$$\frac{d^2u_i}{dx^2} = \nu_1 u_{i-1} + \nu_2 u_i + \nu_3 u_{i+1}, \tag{10}$$

where (ν_1, ν_2, ν_3) are known coefficients.

Other derivatives of u at $x = x_i$ (e.g. du_i/dx) can be obtained by replacing derivative expression in the RBF space on RHS of (9) with an appropriate one (e.g. $\left[I_1^{(1)}(x_i), I_2^{(1)}(x_i), I_3^{(1)}(x_i), 1, 0\right]$).

Compact stencils (CIRBF)

For compact approximations, nodal derivative values are also included on the right site of (10). In [25], the conversion matrix and its associated vectors are constructed as

$$\mathcal{C} = \begin{bmatrix} I_1^{(0)}(x_{i-1}), & I_2^{(0)}(x_{i-1}), & I_3^{(0)}(x_{i-1}), & x_{i-1}, & 1 \\ I_1^{(0)}(x_i), & I_2^{(0)}(x_i), & I_3^{(0)}(x_i), & x_i, & 1 \\ I_1^{(0)}(x_{i+1}), & I_2^{(0)}(x_{i+1}), & I_3^{(0)}(x_{i+1}), & x_{i+1}, & 1 \\ -I_1^{(2)}(x_{i-1}) + \lambda I_1^{(1)}(x_{i-1}), & -I_2^{(2)}(x_{i-1}) + \lambda I_2^{(1)}(x_{i-1}), & -I_3^{(2)}(x_{i-1}) + \lambda I_3^{(1)}(x_{i-1}), & \lambda, & 0 \\ -I_1^{(2)}(x_{i+1}) + \lambda I_1^{(1)}(x_{i+1}), & -I_2^{(2)}(x_{i+1}) + \lambda I_2^{(1)}(x_{i+1}), & -I_3^{(2)}(x_{i-1}) + \lambda I_3^{(1)}(x_{i+1}), & \lambda, & 0 \end{bmatrix},$$

$$\widehat{u} = (u_{i-1}, u_i, u_{i+1}, f_{i-1}, f_{i+1})',$$

$$\widehat{w} = (w_1, w_2, w_3, c_1, c_2)'.$$

The conversion matrix $\mathcal{C}$ here is of dimensions 5×5.

Expression for computing the second derivative of u at $x = x_i$ becomes

$$\frac{d^2 u_i}{dx^2} = \left(I_1^{(2)}(x_i), I_2^{(2)}(x_i), I_3^{(2)}(x_i), 0, 0 \right) \mathcal{C}^{-1} \left(u_{i-1}, u_i, u_{i+1}, f_{i-1}, f_{i+1} \right)', \tag{11}$$

or

$$\frac{d^2 u_i}{dx^2} = \mu_1 u_{i-1} + \mu_2 u_i + \mu_3 u_{i+1} + \overline{\mu}_1 f_{i-1} + \overline{\mu}_2 f_{i+1}, \tag{12}$$

where $\mathcal{C}^{-1}$ is the inverse of matrix $\mathcal{C}$, and $(\mu_1, \mu_2, \mu_3, \overline{\mu}_1, \overline{\mu}_3)$ are known coefficients. An alternative way to obtain these coefficients is to solve the following algebraic equation set:

$$\mathcal{C}'(\mu_1, \mu_2, \mu_3, \overline{\mu}_1, \overline{\mu}_3)' = \left(I_1^{(2)}(x_i), I_2^{(2)}(x_i), I_3^{(2)}(x_i), 0, 0 \right)', \tag{13}$$

where Gaussian elimination can be used without explicitly computing the inverse.

Proposed compact stencils

In this study, we have employed $q = 10$ and imposed the following values of function $f(x)$

and its derivatives:

$$\begin{aligned} & f_{i-1}, && f_{i+1}, \\ & df_{i-1}/dx, && df_{i+1}/dx, \\ & d^2f_{i-1}/dx^2, && d^2f_{i+1}/dx^2, \\ & d^3f_{i-1}/dx^3, && d^3f_{i+1}/dx^3, \\ & d^4f_{i-1}/dx^4, && d^4f_{i+1}/dx^4. \end{aligned} \tag{14}$$

In the present conversion system $\widehat{u} = \mathcal{C}\widehat{w}$, there are 13 equations (3 for function u and 10 for function $f(x)$ and its derivatives) and 13 unknowns

$$\widehat{u} = \left(u_{i-1}, u_i, u_{i+1}, f_{i-1}, f_{i+1}, \cdots, \frac{d^4 f_{i-1}}{dx^4}, \frac{d^6 f_{i+1}}{dx^4}\right)',$$

$$\widehat{w} = (w_1, w_2, w_3, c_1, c_2, \cdots, c_{10})',$$

$$\mathcal{C} = \begin{bmatrix} I_1^{(0)}(x_{i-1}), & \cdots & I_3^{(0)}(x_{i-1}), & \frac{x_{i-1}^9}{9!}, & \cdots & 1 \\ I_1^{(0)}(x_i), & \cdots & I_3^{(0)}(x_i), & \frac{x_i^9}{9!}, & \cdots & 1 \\ I_1^{(0)}(x_{i+1}), & \cdots & I_3^{(0)}(x_{i+1}), & \frac{x_{i+1}^9}{9!}, & \cdots & 1 \\ -I_1^{(2)}(x_{i-1}) + \lambda I_1^{(1)}(x_{i-1}), & \cdots & -I_3^{(2)}(x_{i-1}) + \lambda I_3^{(1)}(x_{i-1}), & -\frac{x_{i-1}^7}{7!} + \lambda \frac{x_{i-1}^8}{8!}, & \cdots, & 0 \\ -I_1^{(2)}(x_{i+1}) + \lambda I_1^{(1)}(x_{i+1}), & \cdots & -I_3^{(2)}(x_{i+1}) + \lambda I_3^{(1)}(x_{i+1}), & -\frac{x_{i+1}^7}{7!} + \lambda \frac{x_{i+1}^8}{8!}, & \cdots, & 0 \\ & \cdots & & & \cdots & \\ -I_1^{(6)}(x_{i-1}) + \lambda I_1^{(5)}(x_{i-1}), & \cdots & -I_3^{(6)}(x_{i-1}) + \lambda I_3^{(5)}(x_{i-1}), & -\frac{x_{i-1}^3}{3!} + \lambda \frac{x_{i-1}^4}{4!}, & \cdots, & 0 \\ -I_1^{(6)}(x_{i+1}) + \lambda I_1^{(5)}(x_{i+1}), & \cdots & -I_3^{(6)}(x_{i+1}) + \lambda I_3^{(5)}(x_{i+1}), & -\frac{x_{i+1}^3}{3!} + \lambda \frac{x_{i+1}^4}{4!}, & \cdots, & 0 \end{bmatrix}.$$

When compared to classical and compact IRBF stencils in [25], it can be seen that an IRBF scheme of higher order is employed here, and not only function $f(x)$ but also its derivatives are imposed at the two end-nodes. It leads to

$$\begin{aligned} \frac{d^2 u_i}{dx^2} = & \left(I_1^{(2)}(x_i), I_2^{(2)}(x_i), I_3^{(2)}(x_i), \frac{x_i^7}{7!}, \frac{x_i^6}{6!}, \cdots, 1, 0, 0\right) \mathcal{C}^{-1} \\ & \left(u_{i-1}, u_i, u_{i+1}, f_{i-1}, f_{i+1}, \cdots, \frac{d^4 f_{i-1}}{dx^4}, \frac{d^4 f_{i+1}}{dx^4}\right)', \end{aligned} \tag{15}$$

or

$$\frac{d^2 u_i}{dx^2} = \eta_1 u_{i-1} + \eta_2 u_i + \eta_3 u_{i+1} + \overline{\eta}_1 f_{i-1} + \overline{\eta}_2 f_{i+1} + \cdots + \overline{\eta}_9 \frac{d^4 f_{i-1}}{dx^4} + \overline{\eta}_{10} \frac{d^4 f_{i+1}}{dx^4}. \tag{16}$$

As an alternative to (15)/(16), the coefficients can be acquired by solving the following algebraic equation set:

$$\mathcal{C}'(\eta_1,\eta_2,\eta_3,\overline{\eta}_1,\cdots,\overline{\eta}_{10})' = \left(I_1^{(2)}(x_i), I_2^{(2)}(x_i), I_3^{(2)}(x_i), \frac{x_i^7}{7!},\cdots,0\right)'. \tag{17}$$

Coefficients (ν_1,ν_2,ν_3) in (10), $(\mu_1,\mu_2,\mu_3,\overline{\mu}_1,\overline{\mu}_2)$ in (12), and $(\eta_1,\eta_2,\eta_3,\overline{\eta}_1,\cdots,\overline{\eta}_{10})$ in (16)/(17) are all obtained numerically. There is no difference in handling between uniform and non-uniform stencils. For problems in higher dimensions, the 1D formulas (10), (12) and (16) can be applied separately in each direction.

Discretisations based on (10), (12) or (16) lead to systems of algebraic equations. In the case of Dirichlet boundary conditions only, the obtained algebraic system can be written as

$$\mathcal{A}\widehat{u} = \widehat{d}, \tag{18}$$

where

$$\mathcal{A} = \begin{bmatrix} 1 & 0 & . & . & . & . & . & . & . \\ c_2 & a_2 & b_2 & 0 & . & . & . & . & . \\ . & c_3 & a_3 & b_3 & 0 & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & 0 & c_{N-1} & a_{N-1} & a_{N-1} \\ . & . & . & . & . & . & . & 0 & 1 \end{bmatrix},$$

$$\widehat{u} = (u_1, u_2, \cdots, u_N)',$$

$$\widehat{d} = (d_1, d_2, \cdots, d_N)',$$

in which d_1 and d_N are given boundary values, (c_i, a_i, b_i) is value of (ν_1,ν_2,ν_3), (μ_1,μ_2,μ_3) and (η_1,η_2,η_3) which are acquired from the ith stencil. When i runs from 2 to $N-1$,

$$d_i = f_i$$

for classical stencils,

$$d_i = f_i - \overline{\eta}_1 f_{i-1} - \overline{\eta}_2 f_{i+1}$$

for compact IRBF stencils and

$$d_i = f_i - \overline{\eta}_1 f_{i-1} - \overline{\eta}_2 f_{i+1} + \cdots - \overline{\eta}_9 \frac{d^4 f_{i-1}}{dx^4} - \overline{\eta}_{10} \frac{d^4 f_{i+1}}{dx^4}$$

for the proposed CIRBF stencils.

Equations in system (18) can be rewritten in general form

$$c_i u_{i-1} + a_i u_i + b_i u_{i+1} = d_i, \quad i = 1, 2, \cdots, N. \tag{19}$$

For the first ($i = 1$) and last ($i = N$) equations, one has $a_1 = 1$, $b_1 = 0$ and $c_1 = 0$, and $a_N = 1$, $b_N = 0$ and $c_N = 0$, respectively. The system (19) can be solved efficiently by the Thomas algorithm or the TDMA (TriDiagonal-Matrix Algorithm), which is based on two processes: forward substitution and backward substitution.

In the case of Dirichlet and Neumann boundary conditions, the last equation in (18) is replaced with expression for computing the first derivative of u at $x = x_N$.

2.2 Two dimensions

For simplicity, the proposed CIRBF stencils are presented in the solution of

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y), \quad x_A \leq x \leq x_B, \quad y_C \leq y \leq y_D, \tag{20}$$

subjected to Dirichlet boundry conditions and defined on a rectangular domain that is disrectised by a set of $N_x \times N_y$ points.

The stencil is made up of 5 nodes. Consider an interior node (i, j) (intersection between the ith horizontal and jth vertical grid lines). Its four neighbours are $(i-1, j)$, $(i+1, j)$,

$(i, j-1)$ and $(i, j+1)$. The IRBF approximation in the x and y directions are constructed on (x_{i-1}, x_i, x_{i+1}) and (y_{j-1}, y_j, y_{j+1}), respectively by means of (16). This discretisation process leads to a set of algebraic equations:

$$\frac{\partial^2 u_{i,j}}{\partial x^2} + \frac{\partial^2 u_{i,j}}{\partial y^2} = f_{i,j}, \tag{21}$$

where $2 \le i \le N_x - 1$ and $2 \le j \le N_y - 1$. In (21), $\partial^2 u_{i,j}/\partial x^2$ is a linear function of $u_{i,j}$, $u_{i-1,j}$, $u_{i+1,j}$, $\partial^2 u_{i-1,j}/\partial x^2$, $\partial^2 u_{i+1,j}/\partial x^2$, ..., $\partial^6 u_{i-1,j}/\partial x^6$ and $\partial^6 u_{i+1,j}/\partial x^6$, and $\partial^2 u_{i,j}/\partial y^2$ a linear function of $u_{i,j}$, $u_{i,j-1}$, $u_{i,j+1}$, $\partial^2 u_{i,j-1}/\partial y^2$, $\partial^2 u_{i,j+1}/\partial y^2$, ..., $\partial^6 u_{i,j-1}/\partial y^6$ and $\partial^6 u_{i,j+1}/\partial y^6$. The nodal derivative values here are unknown, and we employ a Picard-type iteration scheme to find their values. The solution procedure is described below:

1. Set nodal derivatives of the 2nd, 3rd, 4th, 5th and 6th orders of u along the grid lines to zero.

2. Solve the algebraic equation set derived from (21).

3. Relax the solution

$$\mathrm{sol}^k = \zeta \mathrm{sol}^k + (1 - \zeta)\mathrm{sol}^{k-1}, \tag{22}$$

   where subscript k denotes a current iteration, "sol" a vector containing interior nodal values of u, and ζ a relax factor $(0 < \zeta \le 1)$.

4. Compute the derivatives.

5. Compute Convergence Measure defined as the 2-norm ratio of the two vectors $(\mathrm{sol}^k - \mathrm{sol}^{k-1})$ and sol^k.

6. Check CM. If $CM < 10^{-12}$, stop the iteration and output the result. Otherwise, repeat from Step 2.

To compute the derivatives of 2nd, 3rd, 4th, 5th and 6th orders in Step 4, we employ a global one-dimensional IRBF approximation scheme. Along a horizontal grid line, the scheme is

based on

$$\frac{\partial^q u(x)}{\partial x^q} = \sum_{i=1}^{N_x} w_i G_i(x) = \sum_{i=1}^{N_x} w_i I_i^{(q)}(x), \tag{23}$$

$$\frac{\partial^{q-1} u(x)}{\partial x^{q-1}} = \sum_{i=1}^{N_x} w_i I_i^{(q-1)}(x) + c_1, \tag{24}$$

$$\cdots \quad \cdots \quad \cdots$$

$$\frac{\partial u(x)}{\partial x} = \sum_{i=1}^{N_x} w_i I_i^{(1)}(x) + c_1 \frac{x^{q-2}}{(q-2)!} + c_2 \frac{x^{q-3}}{(q-3)!} + \cdots + c_{q-1}, \tag{25}$$

$$u(x) = \sum_{i=1}^{N_x} w_i I_i^{(0)}(x) + c_1 \frac{x^{q-1}}{(q-1)!} + c_2 \frac{x^{q-2}}{(q-2)!} + \cdots + c_{q-1} x + c_q, \tag{26}$$

where N_x is the number of nodes on the grid line. The IRBF coefficients in (23)-(26) can be replaced with nodal values of u by solving the conversion system $\widehat{u} = \mathcal{C}\widehat{w}$, where

$$\mathcal{C} = \begin{bmatrix} I_1^{(0)}(x_1), & I_2^{(0)}(x_1), & \cdots, & I_{N_x}^{(0)}(x_1), & \frac{x_1^{q-1}}{(q-1)!}, & \cdots, & x_1, & 1 \\ I_1^{(0)}(x_2), & I_2^{(0)}(x_2), & \cdots, & I_{N_x}^{(0)}(x_2), & \frac{x_2^{q-1}}{(q-1)!}, & \cdots, & x_2, & 1 \\ & \cdots & & \cdots & & \cdots & & \\ I_1^{(0)}(x_{N_x}), & I_2^{(0)}(x_{N_x}), & \cdots, & I_{N_x}^{(0)}(x_{N_x}), & \frac{x_{N_x}^{q-1}}{(q-1)!}, & \cdots, & x_{N_x}, & 1 \end{bmatrix},$$

$$\widehat{u} = (u_1, u_2, \cdots, u_{N_x})',$$

$$\widehat{w} = (w_1, w_2, \cdots, w_{N_x}, c_1, c_2, \cdots, c_q)'.$$

Derivatives of second order and higher can now be computed using nodal values of u on the grid line only. On the ith horizontal grid line, the expressions are

$$\frac{\partial^2 u_{i,j}}{\partial x^2} = \left[I_1^{(2)}(x_{i,j}), \cdots, I_{N_x}^{(2)}(x_{i,j}), \ \cdots \right] \mathcal{C}^{-1} \widehat{u}, \tag{27}$$

$$\frac{\partial^3 u_{i,j}}{\partial x^3} = \left[I_1^{(3)}(x_{i,j}), \cdots, I_{N_x}^{(3)}(x_{i,j}), \ \cdots \right] \mathcal{C}^{-1} \widehat{u}, \tag{28}$$

$$\cdots \quad \cdots \quad \cdots \tag{29}$$

$$\frac{\partial^6 u_{i,j}}{\partial x^6} = \left[I_1^{(6)}(x_{i,j}), \cdots, I_{N_x}^{(6)}(x_{i,j}), \ \cdots \right] \mathcal{C}^{-1} \widehat{u}, \tag{30}$$

where $j = (1, 2, \cdots, N_x)$. Expressions for computing derivatives on a vertical grid line can be

acquired in a similar way. As in our previous works, the RBF width here is simply chosen as $a_i = \beta_2 h$. For non-rectangular domains, N_x and N_y become variable values. For the steady advection diffusion equation, the convection terms are taken to be the forcing function $f(x)$ (i.e. the right-hand side of Poisson equation).

3 Implementation notes

The proposed stencils result in a sparse system matrix, which has 3 non-zero entries per row (tri-diagonal) for 1D problems and 5 non-zero entries per row (sparse) for 2D problems. These algebraic systems can be solved in an efficient way. On the other hand, the proposed method involves a Picard-type iteration scheme to find derivative values at the two end-nodes of the stencil. In addition, to achieve a high level of accuracy, the method needs to use extended precision arithmetic for computing the inverse of local conversion matrices in the IRBF formulation. In this study, the matrix inverse is computed with 32 digits of precision, and the computations of other tasks such as forming the conversion matrix, computing derivatives, solving the final system of algebraic equations are all carried out with standard double precision. It was observed that the total CPU times used by our Matlab code for finding the IRBF coefficients in (15) and (17) were small: 0.028161 seconds and 0.022366 seconds, respectively, where the computations were carried out using function vpa on a Laptop with an CPU Intel(R) Core(TM) i7-7660U 2.50GHz 2.50GHz. For the region discretised with a uniform grid, one needs to compute the matrix inversion on only one stencil, and the obtained results are then used for other stencils and for every iterative step. They can be taken as a preprocessing step. Also, the final system matrix stays the same during the iteration process. For non-linear problems, an iteration scheme is needed for any numerical method, and the computation of derivatives in the present formulation is simply embedded in those iterative steps without extra loop required.

4 Numerical results

The solution accuracy is measured in the form of relative L_2-norm

$$N_e = \frac{\sum_{i=1}^{N} (u_i - u_i^e)^2}{\sum_{i=1}^{N} (u_i^e)^2}, \tag{31}$$

where u and u^e are the approximate and exact solutions, respectively. For the RBF width, we take $a_i = \beta h$, where β is a positive scaler and h is the size of the grid.

4.1 1D problems

4.1.1 Steady diffusion equation

The proposed method is first verified in

$$\frac{d^2u}{dx^2} = -(9975\sin(100x) + 1000\cos(100x))\exp(-5x), \quad 0 \le x \le 1, \tag{32}$$

whose exact solution is $u^e(x) = \sin(100x)\exp(-5x)$ (Figure 1).

Uniform grids: For an RBF solution, one can also control its accuracy through the width of RBFs. The effects of grid size and RBF width on the solution accuracy are both studied here. Results concerning grid size adaptability together with those by some other techniques are shown in Figure 2. To study the grid-size effect carefully, the number of nodes is increased in small increments of 2. It can be seen that the proposed method yields a very fast convergence, apparently as $O(h^{13.55})$ which is much higher than $O(h^{1.87})$ of the finite difference (FD), $O(h^{4.07})$ of the compact FD [29], and $O(h^{4.23})$ of the compact IRBF [25] methods. Figure 3 displays the effect of the RBF width, measured through β, on the solution accuracy. It can be seen that varying β can also improve the solution accuracy.

For the case of Dirichlet-Neumann boundary conditions, high rates of convergence are also obtained as shown in Figure 4. A rate of 13.94 is achieved using $\beta = 3$. However, their levels

of accuracy are lower when comparing to the case of Dirichlet-Dirichlet boundary conditions.

Non-uniform grids: Such grids are generated here using

$$x_i = \frac{1}{2} + \alpha \sinh\left(\eta\left(1 - \frac{i-1}{N-1}\right) + \theta\frac{i-1}{N-1}\right), \tag{33}$$

where α is a given number, $i = (1, 2, \cdots, N)$, N the number of grid nodes, $\eta = \sinh^{-1}(-1/2\alpha)$ and $\theta = \sinh^{-1}(1/2\alpha)$. This transformation produces the points that are concentrated near $x = 1/2$. The smaller the value of α the more non-uniform the grid will be. We consider 3 values $\alpha = (2, 1/2, 1/3)$ employed with the same value of β ($\beta = 3$). Grids are generated with small increments of 2: $N = (35, 37, \cdots, 161)$ to study the effect of grid size on accuracy. The solution converges as $O(h^{12.35})$ for $\alpha = 2$, $O(h^{13.47})$ for $\alpha = 1/2$, and $O(h^{13.35})$ for $\alpha = 1/3$, where $h = (x_B - x_A)/(N-1) = 1/(N-1)$ - the average grid size (Figure 5).

4.1.2 Steady advection-diffusion equation

We solve the steady constant-velocity advection diffusion equation

$$-\frac{d^2u}{dx^2} + \lambda\frac{du}{dx} = 0, \quad 0 \le x \le 1, \tag{34}$$

with boundary conditions: $u(0) = 0$ and $u(1) = 1$. The exact solution to this problem is $u^e(x) = (1 - \exp(\lambda x))/(1 - \exp(\lambda))$. Four different numerical schemes, namely the upwind differencing (UDS), central differencing (CDS), high-order compact differencing (cFDM) [30] and proposed CIRBFs, are employed. Results for $\lambda = 50$ are shown in Figure 6. The numerical solution converges apparently as $O(h^{1.34})$ for UDS, $O(h^{1.77})$ for CDS, $O(h^{3.70})$ for cFDM and $O(h^{10.20})$ for the proposed method. In addition to the achievement of much greater accuracy, Figure 7 shows that the proposed method has the ability to suppress spurious numerical oscillations that arise in CDS.

4.2 2D problems

Both linear and nonlinear problems, and both rectangular and non-rectangular domains are considered to verify the proposed method.

4.2.1 Rectangular domain

The PDE is chosen as

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 4(1-\pi^2)\sin(2\pi x)\sinh(2y) + 16(1-\pi^2)\cosh(4x)\cos(4\pi y), \tag{35}$$

defined on $-1/2 \leq x, y \leq 1/2$ and subjected to the Dirichlet boundary conditions. The exact solution to this problem is

$$u^{(e)}(x,y) = \sin(2\pi x)\sinh(2y) + \cosh(4x)\cos(4\pi y),$$

from which boundary values of u can be derived. A global one-dimensional IRBF approximation scheme with $q = 6$ is employed to compute the derivatives of second order and higher along the grid lines. In comparison with 1D problems, there is one extra parameter here: the RBF width (a) used on the grid lines, which is measured through β_2 ($a = \beta_2 h$). To carefully study the effect of grid size, grids with small increments are employed: $(11\times11, 13\times13, \cdots, 41\times41)$. For a grid of 21×21, $\beta = 1$ and $\beta_2 = 1$, the convergence behaviour of the Picard iteration scheme is shown in Figure 8. To reach $CM = 10^{-12}$, the scheme requires 41, 55, 78, 123 and 259 iteration steps for $\zeta = 1$, $\zeta = 0.8$, $\zeta = 0.6$, $\zeta = 0.4$ and $\zeta = 0.2$, respectively. A larger value of ζ yields a faster convergence. Convergence rates of the solution with respect to the grid size for several values of β_2 are shown in Figure 9. The proposed method achieves a convergence of $O(h^{7.34})$ for $\beta_2 = 1$, $O(h^{8.46})$ for $\beta_2 = 3$ and $O(h^{9.26})$ for $\beta_2 = 5$. It is noted that a high rate of convergence is achieved with a simple choice of the RBF width/shape-parameter ($a_i = h$). Figure 10 shows a comparison of convergence rates of different numerical methods. The proposed method ($O(h^{9.26})$) clearly

outperforms the central differences ($O(h^{1.87})$) and the compact IRBF method ($O(h^{3.94})$) [25].

4.2.2 Non-rectangular domain

Consider a concentric annulus as shown in Figure 11, where an inner circle is of radius R and an outer square is of side length L. Here, we take $R/L = 0.125$. The domain is simply discretised using a Cartesian grid of density $N_x \times N_y$. Grid nodes inside the physical domain are taken to be interior grid nodes, except those that are within a small distance of $h/6$ to the inner boundary. Boundary points are generated from the intersection of the grid lines and the boundaries.

The PDE chosen is also Poisson equation with its exact solution being

$$u^e(x, y) = e^{-(x-0.25)^2-(y-0.5)^2} \sin \pi x \cos 2\pi y, \tag{36}$$

from which one can derive its forcing function $f(x, y)$ and boundary conditions. A global one-dimensional IRBF approximation scheme with $q = 6$ is employed to compute derivatives of second order and higher on the grid lines. Value of β is simply set to 1.

The convergence behaviour of the Picard scheme for a grid of 18×18 and $\beta_2 = 1$ is shown in Figure 12, where all derivative values are set to zero at the first iteration step. It can be seen that CM is consistently reduced with the number of iteration steps. To reach $CM < 10^{-12}$, the scheme only requires 74, 102, 160 and 333 iteration steps for $\zeta = 0.8$, $\zeta = 0.6$, $\zeta = 0.4$ and $\zeta = 0.2$, respectively. The larger the value of α the faster the convergence will be.

To study the grid size effect carefully, the number of nodes in each direction is increased in small increments of 2: $(10 \times 10, 12 \times 12, \cdots, 50 \times 50)$. Results concerning the dependence of solution accuracy on grid size for several values of β_2 are shown in Figure 13. It can be seen that the error is consistently reduced at small values of h. The proposed method achieves a convergence of $O(h^{7.85})$ for $\beta = 0.1$, $O(h^{8.22})$ for $\beta = 0.5$ and $O(h^{9.02})$ for $\beta = 1$. As in the case of rectangular domains, a high rate of convergence is achieved here with a simple choice

of the RBF width/shape-parameter ($a_i = h$).

4.2.3 Natural convection in annulus domain

The dimensionless governing equations for 2D steady-state buoyancy-driven flows can be written in terms of the streamfunction ψ, vorticity ω and temperature T as

$$\frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial y^2} = \omega, \tag{37}$$

$$\frac{\partial^2\omega}{\partial x^2} + \frac{\partial^2\omega}{\partial y^2} = \sqrt{\frac{Ra}{Pr}}\left(u\frac{\partial\omega}{\partial x} + v\frac{\partial\omega}{\partial y}\right) + \sqrt{\frac{Ra}{Pr}}\frac{\partial T}{\partial x}, \tag{38}$$

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = \sqrt{RaPr}\left(u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y}\right), \tag{39}$$

where u and v are the x and y components of the velocity vector, which are defined by

$$u = \frac{\partial\psi}{\partial y}, \quad v = -\frac{\partial\psi}{\partial x},$$

and Pr and Ra are the Prandtl and Rayleigh numbers defined as $Pr = \nu/\alpha$ and $Ra = \gamma g \Delta T L^3/\alpha\nu$, respectively in which ν is the kinematic viscosity, α the thermal diffusivity, γ the thermal expansion coefficient, g the gravity, and L and ΔT the characteristic length and temperature difference, respectively. In this dimensionless scheme, the velocity scale is taken as $U = \sqrt{gL\gamma\Delta T}$ for the purpose of balancing the buoyancy and inertial forces.

Consider natural convection in a concentric annulus between a heated inner circular cylinder of radius R and a cooled square enclosure of side length L (Figure 11). Simulations with $R/L = 0.2$ and $Pr = 0.71$ are carried out on three uniform Cartesian grids, namely 32×32, 36×36 and 40×40, for a wide range of Ra, (10^4, 10^5, 10^6).

For (37), due to the flow symmetry, the streamfunction ψ is set to zero on both the inner and outer boundaries. For (39), the temperature T is set to 1 on the inner boundary and 0 on the outer boundary.

For (38), the boundary conditions are derived from (37). On the outer, (37) reduces to

$\omega = \partial^2\psi/\partial x^2$ on $x = -L/2$ and $x = L/2$, and $\omega = \partial^2\psi/\partial y^2$ on $y = -L/2$ and $y = L/2$. On the inner, we utilise analytic formulas proposed in [31], i.e.

$$\omega = \left[1 + \left(\frac{y}{x}\right)^2\right] \frac{\partial^2\psi}{\partial x^2} \tag{40}$$

for boundary points that lie on the x-grid lines, and

$$\omega = \left[1 + \left(\frac{x}{y}\right)^2\right] \frac{\partial^2\psi}{\partial y^2} \tag{41}$$

for boundary points that lie on the y-grid lines.

The solution procedure can be summarised as follows

1. Guess initial values of the streamfunction, vorticity and temperature fields. For the Stokes flow ($Ra = 0$), the simulation starts from rest. For $Ra > 0$, the solution at the lower and nearest Ra is taken as an initial guess.

2. Discretise equations (37)-(39), where the RHSs are taken to be forcing functions (the right-hand side of Poisson equation). It is noted that the three equations have the same system matrix.

3. Solve (39) for interior nodal values of T.

4. Relax the temperature field according to (22).

5. Compute the 1st-, 2nd-, 3rd-, 4th-, 5th- and 6th-order derivatives of T on the grid lines.

6. Solve (37) for interior nodal values of ψ.

7. Relax the streamfunction field.

8. Compute the 1st-, 2nd-, 3rd-, 4th-, 5th- and 6th-order derivatives of ψ on the grid lines.

9. Derive a computational boundary condition for ω.

10. Solve (38) for interior nodal values of ω.

11. Relax the vorticity field.

12. Compute the 1st-, 2nd-, 3rd-, 4th-, 5th- and 6th-order derivatives of ω on the grid lines.

13. Compute CM, where "sol" is composed of nodal values of T, ψ and ω.

14. Check $CM < 10^{-12}$. If it is not satisfied, repeat from Step 3.

15. Otherwise, stop the calculation and output the results.

For this problem, value of β and β_2 are simply set to 1. In Step 9, following [31], a global one-dimensional IRBF approximation scheme with $q = 2$ is employed to estimate the vorticity boundary conditions. The non-slip boundary conditions at the inner and outer boundaries (i.e. $v_x = 0$ and $v_y = 0$) lead to $\partial\psi/\partial x = 0$ and $\partial\psi/\partial y = 0$. On the x-grid lines, the vorticity boundary conditions are estimated using (40), where boundary values $\partial\psi/\partial x = 0$ are incorporated into the IRBF approximation of $\partial^2\psi/\partial x^2$ by means of integration constants. On the y-grid lines, the vorticity boundary conditions are estimated using (41), where boundary values $\partial\psi/\partial x = 0$ are incorporated into the IRBF approximation of $\partial^2\psi/\partial y^2$ by means of integration constants.

To be consistent with the boundary estimation, a global one-dimensional IRBF approximation scheme in Steps 5, 8 and 12 are also employed with $q = 2$. To compute the 3rd- and 4th-order derivatives, the 2nd-order derivatives are taken as functions, and to compute the 5rd- and 6th-order derivatives, the 4th-order derivatives are taken as functions.

The obtained results are presented in Figures 14 and 15, and Table 1. Results concerning the convergence behaviour of the Picard scheme for $Ra = 10^4$, $Ra = 10^5$ and $Ra = 10^6$ on a grid of 32×32 are shown in Figure 14, where the value CM is seen to reduce quite monotonically and a higher Ra case requires more iteration steps. Figure 15 presents contour plots of the streamfunction and temperature fields for $Ra = 10^4$, $Ra = 10^5$ and $Ra = 10^6$ acquired from a coarse grid of 32×32. There are 21 lines whose value varies linearly in each contour plot.

The distributions look feasible when compared to those by the finite-volume method [32] and the differential-quadrature method [33]. The proposed method is able to capture thin boundary layers of the temperature field at high values of Ra using a relatively coarse grid. Table 1 displays estimated values of the average Nusselt number defined as

$$\overline{N}_u = \frac{\overline{H}}{\kappa}, \tag{42}$$

where $\overline{H} = -(\kappa/\Delta T) \oint \partial T/\partial n ds$, κ the thermal conductivity and ΔT the temperature difference between the inner and outer boundaries. To compare with those in [32] and [33], where the computational domain is taken as one-half of the physical domain owing to the flow symmetry, the computed values of $\overline{N}_u$ in the present work (Table 1) are divided by 2. Good agreement is achieved.

5 Concluding remarks

This paper presents a new Cartesian-grid-based RBF method, where the system matrix has only 3 non-zero entries per row (tridiagonal) for 1D problems and 5 non-zero entries per row (sparse) for 2D problems. A salient feature of the proposed method lies in the inclusion of nodal values of the 2nd-, 3rd-, 4th, 5th- and 6th-order derivatives in the RBF approximations. The method needs a Picard-type iteration scheme to compute the imposed derivatives and the use of higher-precision arithmetic to compute relatively-small conversion matrices. The proposed method is verified in linear and non-linear problems with rectangular and non-rectangular domains, and uniform and nonuniform grids. For non-linear problems, like other numerical techniques, the proposed method involves only one loop to solve the algebraic system. New compact approximations allow the order of grid convergence of 3-point stencils in 1D and 5-point stencils in 2D to be higher than 4. High rates of convergence are acquired even with a simple choice of the RBF-width/shape-parameter (the nodal-distance/grid-size taken as the RBF width). In the simulation of natural convection in an annulus domain, the flow behaviour is well captured using relatively-coarse grids.

Appendix

The following are integrated basis functions derived from the multiquadric function by using Mathematica

$$I_i^{(1)}(x) = \frac{(x-x_i)}{2}A + \frac{a_i^2}{2}B,$$

$$I_i^{(2)}(x) = \left(\frac{-a_i^2}{3} + \frac{(x-x_i)^2}{6}\right)A + \frac{a_i^2(x-x_i)}{2}B,$$

$$I_i^{(3)}(x) = \left(\frac{-13a_i^2(x-x_i)}{48} + \frac{(x-x_i)^3}{24}\right)A + \left(\frac{-a_i^4}{16} + \frac{a_i^2(x-x_i)^2}{4}\right)B,$$

$$I_i^{(4)}(x) = \left(\frac{a_i^4}{45} - \frac{83a_i^2(x-x_i)^2}{720} + \frac{(x-x_i)^4}{120}\right)A + \left(\frac{-3a_i^4(x-x_i)}{48} + \frac{4a_i^2(x-x_i)^3}{48}\right)B,$$

$$I_i^{(5)}(x) = \left(\frac{113a_i^4(x-x_i)}{5760} - \frac{97a_i^2(x-x_i)^3}{2880} + \frac{(x-x_i)^5}{720}\right)A + \left(\frac{a_i^6}{384} - \frac{3a_i^4(x-x_i)^2}{96} + \frac{2a_i^2(x-x_i)^4}{96}\right)B,$$

$$I_i^{(6)}(x) = \left(\frac{-a_i^6}{1575} + \frac{593a_i^4(x-x_i)^2}{67200} - \frac{253a_i^2(x-x_i)^4}{33600} + \frac{(x-x_i)^6}{5040}\right)A +$$
$$\left(\frac{5a_i^6(x-x_i)}{1920} - \frac{20a_i^4(x-x_i)^3}{1920} + \frac{8a_i^2(x-x_i)^5}{1920}\right)B,$$

$$I_i^{(7)}(x) = \left(\frac{-1873a_i^6(x-x_i)}{3225600} + \frac{4327a_i^4(x-x_i)^3}{1612800} - \frac{551a_i^2(x-x_i)^5}{403200} + \frac{(x-x_i)^7}{403200}\right)A +$$
$$\left(\frac{-a_i^8}{18432} + \frac{15a_i^6(x-x_i)^2}{11520} - \frac{30a_i^4(x-x_i)^4}{11520} + \frac{8a_i^2(x-x_i)^6}{11520}\right)B,$$

$$I_i^{(8)}(x) = \left(\frac{a_i^8}{99225} - \frac{54511a_i^6(x-x_i)^2}{203212800} + \frac{20939a_i^4(x-x_i)^4}{33868800} - \frac{5309a_i^2(x-x_i)^6}{25401600} + \frac{(x-x_i)^8}{362880}\right)A +$$
$$\left(\frac{-35a_i^8(x-x_i)}{645120} + \frac{280a_i^6(x-x_i)^3}{645120} - \frac{336a_i^4(x-x_i)^5}{645120} + \frac{64a_i^2(x-x_i)^7}{645120}\right)B,$$

$$I_i^{(9)}(x) = \left(\frac{30563a_i^8(x-x_i)}{3251404800} - \frac{135389a_i^6(x-x_i)^3}{1625702400} + \frac{15569a_i^4(x-x_i)^5}{135475200} - \frac{5617a_i^2(x-x_i)^7}{203212800} + \frac{(x-x_i)^9}{3628800}\right)A +$$
$$\left(\frac{a_i^{10}}{1474560} - \frac{35a_i^8(x-x_i)^2}{1290240} + \frac{140a_i^6(x-x_i)^4}{1290240} - \frac{112a_i^4(x-x_i)^6}{1290240} + \frac{16a_i^2(x-x_i)^8}{1290240}\right)B,$$

$$I_i^{(10)}(x) = \left(-\frac{a_i^{10}}{9823275} + \frac{284021a_i^8(x-x_i)^2}{64377815040} - \frac{41885a_i^6(x-x_i)^4}{2145927168} + \frac{721109a_i^4(x-x_i)^6}{40236134400} - \frac{64811a_i^2(x-x_i)^8}{20118067200} + \frac{(x-x_i)^{10}}{39916800}\right)A +$$
$$\left(\frac{63a_i^{10}(x-x_i)}{92897280} - \frac{840a_i^8(x-x_i)^3}{92897280} + \frac{2016a_i^6(x-x_i)^5}{92897280} - \frac{1152a_i^4(x-x_i)^7}{92897280} + \frac{128a_i^2(x-x_i)^9}{92897280}\right)B$$

where $A = \sqrt{(x-x_i)^2 + a_i^2}$ and $B = \ln\left((x-x_i) + \sqrt{(x-x_i)^2 + a_i^2}\right)$.

References

1. Liu GR. Mesh free methods: moving beyond the finite element method. 2nd ed. Boca Raton: CRC Press; 2018.

2. Shortley GH, Weller R. The numerical solution of Laplaces equation. J. Appl. Phys. 1938;9:334-348.

3. Jomaa Z, Macaskill C. The embedded finite difference method for the Poisson equation in a domain with an irregular boundary and Dirichlet boundary conditions. Journal of Computational Physics 2005;202(2):488-506.

4. Johansen H, Colella P. A Cartesian grid embedded boundary method for poisson's equation on irregular domains. Journal of Computational Physics 1998;147(1):60-85.

5. Mai-Duy N, Tanner RI. Computing non-Newtonian fluid flow with radial basis function networks. International journal for numerical methods in fluids 2005;48(12):1309-1336;

6. Mai-Duy N, Tran-Cong T. A multidomain integrated-radial-basis-function collocation method for elliptic problems. Numerical Methods for Partial Differential Equations 2008;24(5):1301-1302.

7. Harris MF, Kassab AJ, Divo E. A shock-capturing meshless scheme using RBF blended interpolation and moving least squares. Engineering Analysis with Boundary Elements 2019;109:81-93.

8. Jankowska MA, Karageorghis A. Variable shape parameter Kansa RBF method for the solution of nonlinear boundary value problems. Engineering Analysis with Boundary Elements 2019;103:32-40.

9. Safarpoor M, Shirzadi A. A localized RBF-MLPG method for numerical study of heat and mass transfer equations in elliptic fins. Engineering Analysis with Boundary Elements 2019;98:35-45.

10. Dehghan M, Mohammadi V. Two-dimensional simulation of the damped Kuramoto-Sivashinsky equation via radial basis function-generated finite difference scheme combined with an exponential time discretization. Engineering Analysis with Boundary Elements 2019;107:168-184.

11. Mramor K, Vertnik R, Sarler B. Application of the local RBF collocation method to natural convection in a 3D cavity influenced by a magnetic field. Engineering Analysis with Boundary Elements 2020;116:1-13.

12. Deng C, Zheng H, Fu M, Xiong J, Chen CS. An efficient method of approximate particular solutions using polynomial basis functions. Engineering Analysis with Boundary Elements 2020;111:1-8.

13. Bhanot RP, Strunin DV, and Ngo-Cong D, Numerical solution of a highly nonlinear and non-integrable equation using integrated radial basis function network method. Chaos: An Interdisciplinary Journal of Nonlinear Science 2020;30:083119.

14. Kansa EJ. Multiquadrics - A scattered data approximation scheme with applications to computational fluid-dynamics - II. Solutions to parabolic, hyperbolic and elliptic partial differential equations. Computers & Mathematics with Applications 1990;19(8/9):147-161.

15. Madych WR, Nelson SA. Multivariate interpolation and conditionally positive definite functions. Approximation Theory and its Applications 1988;4:77-89.

16. Madych WR, Nelson SA. Multivariate interpolation and conditionally positive definite functions 2. Mathematics of Computation 1990;54(189):211-230.

17. Cheng AH-D. Multiquadric and its shape parameterA numerical investigation of error estimate, condition number, and round-off error by arbitrary precision computation. Engineering Analysis with Boundary Elements 2012;36(2):220-239.

18. Huang C-S, Lee C-F, Cheng AH-D. Error estimate, optimal shape factor, and high precision computation of multiquadric collocation method. Engineering Analysis with Boundary Elements 2007;31(7):614-623.

19. Sarra SA. Radial basis function approximation methods with extended precision floating point arithmetic. Engineering Analysis with Boundary Elements 2011;35(1):68-76.

20. Shu C, Ding H, Yeo KS. Local radial basis function-based differential quadraturemethod and its application to solve two-dimensional incompressible Navier-Stokes equations. Comput. Methods Appl. Mech. Engrg. 2003;192:941-954.

21. Bayona V, Moscoso M, Carretero M, Kindelan M. RBF-FD formulas and convergence properties. Journal of Computational Physics 2010;229(22):8281-8295.

22. Tolstykh AI, Shirobokov DA. On using radial basis functions in a "finite difference mode" with applications to elasticity problems. Computational Mechanics 2003;33(1):68-79.

23. Wright GB, Fornberg B. Scattered node compact finite difference-type formulas generated from radial basis functions. Journal of Computational Physics 2006;212(1):99-123.

24. Mai-Duy N, Tran-Cong T. Compact local integrated-RBF approximations for second-order elliptic differential problems. Journal of Computational Physics 2011;230(12):4772-4794.

25. Mai-Duy N, Tran-Cong T. A compact five-point stencil based on integrated RBFs for 2D second-order differential problems. Journal of Computational Physics 2013;235:302-321.

26. Mai-Duy N, Le TTV, Tien CMT, Ngo-Cong D, Tran-Cong T. Compact approximation stencils based on integrated flat radial basis functions. Engineering Analysis with Boundary Elements 2017;74:79-87.

27. Mai-Duy N, Dalal D, Le TTV, Ngo-Cong D, Tran-Cong T. A symmetric integrated radial basis function method for solving differential equations. Numerical Methods for Partial Differential Equations 2018;34(3):959-981.

28. Xiao D, Fang F, Pain CC, Navon IM. A parameterized non-intrusive reduced order model and error analysis for general time-dependent nonlinear partial differential equations and its applications. Comput. Methods Appl. Mech. Engrg. 2017;317:868-889.

29. Hirsh RS. Higher order accurate difference solutions of fluid mechanics problems by a compact differencing technique. Journal of Computational Physics 1975;19(1):90-109.

30. Spotz WF. High-order Compact Finite Difference Schemes for Computational Mechanics, Ph.D. Thesis, University of Texas at Austin, Austin, TX, 1995

31. Le-Cao K, Mai-Duy N, Tran-Cong T. An effective integrated-RBFN Cartesian-grid discretisation for the stream function-vorticitytemperature formulation in non-rectangular domains. Numerical Heat Transfer (Part B) 2009;55(6):480-502.

32. Moukalled F, Acharya S. Natural convection in the annulus between concentric horizontal circular and square cylinders. Journal of Thermophysics and Heat Transfer 1996;10(3):524-531.

33. Shu C, Zhu YD. 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 2002;38:429-445.

Table 1: Natural convection, annulus domain: the Nusselt number. Values of the Nusselt number by the finite-volume [32] and differential-quadrature [33] methods are also included for comparison purposes.

	$\overline{N}_u$		
$N_x \times Ny$	$Ra = 10^4$	$Ra = 10^5$	$Ra = 10^6$
32×32	3.2205	4.8494	10.4656
36×36	3.2218	4.8724	9.6596
40×40	3.2227	4.8851	9.1403
Ref. [32]	3.24	4.86	8.90
Ref. [33]	3.331	5.080	9.374

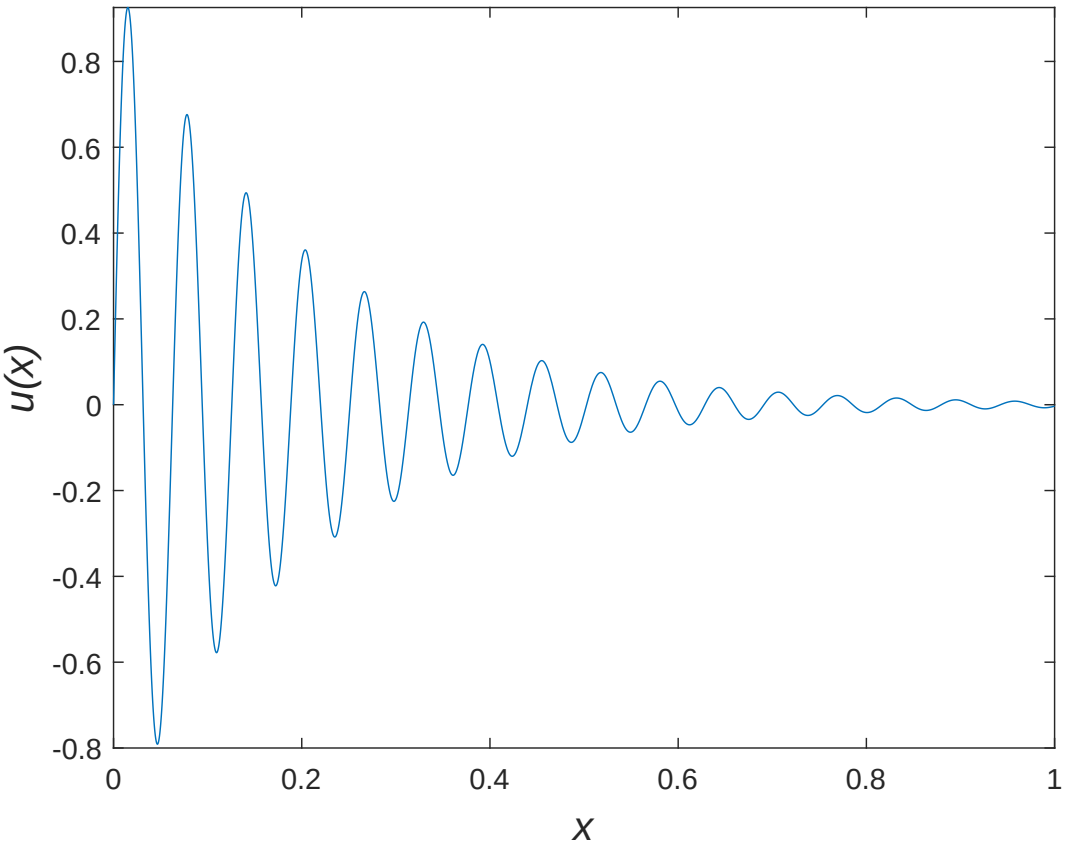


Figure 1: Steady diffusion equation: Exact solution.

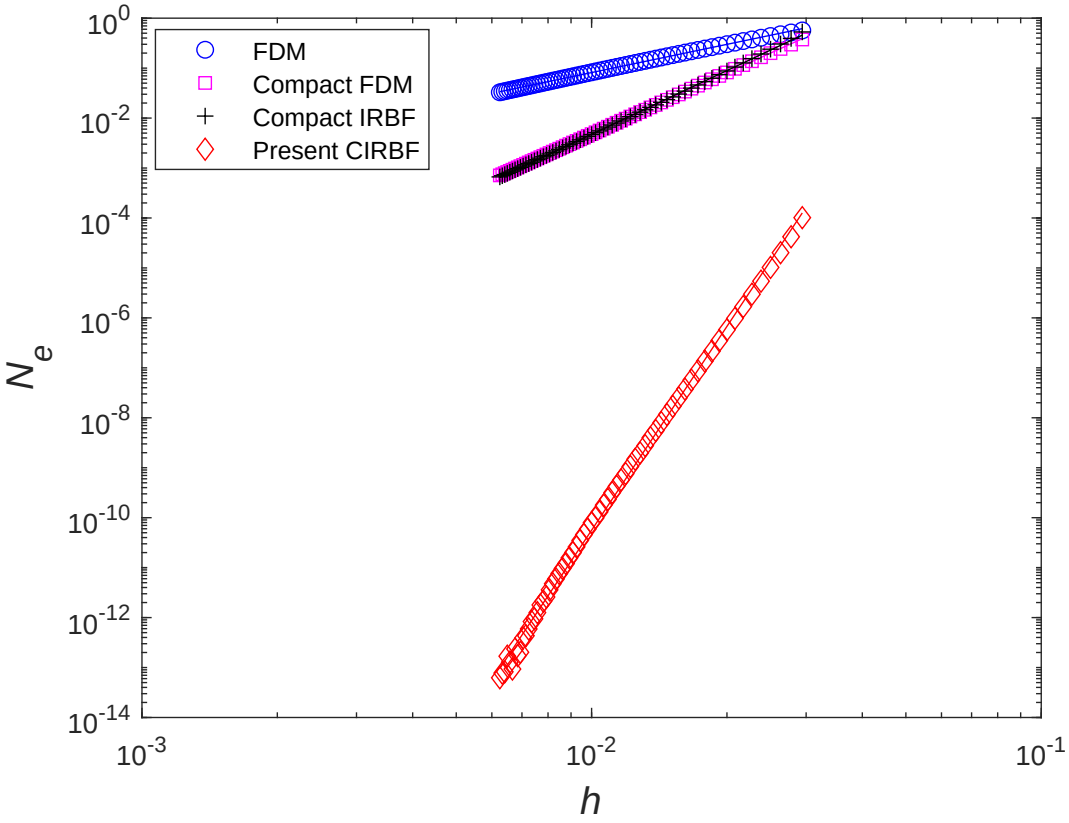


Figure 2: Steady diffusion equation, $N = (35, 37, \cdots, 161)$: Effect of the grid size on accuracy. To study the grid-size effect carefully, the number of nodes is increased in small increments of 2. The numerical solution converges apparently as $O(h^{1.87})$ for the central differences, $O(h^{4.07})$ for the compact central differences, $O(h^{4.23})$ for the compact IRBF [25] and $O(h^{13.55})$ for the present CIRBF ($\beta = 5$).

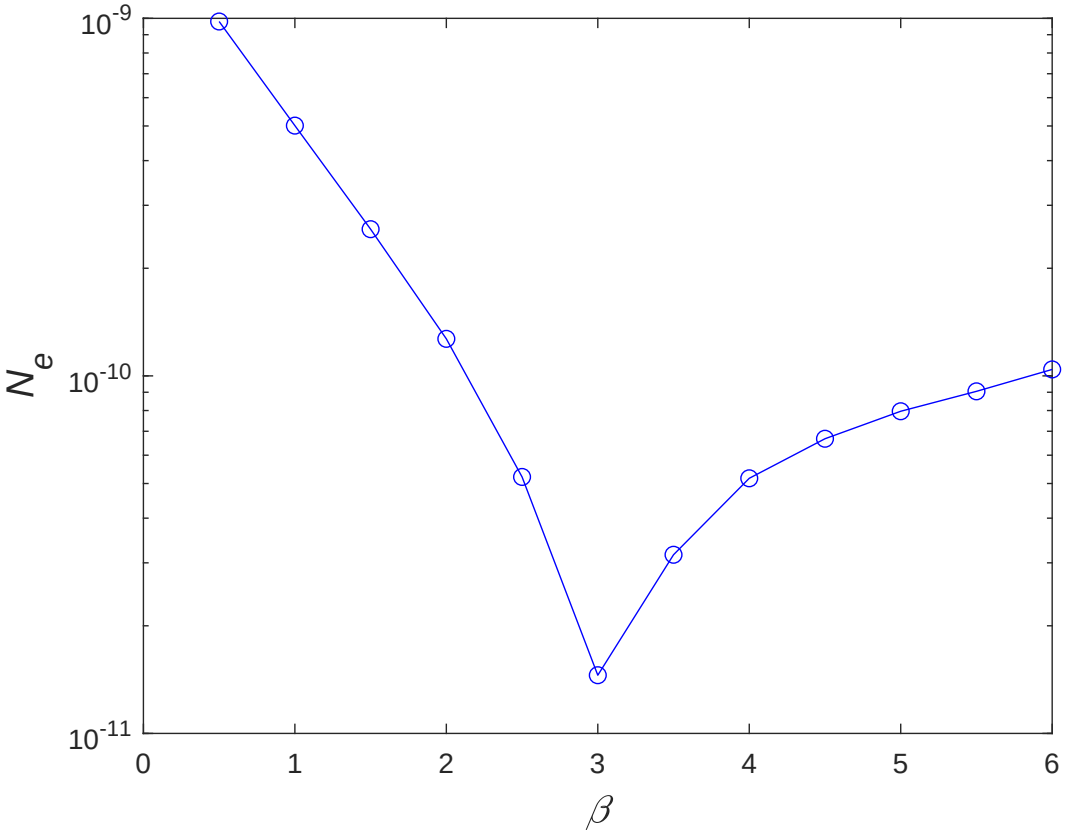


Figure 3: Steady diffusion equation, $N = 101$: Effect of the RBF width, measured through β, on numerical accuracy.

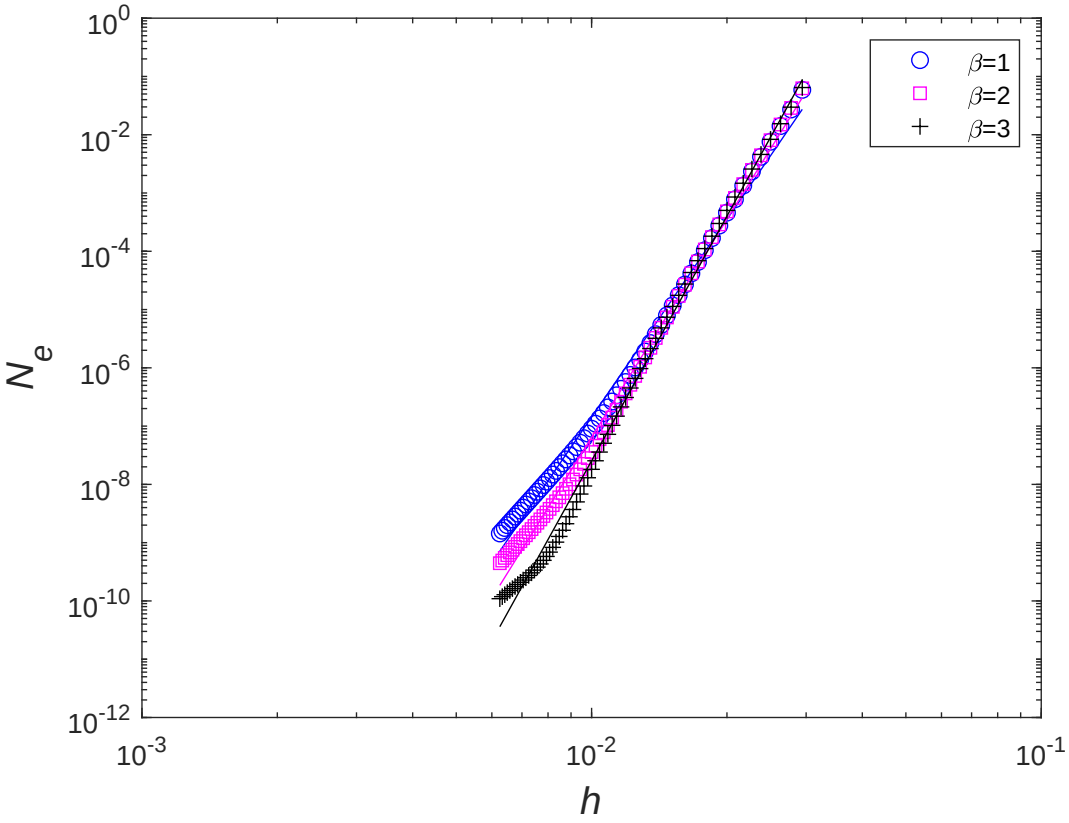


Figure 4: Steady diffusion equation, $N = (35, 37, \cdots, 161)$: With Dirichlet-Neumann boundary conditions, the numerical solution converges as $O(h^{11.27})$ for $\beta = 1$, $O(h^{12.44})$ for $\beta = 2$ and $O(h^{13.94})$ for $\beta = 3$, where h is the average grid size. When compared to Dirichlet-Dirichlet boundary conditions, high rates of convergence are also obtained here, but their levels of accuracy are lower.

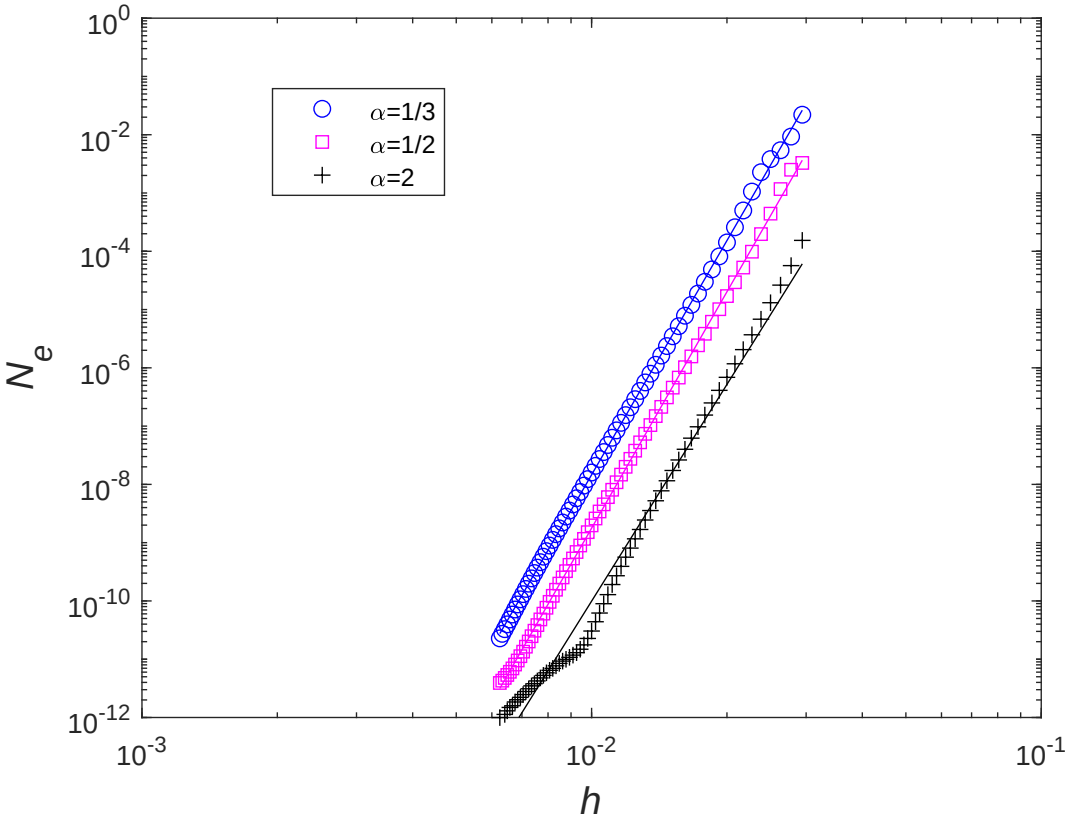


Figure 5: Steady diffusion equation, $N = (35, 37, \cdots, 161)$, $\beta = 3$: With non-uniform grids, the solution converges as $O(h^{13.35})$ for $\alpha = 1/3$, $O(h^{13.47})$ for $\alpha = 1/2$, and $O(h^{12.35})$ for $\alpha = 2$, where h is the average grid size.

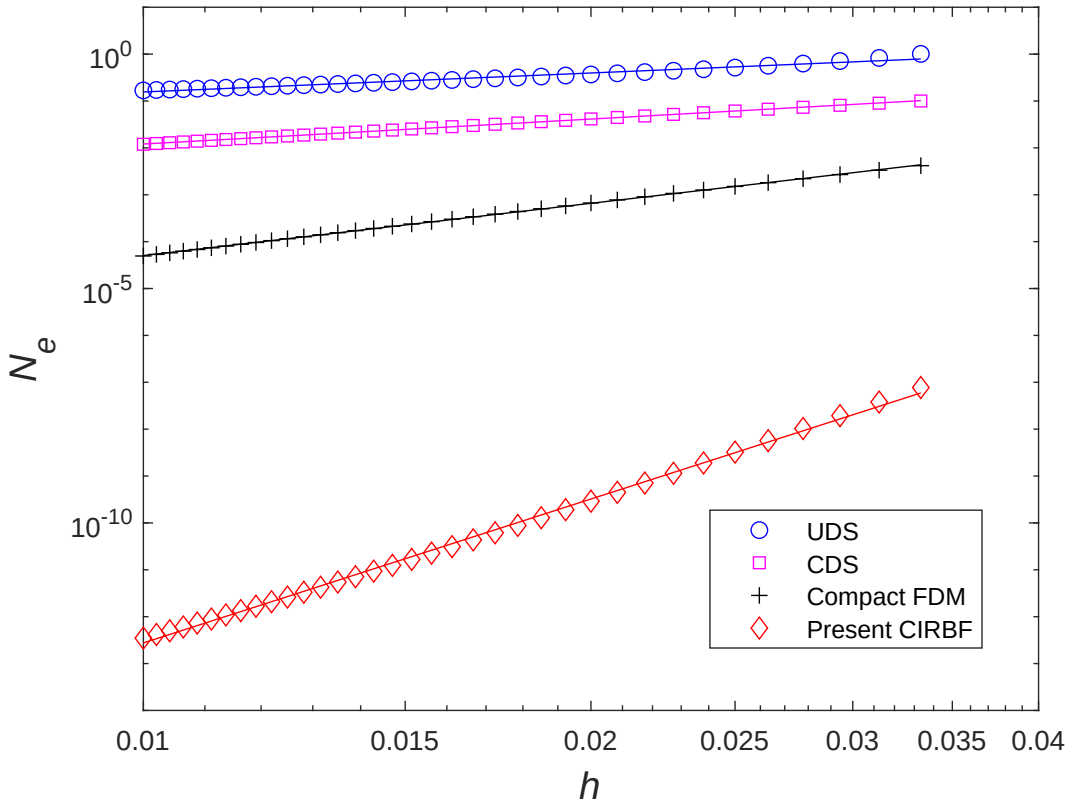


Figure 6: Steady advection-diffusion equation in 1D, $\lambda = 50$, $N = (31, 33, \cdots, 101)$: Comparison of convergence rates of different numerical methods. The numerical solution converges apparently as $O(h^{1.34})$ for the upwind difference scheme, $O(h^{1.77})$ for central difference scheme, $O(h^{3.70})$ for compact FDM and $O(h^{10.20})$ for the proposed method.

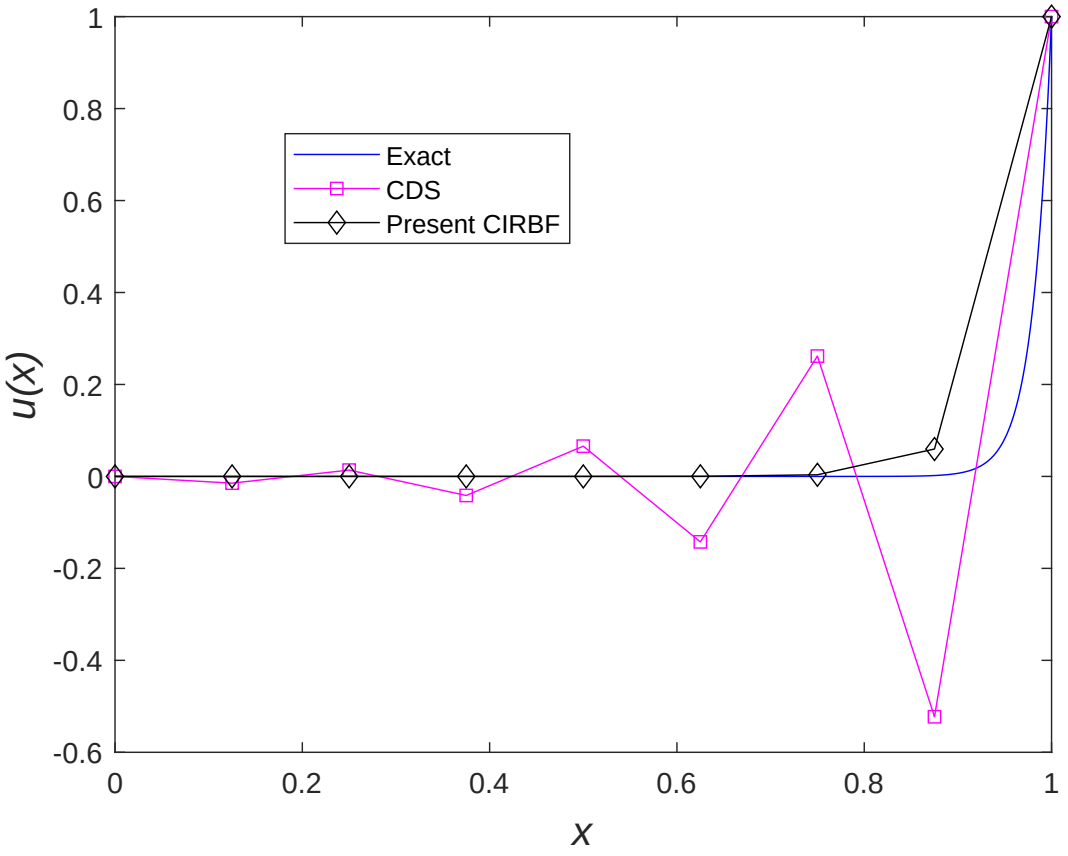


Figure 7: Steady advection-diffusion equation in 1D, $\lambda = 50$, $N = 9$: Solutions by the central difference method and the proposed method.

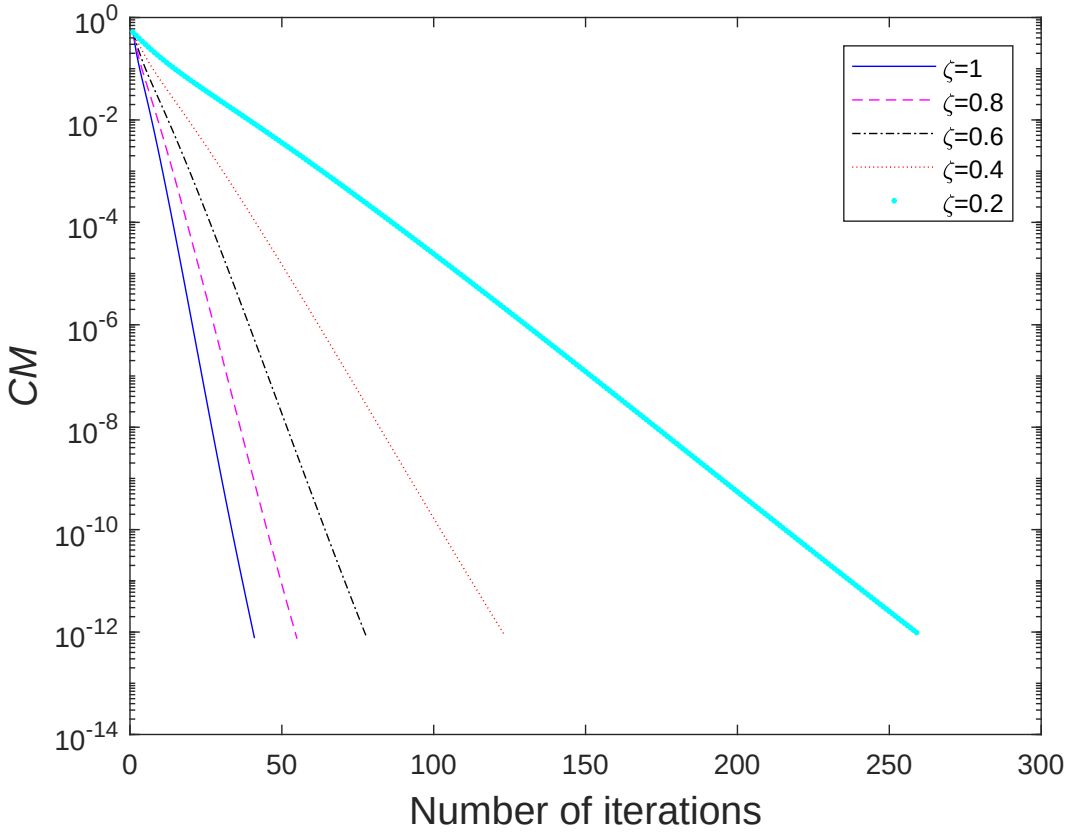


Figure 8: PDE, 21×21, $\beta = 1$, $\beta_2 = 1$, zeros as initial guess: Convergence behaviour of the Picard scheme. To reach $CM = 10^{-12}$, the scheme requires 41, 55, 78, 123 and 259 iteration steps for $\zeta = 1$, $\zeta = 0.8$, $\zeta = 0.6$, $\zeta = 0.4$ and $\zeta = 0.2$, respectively.

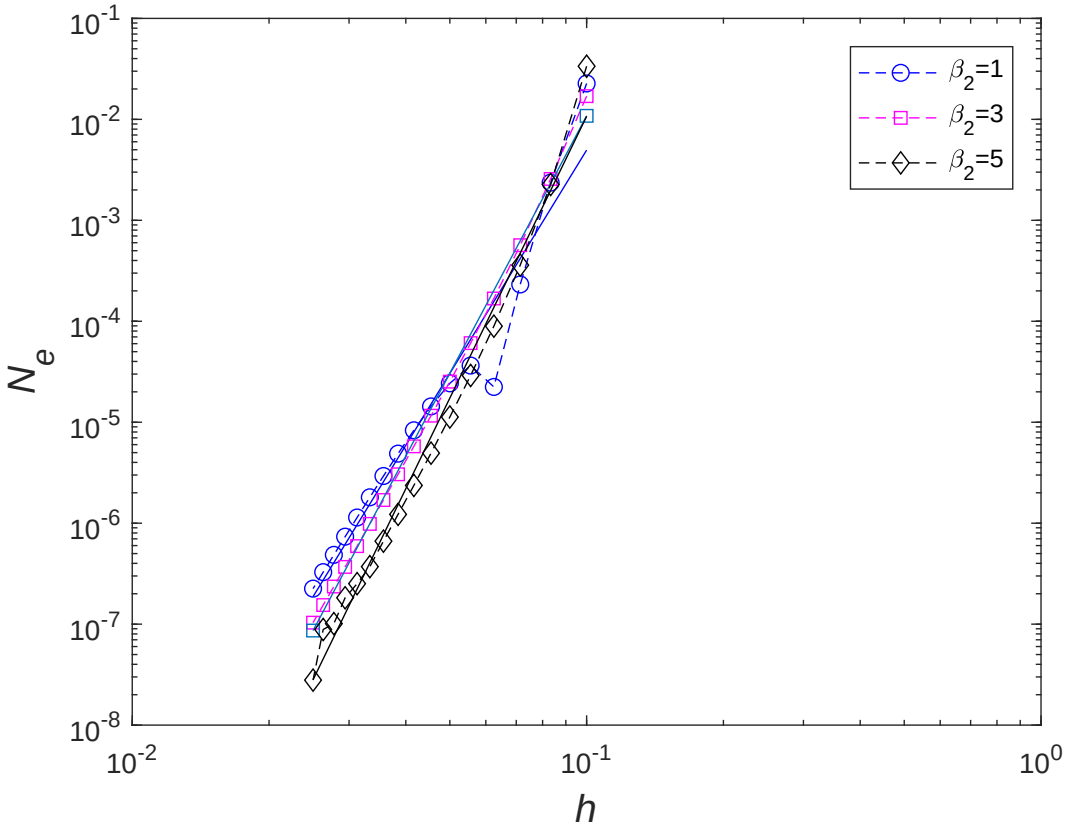


Figure 9: PDE, $N_x \times N_y = (11 \times 11, 13 \times 13, \cdots, 41 \times 41)$, $\beta = 1$: Effect of the grid size on accuracy for several values of the RBF width employed on the grid lines. To study the grid-size effect carefully, the number of nodes in each direction is increased in small increments of 2. The numerical solution converges apparently as $O(h^{7.34})$ for $\beta_2 = 1$, $O(h^{8.46})$ for $\beta_2 = 3$ and $O(h^{9.26})$ for $\beta_2 = 5$.

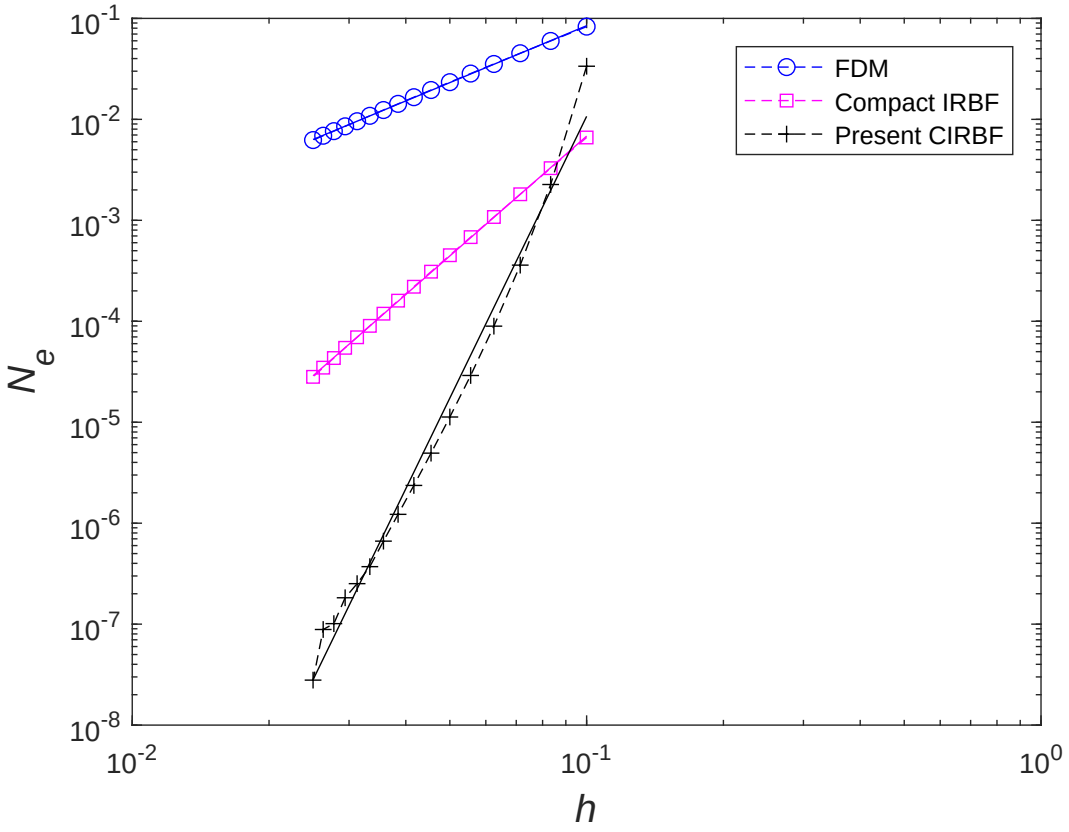


Figure 10: PDE, $N_x \times N_y = (11 \times 11, 13 \times 13, \cdots, 41 \times 41)$: Comparison of convergence rates of different numerical methods. The numerical solution converges apparently as $O(h^{1.87})$ for the central differences, $O(h^{3.94})$ for the compact IRBF [25] and $O(h^{9.26})$ for the present CIRBF.

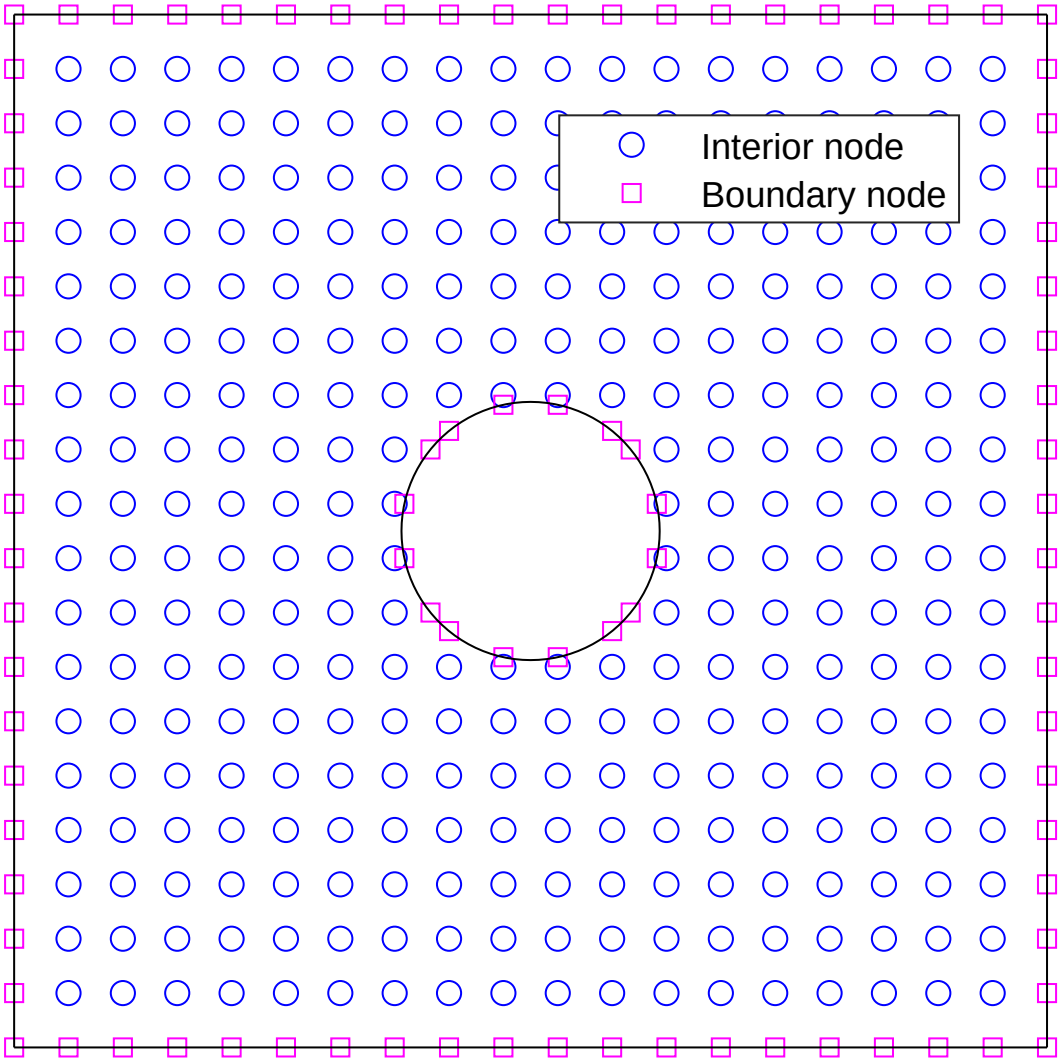


Figure 11: PDE, annulus domain ($R/L = 0.125$): Cartesian-grid-based discretisation.

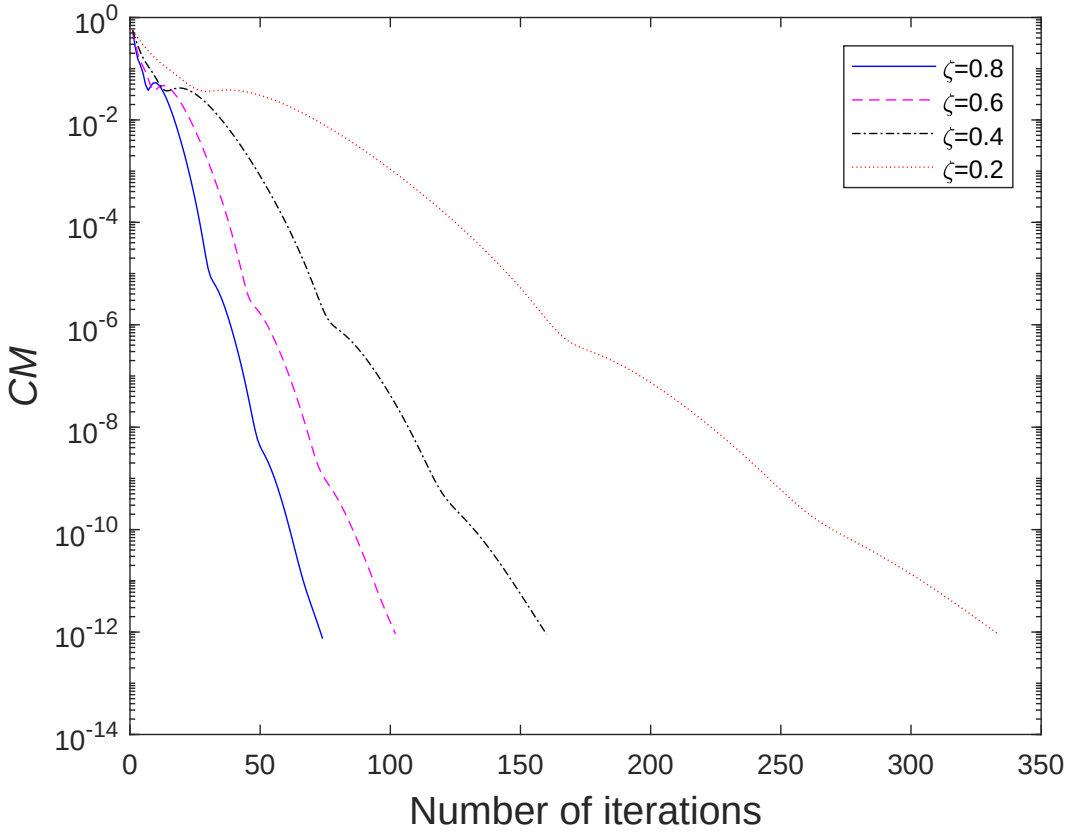


Figure 12: PDE, annulus domain ($R/L = 0.125$), $18 \times 18, \beta = 1, \beta_2 = 1$, zeros as initial guess: Convergence behaviour of the Picard scheme. To reach $CM = 10^{-12}$, the scheme requires 74, 102, 160 and 333 iteration steps for $\zeta = 0.8$, $\zeta = 0.6$, $\zeta = 0.4$ and $\zeta = 0.2$, respectively.

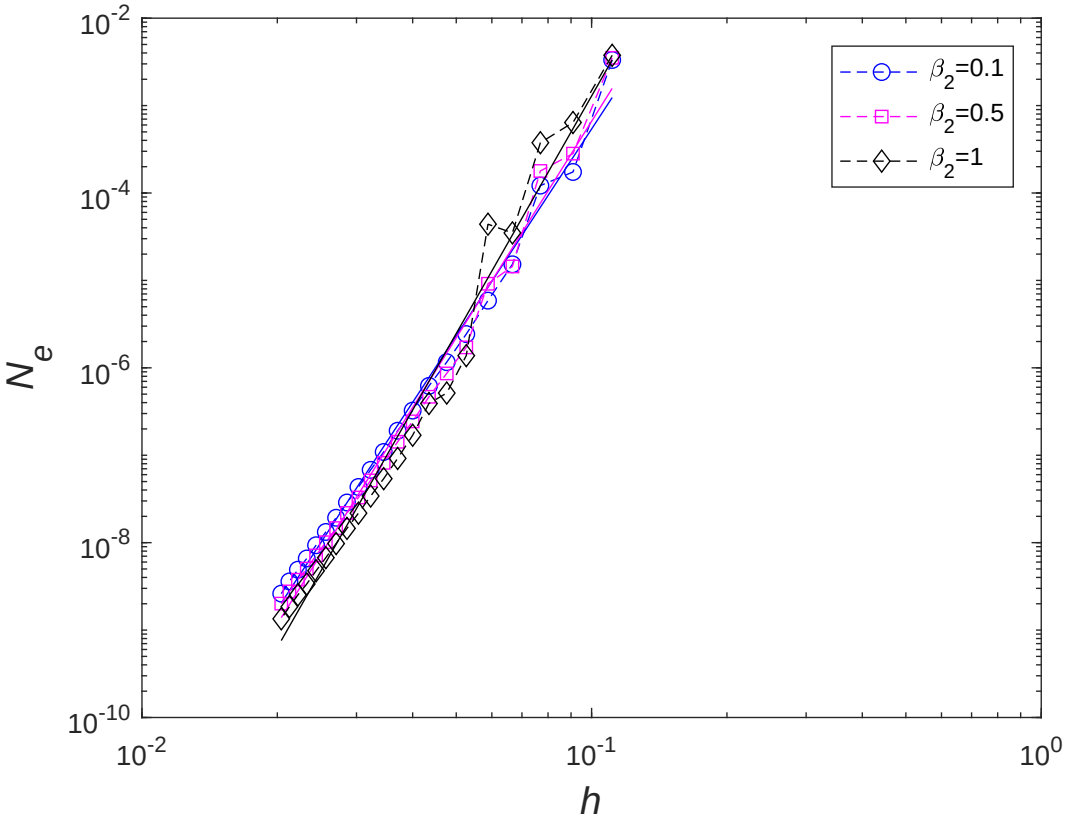


Figure 13: PDE, annulus domain, $N_x \times N_y = (10 \times 10, 12 \times 12, \cdots, 50 \times 50)$, $\beta = 1$: Effect of the grid size on accuracy for several values of the RBF width employed on the grid lines. To study the grid-size effect carefully, the number of nodes in each direction is increased in small increments of 2. The proposed method achieves a convergence of $O(h^{7.85})$ for $\beta_2 = 0.1$, $O(h^{8.22})$ for $\beta_2 = 0.5$ and $O(h^{9.02})$ for $\beta_2 = 1$.

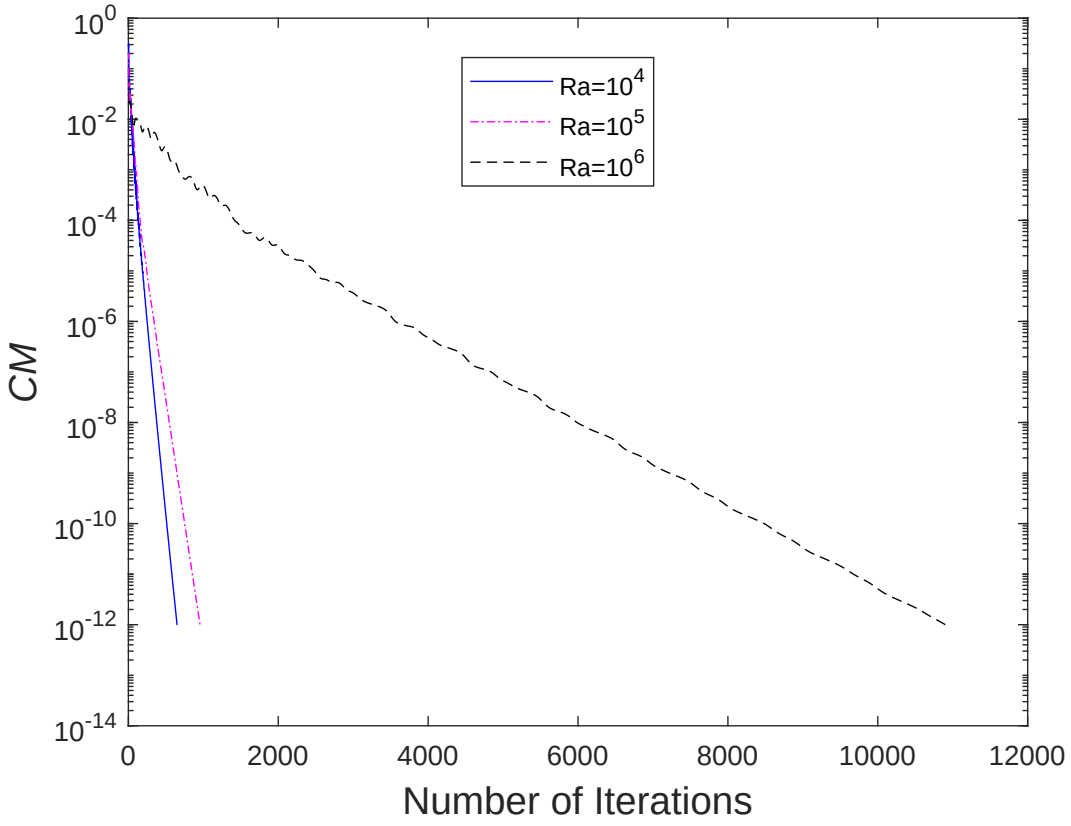
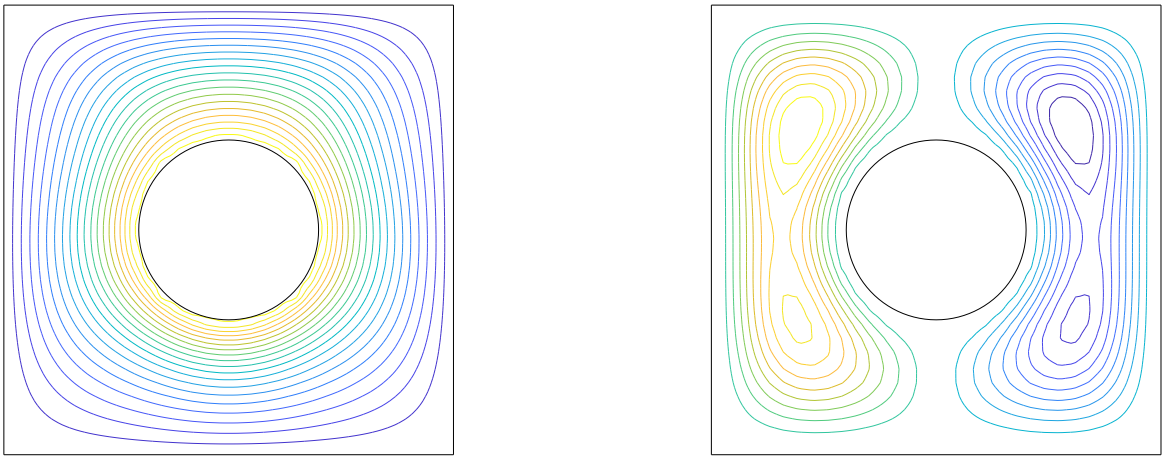
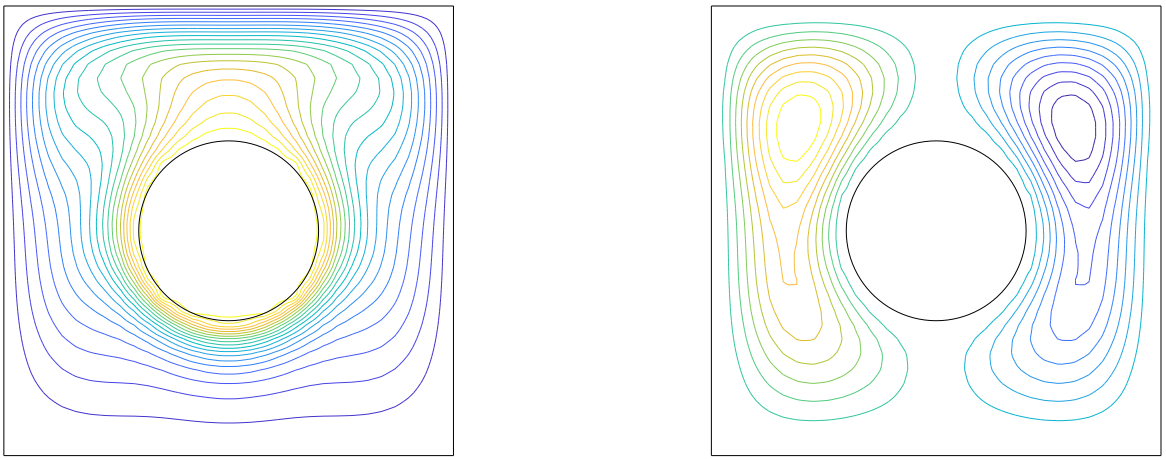


Figure 14: Natural convection, annulus domain ($R/L = 2$), 32×32, $\beta = 1$: Convergence behaviour of the Picard scheme, where the solution at lower Ra is taken as initial guess. To reach $CM = 10^{-12}$, the scheme requires 650, 957 and 10901 iteration steps for $Ra = 10^4(\zeta = 0.1)$, $Ra = 10^5(\zeta = 0.1)$ and $Ra = 10^6(\zeta = 0.01)$, respectively.

$Ra = 10^4$



$Ra = 10^5$



$Ra = 10^6$

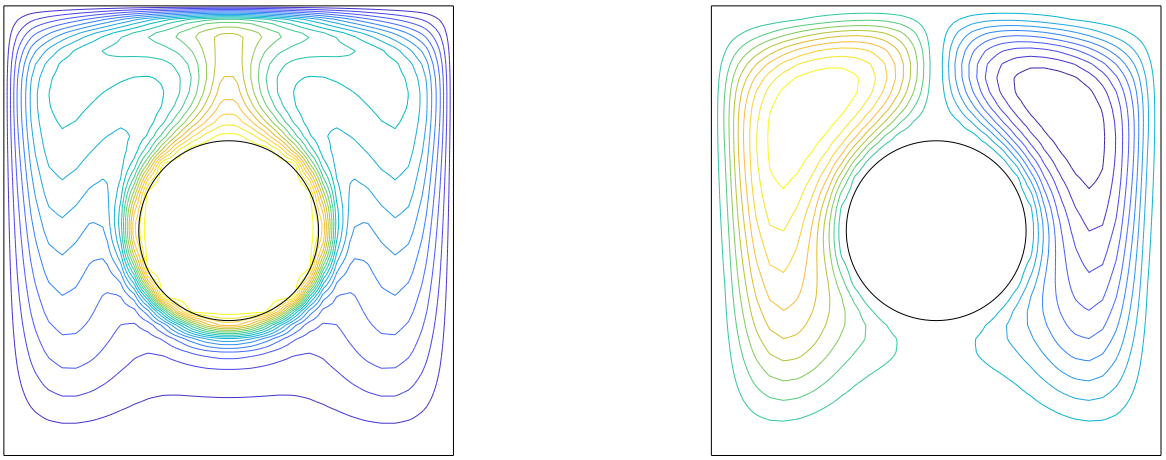


Figure 15: Natural convection, annulus domain: Distributions of the temperature (left) and stream function (right) acquired from a coarse grid of 32×32. Each plot contains 21 iso-lines whose associated values vary linearly between the two extremes.