Compact approximation stencils based on integrated flat radial basis functions

N. Mai-Duy[*], T.T.V. Le, C.M.T. Tien, D. Ngo-Cong and T. Tran-Cong
Computational Engineering and Science Research Centre,
School of Mechanical and Electrical Engineering,
University of Southern Queensland, Toowoomba, QLD 4350, Australia

Submitted to *Engineering Analysis with Boundary Elements*, Feb/2016;
revised (1) Jul/2016; revised (2) Sep/2016; revised (3) Oct/2016

Abstract This paper presents improved ways of constructing compact integrated radial basis function (CIRBF) stencils, based on extended precision, definite integrals, higher-order IRBFs and minimum number of derivative equations, to enhance their performance over large values of the RBF width. The proposed approaches are numerically verified through second-order linear differential equations in one and two variables. Significant improvements in the matrix condition number, solution accuracy and convergence rate with grid refinement over the usual approaches are achieved.

Keywords: compact approximation, local approximation, integrated radial basis function, flat radial basis function

1 Introduction

Radial basis functions (RBFs) have become one of the main fields of research in numerical analysis. It is theoretically proved that RBF networks having one hidden layer are capable of universal approximation [22]. They can represent an arbitrary continuous function within an arbitrarily small error bound. The application of RBFs for the numerical solution of

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

ordinary/partial differential equations (ODEs/PDEs) has received a great deal of attention. In RBF methods, the field variables/their derivatives are represented by linear combinations of RBFs, while the differential equations can be discretised by means of point collocation [6,7,23,26,9,8,21], subregion collocation [20,16], weak form [30,15] or inverse form [19].

Several types of RBFs contain a free parameter. This class can exhibit an exponential rate of convergence with the number of RBFs and the RBF's width [11]. One of the most widely used RBFs is the multiquadric (MQ) function defined as

$$G_i(\mathbf{x}) = \sqrt{(\mathbf{x}-\mathbf{c}_i)^T(\mathbf{x}-\mathbf{c}_i)+a_i^2} \tag{1}$$

where $\mathbf{c}_i$ and a_i are the centre and width of the ith MQ, or

$$G_i(\mathbf{x}) = \sqrt{\epsilon_i(\mathbf{x}-\mathbf{c}_i)^T(\mathbf{x}-\mathbf{c}_i)+1} \tag{2}$$

where ϵ_i is the shape parameter. The MQ function becomes increasingly flat when $a_i \rightarrow \infty$ or $\epsilon_i \rightarrow 0$.

RBF approximations for the field variable and its derivatives can be constructed through the differentiation (DRBF) [6] or integration (IRBF) [13,24,10,7,25] process. The latter was developed with the aim of avoiding the reduction in convergence rate caused by differentiation. It was also found that integration constants provide effective mechanisms for the implementation of multiple boundary conditions [12] and compact approximations [27], and the enhancement of continuity order of the approximate solution across subdomain interfaces [14]. Numerical experiments indicated that IRBFs converge faster, but produce the interpolation matrix with larger condition number than those by DRBFs.

When all RBFs are employed for the approximation at a point, the RBF method is regarded as a global method. It is easy to implement global RBF methods since no mesh (i.e. no connection between nodes) is involved. A highly accurate solution is typically obtained. On the other hand, the system matrix is fully populated and as a result, only a relatively low number of nodes can be employed in practice. Global approximations can work with small values of a_i only, typically the minimum distance between the ith RBF and its neighbours.

When only a few RBFs are activated for the approximation at a point (local approximation), there is a significant improvement in the matrix condition number but the solution accuracy is significantly reduced. The latter can be overcome by using compact approximations, where the approximation involves nodal values of not only the field variable but also its derivatives [28,29,31,17,27]. With compact RBF approximations, high levels of the solution accuracy and sparseness of the system matrix can be achieved together. They are capable of providing a very efficient solution to a differential problem. In contrast to global RBF methods, larger values of a_i can be employed here. It was shown in [3,2] that the RBF approximation is more accurate when a_i is increased (or ϵ_i is reduced) and the most accurate approximation occurs before a_i approaches infinity (or $\epsilon_i \to 0$). Furthermore, in the limit of $\epsilon_i \to 0$, the RBF approximation for a set of centres in one dimension reduces to the Largrange interpolating polynomial on that set of nodes [1]. Numerical experiments indicated that the interpolation matrices for local RBF and compact RBF stencils at large values of the RBF width are ill-conditioned and special treatments are needed. Effective treatments for compact RBF Hermite interpolation schemes (differentiated) were reported in, e.g. [31], where the Contour-Pade algorithm is employed. This work presents several simple but effective approaches to extend the working range of a_i for compact integrated RBF approximations.

The paper is organised as follows. A brief overview of CIRBF stencils is given in Section 2. In section 3, some numerical investigations are conducted to identify numerical issues due to the use of large values of a_i. In section 4, improved constructions for CIRBF stencils to extend the working range of a_i are presented and then numerically verified in analytic tests. Section 5 gives some concluding remarks.

2 Compact local integrated RBF stencils

Consider a 3-point stencil $[x_1, x_2, x_3]$. On the stencil, the second derivative of the dependent variable u is decomposed into

$$\frac{d^2u(x)}{dx^2} = \sum_{i=1}^{3} w_i G_i(x) \tag{3}$$

where $\{G_i(x)\}_{i=1}^3$ is the set of RBFs and $\{w_i\}_{i=1}^3$ the set of weights to be found. In one dimension, the multiquadric (MQ) function takes the form $G_i(x) = \sqrt{(x-c_i)^2 + a_i^2}$. We choose the width according to $a_i = \beta d_i$, where β is a scalar and d_i is the smallest distance between c_i and its neighbours.

Its first derivative and function are then obtained through integration

$$\frac{du(x)}{dx} = \sum_{i=1}^{3} w_i H_i(x) + C_1 \tag{4}$$

$$u(x) = \sum_{i=1}^{3} w_i \overline{H}_i(x) + C_1 x + C_2 \tag{5}$$

where $H_i(x) = \int G_i(x)dx$ and $\overline{H}_i(x) = \int H_i(x)dx$ are integrated basis functions and C_1 and C_2 the constants of integration.

For compact approximations, nodal values of the derivative (or the differential equation) at the side nodes of the stencil are also incorporated in the process of converting the RBF space into the physical space. Assuming that the differential equation takes the form $d^2u(x)/dx^2 = f(x)$ ($f(x)$ is a prescribed function), the mapping can be constructed as

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \frac{d^2u_1}{dx^2} \\ \frac{d^2u_3}{dx^2} \end{pmatrix} = \underbrace{\begin{bmatrix} \overline{H}_1(x_1), & \overline{H}_2(x_1), & \overline{H}_3(x_1), & x_1, & 1 \\ \overline{H}_1(x_2), & \overline{H}_2(x_2), & \overline{H}_3(x_2), & x_2, & 1 \\ \overline{H}_1(x_3), & \overline{H}_2(x_3), & \overline{H}_3(x_3), & x_3, & 1 \\ G_1(x_1), & G_2(x_1), & G_3(x_1), & 0, & 0 \\ G_1(x_3), & G_2(x_3), & G_3(x_3), & 0, & 0 \end{bmatrix}}_{\mathcal{C}} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ C_1 \\ C_2 \end{pmatrix} \tag{6}$$

where $\mathcal{C}$ is a 5×5 matrix that will hereafter be called the conversion matrix. Solving (6) leads to

$$\begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ C_1 \\ C_2 \end{pmatrix} = \mathcal{C}^{-1} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \frac{d^2u_1}{dx^2} \\ \frac{d^2u_3}{dx^2} \end{pmatrix} \tag{7}$$

The second derivative of function u at the middle node is thus computed as

$$\frac{d^2u_2}{dx^2} = [G_1(x_2), G_2(x_2), G_3(x_2), 0, 0]\,\mathcal{C}^{-1}\left(u_1, u_2, u_3, \frac{d^2u_1}{dx^2}, \frac{d^2u_3}{dx^2}\right)^T \tag{8}$$

or

$$\frac{d^2u_2}{dx^2} = \eta_1 u_1 + \eta_2 u_2 + \eta_3 u_3 + \eta_4 \frac{d^2u_1}{dx^2} + \eta_5 \frac{d^2u_3}{dx^2} \tag{9}$$

where $d^2u_1/dx^2 = f(x_1)$, $d^2u_3/dx^2 = f(x_3)$ and $\{\eta_i\}_{i=1}^5$ are known values. In the case of Dirichlet boundary conditions and the domain represented by a set of N nodes, the collocation of the differential equation at the interior nodes results in the following system

$$\mathcal{A}\overrightarrow{u} = \overrightarrow{b} \tag{10}$$

where $\mathcal{A}$ is the system matrix of dimensions $(N-2)\times(N-2)$, $\overrightarrow{u}$ the vector consisting of values of u at interior nodes and $\overrightarrow{b}$ the vector formed by the RHS of the differential equation and the boundary conditions. Like the central finite-difference method, the structure of $\mathcal{A}$ is tri-diagonal and the system can be efficiently solved for the nodal variable values.

3 Numerical investigation

We apply the CIRBF solution procedure to the following second-order ODE

$$\frac{d^2u}{dx^2} = -\exp(-5x)\left(9975\sin(100x) + 1000\cos(100x)\right), \quad 0 \leq x \leq 1 \tag{11}$$

subject to Dirichlet boundary conditions. The exact solution can be verified to be

$$u_e(x) = \sin(100x)\exp(-5x) \tag{12}$$

and is displayed in Figure 1.

Two 3-point stencils, IRBF and compact IRBF (CIRBF), are implemented. The two system matrices have the same structure (tri-diagonal), but as shown in Figure 2 the latter is much more accurate than the former. The RBF solution converges as $O(h^{4.79})$ for CIRBF and

$O(h^{1.95})$ for IRBF, indicating that the inclusion of nodal second derivative values significantly enhances the performance of local IRBF stencils.

Figure 3 shows variations in the condition number of the conversion and system matrices against the MQ width represented by β for a fixed grid size ($N_x = 1001$). For the system matrix $\mathcal{A}$, the condition number is rather low ($O(10^5)$) and it has similar values over a wide range of β. In contrast, the condition number of the conversion matrix $\mathcal{C}$ grows fast at a rate of 4.5 and the matrix becomes ill-conditioned at large values of β. Therefore, in using CIRBF stencils, attention should be paid to the handling of matrix C resulting from flat MQ functions.

4 Improved constructions for CIRBF stencils

Below are several treatments proposed to stably compute $\mathcal{C}$ at large values of β.

4.1 Approach 1: Extended precision

As shown in [3], by constructing the RBF interpolation with the Contour-Pade algorithm, the numerical solution still behaves stably when the basis functions become increasingly flat. The trade-off between accuracy and stability, which was reported widely in the RBF literature, is due to the use of finite (double) precision in computation. In this regard, the employment of higher precision is expected to improve the stability of the RBF solution, which was verified in [4,5]. Our program is written in Matlab and we employ function vpa (variable-precision arithmetic) to increase the number of significant decimal digits from 16 to 50 in constructing the conversion matrix $\mathcal{C}$ and computing its inverse. Higher computational cost is required. On the other hand, as shown in Figure 4, the IRBF solution is stable at large values of β and the optimal value of β is also clearly detected. It is noted that (i) by defining a stencil on the unit length, one may need to compute the inversion of the conversion matrix once and the result can be applied for any grid size to be employed; and (ii) in the present code, parts other than the computation of $\mathcal{C}$ are carried out using double precision, and numerical results indicate that the same level of accuracy is obtained as in the case of using extended precision for the entire computation.

4.2 Approach 2: Definite integral

We propose to compute the integrals in their definite form rather than indefinite in constructing the conversion matrix $\mathcal{C}$. The advantage of this approach is that the size of $\mathcal{C}$ is reduced from 5×5 to 3×3, and the numerical stability is thus expected to be improved.

The integrals in (4)-(5) can be rewritten as

$$\begin{aligned}
\frac{du(x)}{dx} - \frac{du_1}{dx} &= \sum_{i=1}^{3} \int_{x_1}^{x} w_i G_i(x) dx \\
&= \sum_{i=1}^{3} w_i \left[H_i(x) - H_i(x_1) \right] \qquad (13) \\
u(x) - u_1 - (x - x_1)\frac{du_1}{dx} &= \sum_{i=1}^{3} \int_{x_1}^{x} w_i \left[H_i(x) - H_i(x_1) \right] dx \\
&= \sum_{i=1}^{3} w_i \left[\overline{H}_i(x) - \overline{H}_i(x_1) - (x - x_1) H_i(x_1) \right] \qquad (14)
\end{aligned}$$

Letting $H_i^d(x) = H_i(x) - H_i(x_1)$, $\overline{H}_i^d = \overline{H}_i(x) - \overline{H}_i(x_1) - (x - x_1)H_i(x_1)$ and $u'(x) = du(x)/dx$, expressions (13) and (14) reduce to

$$u'(x) - u'_1 = \sum_{i=1}^{3} w_i H_i^d(x) \qquad (15)$$

$$u(x) - u_1 - (x - x_1)u'_1 = \sum_{i=1}^{3} w_i \overline{H}_i^d(x) \qquad (16)$$

Our objective now is to express the weights w_1, w_2 and w_3 in terms of u_1, u_2, u_3, u''_1 and u''_3. The conversion system is generated by collocating the function expression (16) at $x = x_2$

and $x = x_3$, and the second-derivative expression (3) at $x = x_1$

$$\begin{pmatrix} u_2 - u_1 - (x_2 - x_1)u'_1 \\ u_3 - u_1 - (x_3 - x_1)u'_1 \\ u''_1 \end{pmatrix} = \underbrace{\begin{bmatrix} \overline{H}^d_1(x_1), & \overline{H}^d_2(x_1), & \overline{H}^d_3(x_1) \\ \overline{H}^d_1(x_2), & \overline{H}^d_2(x_2), & \overline{H}^d_3(x_2) \\ G_1(x_1), & G_2(x_1), & G_3(x_1) \end{bmatrix}}_{\mathcal{C}} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} \tag{17}$$

Solving this system for the weights yields

$$\begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \mathcal{C}^{-1} \begin{pmatrix} u_2 - u_1 - (x_2 - x_1)u'_1 \\ u_3 - u_1 - (x_3 - x_1)u'_1 \\ u''_1 \end{pmatrix} \tag{18}$$

A next step is to incorporate u''_3 into the vector on the RHS of (18). We first collocate the second-derivative expression (3) at $x = x_3$

$$u''_3 = [G_1(x_3), G_2(x_3), G_3(x_3)]\, \mathcal{C}^{-1} \begin{pmatrix} u_2 - u_1 - (x_2 - x_1)u'_1 \\ u_3 - u_1 - (x_3 - x_1)u'_1 \\ u''_1 \end{pmatrix} \tag{19}$$

and then solving this equation for u'_1. Making substitution into the RHS of (18), the mapping of the RBF space into the physical space takes the form

$$\begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \mathcal{C}^{-1}\mathcal{T} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \frac{d^2u_1}{dx^2} \\ \frac{d^2u_3}{dx^2} \end{pmatrix} \tag{20}$$

where $\mathcal{C}$ is of dimension 3×3 and $\mathcal{T}$ is of 3×5, which is constructed using results from solving equation (19).

Figure 5 concerning the conversion matrix $\mathcal{C}$ shows a significant improvement in the condition number of the present definite-integral approach over the usual indefinite-integral approach. The former grows as $O(\beta^{3.88})$ only while the rate of the latter is much higher, up to 6.32.

Figure 6 indicates that the present approach makes the solution accuracy significantly less fluctuating over large values of β.

4.3 Approach 3: Higher-order IRBF approximations

The MQ function $G_i(x)$ is now integrated 4 times (IRBF4) instead of twice (IRBF2). Let $\overline{H}_i(x) = \int H_i(x)dx$, $\widehat{H}_i(x) = \int \overline{H}_i(x)dx$ and $\widetilde{H}_i(x) = \int \widehat{H}_i(x)dx$. We employ the integrated basis function $\overline{H}_i(x)$ instead of $G_i(x)$ to approximate the second-order derivative

$$\frac{d^2u}{dx^2} = \sum_{i=1}^{3} w_i \overline{H}_i(x) \tag{21}$$

$$\frac{du}{dx} = \sum_{i=1}^{3} w_i \widehat{H}_i(x) + C_1 \tag{22}$$

$$u = \sum_{i=1}^{3} w_i \widetilde{H}_i(x) + C_1 x + C_2 \tag{23}$$

where $\widehat{H}_i(x) = \int \overline{H}_i(x)dx$ and $\widetilde{H}_i(x) = \int \widehat{H}_i(x)dx$.

It was reported in [24] that the matrix condition number of IRBF4 is higher than that of IRBF2. However, with only three RBFs involved, the trend is reversed. As RBFs are integrated, the corresponding interpolation matrix has a lower condition number, particularly over a large range of β (Figure 7). When second-derivative values are added, as shown in Figure 8, the observation is similar. CIRBF4 is more stable than CIRBF2. This interesting property of higher-order IRBFs with 3 centres will be utilised here to construct compact IRBF stencils.

The conversion system in this approach is formed as

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \frac{d^2u_1}{dx^2} \\ \frac{d^2u_3}{dx^2} \end{pmatrix} = \underbrace{\begin{bmatrix} \widetilde{H}_1(x_1), & \widetilde{H}_2(x_1), & \widetilde{H}_3(x_1), & x_1, & 1 \\ \widetilde{H}_1(x_2), & \widetilde{H}_2(x_2), & \widetilde{H}_3(x_2), & x_2, & 1 \\ \widetilde{H}_1(x_3), & \widetilde{H}_2(x_3), & \widetilde{H}_3(x_3), & x_3, & 1 \\ \overline{H}_1(x_1), & \overline{H}_2(x_1), & \overline{H}_3(x_1), & 0, & 0 \\ \overline{H}_1(x_3), & \overline{H}_2(x_3), & \overline{H}_3(x_3), & 0, & 0 \end{bmatrix}}_{\mathcal{C}} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ C_1 \\ C_2 \end{pmatrix} \tag{24}$$

It can be seen from Figure 9 that, for a given β, a much more stable solution is obtained with the present approach as the grid size is reduced. At a very small grid size, the present approach is much more accurate and more stable over a large value range of β than the usual approach (Figure 10).

These improved 3-point CIRBF stencils can be extended to construct 5-point stencils for solving problems in two dimensions. The implementation process is exactly the same as that presented in [18]. For elliptic PDEs, the algebraic system, where each row has 5 non-zero entries, can be solved iteratively using a Picard scheme. For parabolic PDEs, systems of tridiagonal equations can be formed and solved efficiently with Thomas algorithm. It requires that the problem domain is represented by a Cartesian grid (not by a set of scattered points). Thus, for non-rectangular domains, the discretisation is still based on Cartesian grid but with non-uniformly-spaced stencils.

Consider Poisson equation (32) defined on a non-rectangular domain (Figure 11) and subjected to Dirichlet boundary conditions. The exact solution is chosen to be $u^{(e)}(x,y) = \exp(-(x-0.25)^2-(y-0.5)^2)\sin(\pi x)\cos(2\pi y)$. The problem domain is embedded in Cartesian grid, where the interior nodes are grid nodes inside the domain and the boundary nodes are the intersections of the grid lines and the boundary. Figure 11 also shows that as the RBF width increases, the present construction of CIRBF approximations results in a much more accurate and stable solution than the usual approach.

4.4 Approach 4: Separate construction in each direction and minimum number of derivative equations

This approach is developed for CIRBF stencils based on two-dimensional approximations. In this section, new compact 9-point IRBF stencils are constructed. Unlike our previous work [17], the conversion process of the RBF space into the physical space is now conducted independently in each direction, where the size of the conversion matrix is reduced about half. Below is a schematic diagram 9-point stencil associated with node (i, j)

$$\begin{bmatrix} \mathbf{x}_3 & \mathbf{x}_6 & \mathbf{x}_9 \\ \mathbf{x}_2 & \mathbf{x}_5 & \mathbf{x}_8 \\ \mathbf{x}_1 & \mathbf{x}_4 & \mathbf{x}_7 \end{bmatrix}$$

The nodes are locally numbered from left to right and from bottom to top, where node (i, j) is located at the centre (i.e. $(i, j) \equiv$ node 5). In the x direction, the process of approximating the field variable and its derivatives starts with

$$\frac{\partial^2 u(x,y)}{\partial x^2} = \sum_{i=1}^{9} w_i G_i(x,y), \tag{25}$$

where $G_i(x,y) = \sqrt{[x-(c_i)_x]^2 + [y-(c_i)_y]^2 + a_i^2}$. Integrating (25) once and twice yields

$$\begin{aligned} \frac{\partial u}{\partial x}(x,y) &= \sum_{i=1}^{9} w_i H_i(x,y) + C_1(y) && (26) \\ u(x,y) &= \sum_{i=1}^{9} w_i \overline{H}_i(x,y) + xC_1(y) + C_2(y) && (27) \end{aligned}$$

where C_1 and C_2 are functions of y. It was shown in [17] that the most accurate approximation is achieved when the derivative values incorporated in the conversion system are taken at nodes 2, 4, 6 and 8. We follow this strategy in the present construction.

The conversion system is formed as

$$\begin{pmatrix} \overrightarrow{u} \\ \overrightarrow{\frac{\partial^2 u}{\partial x^2}} \end{pmatrix} = \underbrace{\begin{bmatrix} \overline{\mathcal{H}} \\ \mathcal{K} \end{bmatrix}}_{\mathcal{C}^{[x]}} \begin{pmatrix} \overrightarrow{w} \\ \overrightarrow{C} \end{pmatrix} \tag{28}$$

where $\mathcal{C}^{[x]}$ is the conversion matrix;

$$
\begin{aligned}
\overrightarrow{u} &= (u_1, u_2, \cdots, u_9)^T \\
\overrightarrow{w} &= (w_1, w_2, \cdots, w_9)^T \\
\overrightarrow{C} &= (C_1(y_1), C_1(y_2), C_1(y_3), C_2(y_1), C_2(y_2), C_2(y_3))^T \\
\overline{\mathcal{H}} &= \begin{bmatrix}
\overline{H}_1(\mathbf{x}_1), & \cdots, & \overline{H}_9(\mathbf{x}_1), & x_1, & 0, & 0, & 1, & 0, & 0 \\
\overline{H}_1(\mathbf{x}_2), & \cdots, & \overline{H}_9(\mathbf{x}_2), & 0, & x_2, & 0, & 0, & 1, & 0 \\
\overline{H}_1(\mathbf{x}_3), & \cdots, & \overline{H}_9^{[x]}(\mathbf{x}_3), & 0, & 0, & x_3, & 0, & 0, & 1 \\
\overline{H}_1(\mathbf{x}_4), & \cdots, & \overline{H}_9(\mathbf{x}_4), & x_4, & 0, & 0, & 1, & 0, & 0 \\
\overline{H}_1(\mathbf{x}_5), & \cdots, & \overline{H}_9(\mathbf{x}_5), & 0, & x_5, & 0, & 0, & 1, & 0 \\
\overline{H}_1(\mathbf{x}_6), & \cdots, & \overline{H}_9(\mathbf{x}_6), & 0, & 0, & x_6, & 0, & 0, & 1 \\
\overline{H}_1(\mathbf{x}_7), & \cdots, & \overline{H}_9(\mathbf{x}_7), & x_7, & 0, & 0, & 1, & 0, & 0 \\
\overline{H}_1(\mathbf{x}_8), & \cdots, & \overline{H}_9(\mathbf{x}_8), & 0, & x_8, & 0, & 0, & 1, & 0 \\
\overline{H}_1(\mathbf{x}_9), & \cdots, & \overline{H}_9(\mathbf{x}_9), & 0, & 0, & x_9, & 0, & 0, & 1
\end{bmatrix}
\end{aligned}
$$

and

$$\frac{\overrightarrow{\partial^2 u}}{\partial x^2} = \mathcal{K}\begin{pmatrix} \overrightarrow{w} \\ \overrightarrow{C} \end{pmatrix}$$

are derivative equations. We observe that using a larger number of derivative equations can lead to a more accurate approximation but also increase the condition number of $\mathcal{C}$. We investigate the following two typical cases:

1. Case 1: two derivative equations

$$
\begin{aligned}
\frac{\overrightarrow{\partial^2 u}}{\partial x^2} &= \left(\frac{\partial^2 u_2}{\partial x^2}, \frac{\partial^2 u_8}{\partial x^2}\right)^T \\
\mathcal{K} &= \begin{bmatrix}
G_1(\mathbf{x}_2), & \cdots, & G_9(\mathbf{x}_2), & 0, & \cdots, & 0 \\
G_1(\mathbf{x}_8), & \cdots, & G_9(\mathbf{x}_8), & 0, & \cdots, & 0
\end{bmatrix}
\end{aligned}
$$

2. Case 2: four derivative equations

$$\overrightarrow{\frac{\partial^2 u}{\partial x^2}} = \left(\frac{\partial^2 u_2}{\partial x^2}, \frac{\partial^2 u_4}{\partial x^2}, \frac{\partial^2 u_6}{\partial x^2}, \frac{\partial^2 u_8}{\partial x^2} \right)^T$$

$$\mathcal{K} = \begin{bmatrix} G_1(\mathbf{x}_2), & \cdots, & G_9(\mathbf{x}_2), & 0, & \cdots, & 0 \\ G_1(\mathbf{x}_4), & \cdots, & G_9(\mathbf{x}_4), & 0, & \cdots, & 0 \\ G_1(\mathbf{x}_6), & \cdots, & G_9(\mathbf{x}_6), & 0, & \cdots, & 0 \\ G_1(\mathbf{x}_8), & \cdots, & G_9(\mathbf{x}_8), & 0, & \cdots, & 0 \end{bmatrix}$$

One can compute $\partial^2 u/\partial x^2$ at node 5 as

$$\frac{\partial^2 u_5}{\partial x^2} = [G_1(\mathbf{x}_5), \cdots, G_9(\mathbf{x}_5), 0, \cdots, 0] \left(\mathcal{C}^{[x]}\right)^{-1} \left(\overrightarrow{u}, \overrightarrow{\frac{\partial^2 u}{\partial x^2}} \right)^T \tag{29}$$

The approximation in the y direction can be derived in a similar fashion

$$\frac{\partial^2 u_5}{\partial y^2} = [G_1(\mathbf{x}_5), \cdots, G_9(\mathbf{x}_5), 0, \cdots, 0] \left(\mathcal{C}^{[y]}\right)^{-1} \left(\overrightarrow{u}, \overrightarrow{\frac{\partial^2 u}{\partial y^2}} \right)^T \tag{30}$$

where

$$\overrightarrow{\frac{\partial^2 u}{\partial y^2}} = \left(\frac{\partial^2 u_4}{\partial y^2}, \frac{\partial^2 u_6}{\partial y^2} \right)^T$$

for the case of two derivative equations, and

$$\overrightarrow{\frac{\partial^2 u}{\partial y^2}} = \left(\frac{\partial^2 u_2}{\partial y^2}, \frac{\partial^2 u_4}{\partial y^2}, \frac{\partial^2 u_6}{\partial y^2}, \frac{\partial^2 u_8}{\partial y^2} \right)^T$$

for the case of four derivative equations.

At each interior node, there are 3 unknowns, namely u, $\partial^2 u/\partial x^2$ and $\partial^2 u/\partial y^2$, and one can also establish 3 independent algebraic equations derived from collocating the differential equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial x^2} = f(x, y) \tag{31}$$

and applying the CIRBF equations of second derivative in the x (i.e. (29)) and y (i.e. (30)) direction at the interior node.

We employ an iterative procedure to reduce the number of unknowns from 3 to 1. Substituting (29) and (30) into (31) and then collocating the obtained equation at node 5 leads to the following algebraic equation, e.g. for the case of two derivative equations,

$$\sum_{i=1}^{9} \mu_i u_i^k = f_5 + \sum_{i=(2,8)} \gamma_i \frac{\partial^2 u_i^{k-1}}{\partial x^2} + \sum_{i=(4,6)} \lambda_i \frac{\partial^2 u_i^{k-1}}{\partial y^2} \tag{32}$$

where the superscript k is used to denote the present iteration. The solution procedure is as follows.

1. Guess a distribution of the field variable $u_{i,j}$

2. Compute second derivatives at grid nodes using equations (29) and (30).

3. Collocate (32) at the interior grid nodes, impose the prescribed boundary conditions and solve the obtained system of equations. Note that the system matrix is sparse as each row contains only 9 non-zero entries.

4. Check the convergence of the iterative procedure

$$CM = \frac{\sqrt{\sum \left(u_{i,j}^k - u_{i,j}^{k-1}\right)^2}}{\sqrt{\sum \left(u_{i,j}^k\right)^2}} < 10^{-12}$$

5. If not, relax the solution and then go back to step 2

$$u_{i,j}^k = \alpha u_{i,j}^k + (1-\alpha) u_{i,j}^{k-1}$$

   where α is the relaxation factor $(0 < \alpha \leq 1)$

6. If yes, stop and output the solution

Consider Poisson equation (32) defined on $0 \leq x, y \leq 1$ and subjected to Dirichlet boundary conditions. The exact solution is chosen to be $u^{(e)}(x,y) = \exp(-(x-0.25)^2 - (y-0.5)^2)\sin(\pi x)\cos(2\pi y)$. Using a grid of 37×37 and $\beta = 35$, the iterative scheme reaches $CM = 10^{-12}$ with 314 iterations for $\alpha = 0.1$, 95 for $\alpha = 0.3$, 51 for $\alpha = 0.5$, 31 for $\alpha = 0.7$ and 14 for $\alpha = 1$. The larger the value of α the faster the convergence will be. It is noted that the present iterative scheme can work with the largest value of α. In (32), the values of second derivative at the side nodes of the stencil are imposed. Alternatively, one can impose the differential equation by making the following replacements

$$\left(\frac{\partial^2 u}{\partial x^2}\right)^{k-1}_{i-1,j} \to f_{i-1,j} - \left(\frac{\partial^2 u}{\partial y^2}\right)^{k-1}_{i-1,j}, \quad \left(\frac{\partial^2 u}{\partial x^2}\right)^{k-1}_{i+1,j} \to f_{i+1,j} - \left(\frac{\partial^2 u}{\partial y^2}\right)^{k-1}_{i+1,j}$$
$$\left(\frac{\partial^2 u}{\partial y^2}\right)^{k-1}_{i,j-1} \to f_{i,j-1} - \left(\frac{\partial^2 u}{\partial x^2}\right)^{k-1}_{i,j-1}, \quad \left(\frac{\partial^2 u}{\partial y^2}\right)^{k-1}_{i,j+1} \to f_{i,j+1} - \left(\frac{\partial^2 u}{\partial x^2}\right)^{k-1}_{i,j+1}$$

Numerical results indicate that the imposition of PDE rather than second derivatives results in a much faster convergence of the iterative scheme. For example, for $\alpha = 0.5$, the number of iteration is reduced from 51 to 34 as shown in Figure 12.

Figure 13 shows the effect of the MQ width represented by β on the condition number of matrix $\mathcal{C}$ and the solution accuracy for a given grid size. Reducing the number of derivative equations leads to a much stable calculation over large values of β. At $\beta = 38$, the condition number of matrix $\mathcal{C}$ by using two derivative equations is about 6 orders of magnitude lower than the case of 4 derivative equations. The former produces highly accurate solutions at large β. The optimal value of β is clearly detected; the corresponding error Ne is 1.02×10^{-08}. When β is small (i.e. $\beta < 10$), it can be seen that the matrix $\mathcal{C}$ is well conditioned and using more derivative equations results in a better accuracy. Note that at large values of β, better accuracy is also obtained with the case of more derivative equations if extended precision is employed.

Figure 14 shows the effect of the grid size on the matrix condition number and the solution accuracy at a large value of β. By constructing CIRBF approximations on a stencil defined on $[0,1] \times [0,1]$, the conversion matrix $\mathcal{C}$ is independent of the grid size. It can be seen that the condition numbers of $\mathcal{C}$ by the use of 2 and 4 derivative equations differ by 6 orders of magnitude for all grid sizes. However, the matrix A is well-conditioned for the two cases, where their condition numbers all grow slow at the rate $O(h^{-2.00})$. The solution converges as $O(h^{5.12})$ for the case of two derivative equations and only $O(h^{2.31})$ for the case of 4 equations. At small values of h, the solution by the former is highly accurate with its error Ne being reduced to $O(10^{-9})$. The solution accuracy for the case of 4 derivative equations is significantly improved when extended precsion is used; it produces greater accuracy than the case of 2 derivative equations.

5 Concluding remarks

This paper shows that by taking appropriate ways of constructing the approximations in the process of converting the RBF space into the physical space, compact local integrated RBF stencils based on one- and two-dimensional approximations are capable of producing a stable solution over large values of the RBF width. Four approaches, based on extended precision, definite integrals, higher IRBF approximations and minimum number of derivative equations, are presented and numerically verified. For differential problems with smooth solutions, much more stable calculations and highly accurate results over the usual approaches are obtained. Each approach has its own strengths and weaknesses. Better accuracy and stable solution are achieved with extended precision at the expense of higher computational costs and the need for specialized computational tools such as function vpa in Matlab. However, in the case of uniform grids, by defining stencils on the unit length (1D) and the unit square (2D), one may need to compute the inversion once and then store/apply for any grid to be employed. The other approaches, which are simple and easy to implement, are capable of making the working range of the RBF witdth much larger. Their computational costs are relatively low. This works further shows a great potential of compact RBF stencils in solving differential problems.

References

1. Driscoll TA, Fornberg B. Interpolation in the limit of increasingly flat radial basis functions. Computers & Mathematics with Applications 2002; 43(3-5):413-422.

2. Fornberg B, Larsson E, Flyer N. Stable computations with Gaussian radial basis functions. SIAM Journal Scientific Computing 2011; 33(2):869-892.

3. Fornberg B, Wright G. Stable computation of multiquadric interpolants for all values of the shape parameter. Computers & Mathematics with Applications 2004; 48(5-6): 853-867.

4. 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.

5. Huang C-S, Yen H-D, Cheng AH-D. On the increasingly flat radial basis function and optimal shape parameter for the solution of elliptic PDEs. Engineering Analysis with Boundary Elements 2010; 34(9):802-809.

6. 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.

7. Kansa EJ, Power H, Fasshauer GE, Ling L. A volumetric integral radial basis function method for time-dependent partial differential equations: I. Formulation. Engineering Analysis with Boundary Elements 2004; 28:1191-1206.

8. Li M, Chen W, Chen CS. The localized RBFs collocation methods for solving high dimensional PDEs. Engineering Analysis with Boundary Elements 2013; 37(10):1300-1304.

9. Li M, Jiang T, Hon YC. A meshless method based on RBFs method for nonhomogeneous backward heat conduction problem. Engineering Analysis with Boundary Elements 2010; 34(9):785-792.

10. Ling L, Trummer MR. Multiquadric collocation method with integral formulation for boundary layer problems. Computers & Mathematics with Applications 2004; 48(5-6):927-941.

11. Madych WR. Miscellaneous error bounds for multiquadric and related interpolators. Computers & Mathematics with Applications 1992; 24(12):121-138.

12. Mai-Duy N. Solving high order ordinary differential equations with radial basis function networks. International Journal for Numerical Methods in Engineering 2005; 62:824-852.

13. Mai-Duy N, Tran-Cong T. Numerical solution of differential equations using multiquadric radial basis function networks. Neural Networks 2001; 14(2):185-199.

14. 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.

15. Mai-Duy N, Tran-Cong T. An integrated-RBF technique based on Galerkin formulation for elliptic differential equations. Engineering Analysis with Boundary Elements 2009; 33(2):191-199.

16. Mai-Duy N, Tran-Cong T. A control volume technique based on integrated RBFNs for the convection-diffusion equation. Numerical Methods for Partial Differential Equations 2010; 26(2):426-447.

17. 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.

18. 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.

19. Mai-Duy N, Tran-Cong T, Tanner RI. A domain-type boundary-integral-equation method for two-dimensional biharmonic Dirichlet problem. Engineering Analysis with Boundary Elements 2006; 30(10):809-817.

20. Moroney TJ, Turner IW. A finite volume method based on radial basis functions for two-dimensional nonlinear diffusion equations. Applied Mathematical Modelling 2006; 30(10):1118-1133.

21. Ngo-Cong D, Mohammed FJ, Strunin DV, Skvortsov AT, Mai-Duy N, Tran-Cong T. Higher-order approximation of contaminant transport equation for turbulent channel flows based on centre manifolds and its numerical solution. Journal of Hydrology 2015; 525:87-101.

22. Park J, Sandberg IW. Universal approximation using radial basis function networks. Neural Computation 1991; 3:246-257.

23. Sarler B. A radial basis function collocation approach in computational fluid dynamics. CMES: Computer Modeling in Engineering & Sciences 2005; 7(2):185-194.

24. Sarra SA. Integrated multiquadric radial basis function approximation methods. Computers & Mathematics with Applications 2006; 51(8):1283-1296.

25. Shu C, Wu YL. Integrated radial basis functions-based differential quadrature method and its performance. International Journal for Numerical Methods in Fluids 2007; 53(6):969-984.

26. Stevens D, Power H, Cliffe KA. A solution to linear elasticity using locally supported RBF collocation in a generalised finite-difference mode. Engineering Analysis with Boundary Elements 2013; 37(1):32-41.

27. Tien CMT, Thai-Quang N, Mai-Duy N, Tran C-D, Tran-Cong T. A three-point coupled compact integrated RBF scheme for second-order differential problems; CMES: Computer Modeling in Engineering and Sciences 2015; 104(6):425-469.

28. 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.

29. Tolstykh AI, Shirobokov DA. Using radial basis functions in a “finite difference mode”. CMES: Computer Modeling in Engineering & Sciences 2005; 7(2):207-222.

30. Wendland H. Meshless Galerkin methods using radial basis functions. Mathematics of Computation 1999; 68:1521-1531.

31. 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.

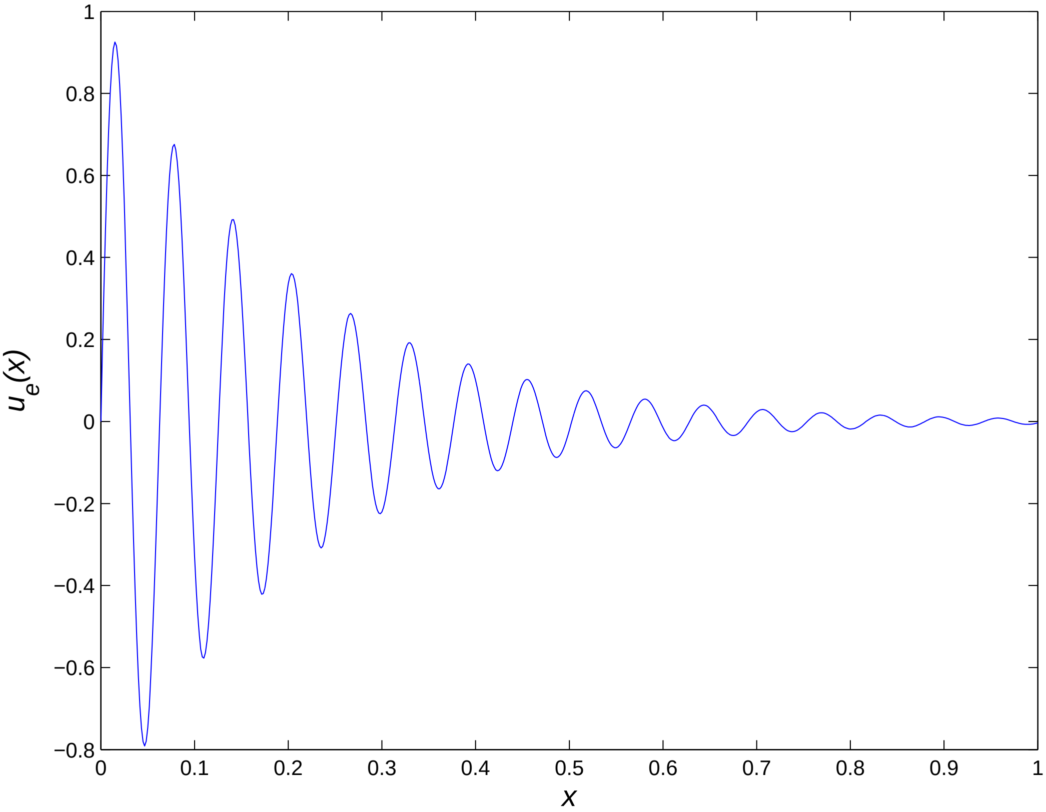

Figure 1: Second-order ODE, $u_e(x) = \sin(100x)\exp(-5x)$: Exact solution. The function is smooth and varies significantly over the domain. Such a variation requires a relatively large number of nodes for an accurate interpolation.

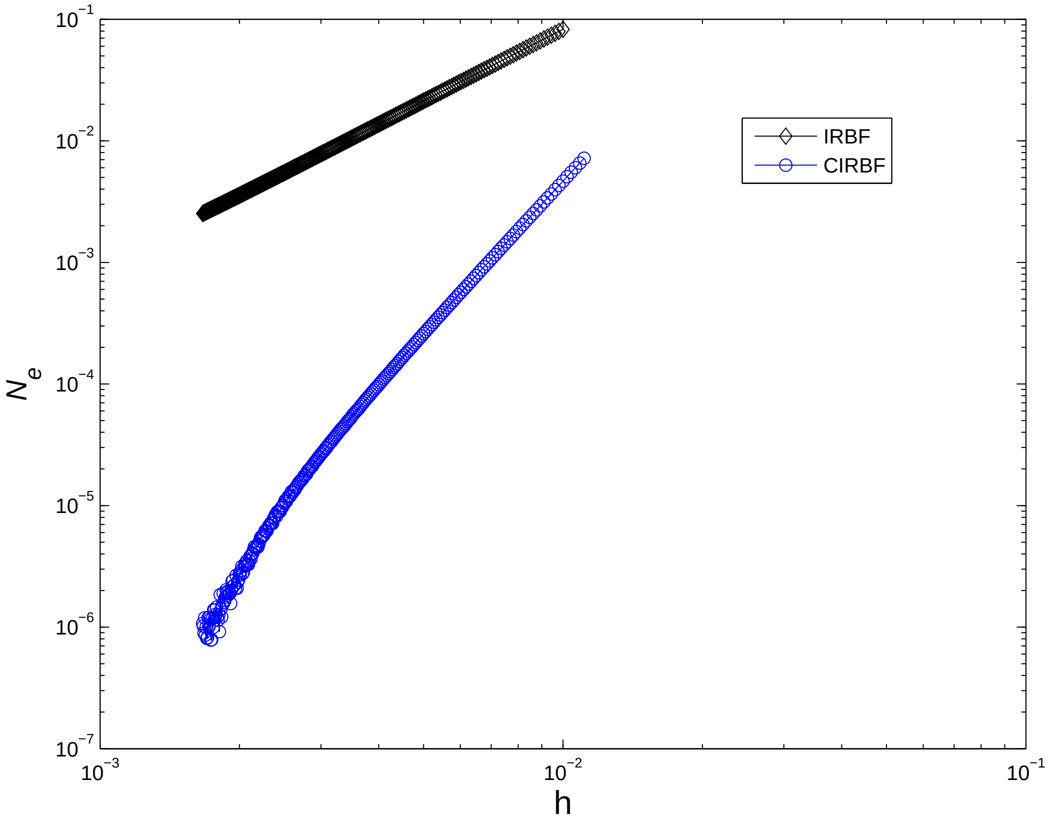

Figure 2: Second-order ODE, 3-point stencil, $0 \leq x \leq 1$, $91 \leq N_x \leq 601$, $\beta = 20$: Solution accuracy by IRBF and CIRBF. The solution converges as $O(h^{1.95})$ for IRBF and $O(h^{4.79})$ for CIRBF. Note that the 3-point stencil is constructed on a unit length.

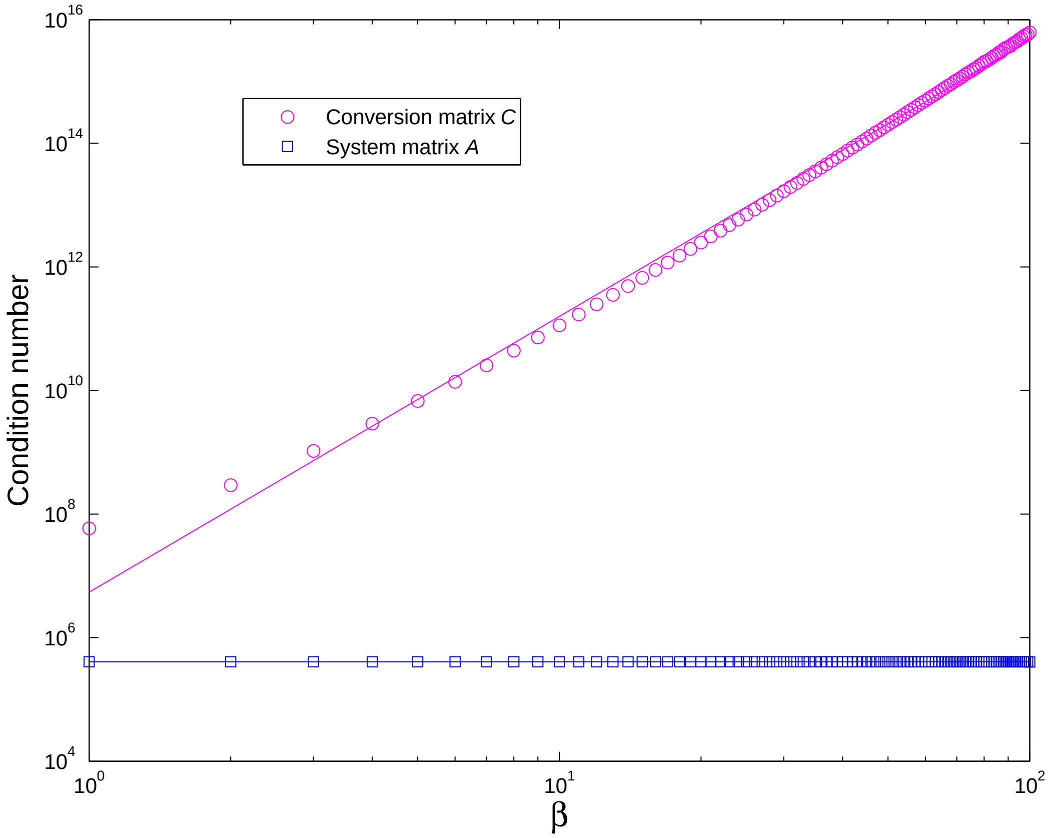

Figure 3: Second-order ODE, 3-point stencil, $0 \leq x \leq 1$, $N_x = 1001$: Condition numbers of the system matrix A and conversion matrix C as functions of β representing the RBF width. When β increases, the growth rate is about 4.46 for cond(C) and 0.00 for cond(A). Note that the 3-point stencil is constructed on a unit length.

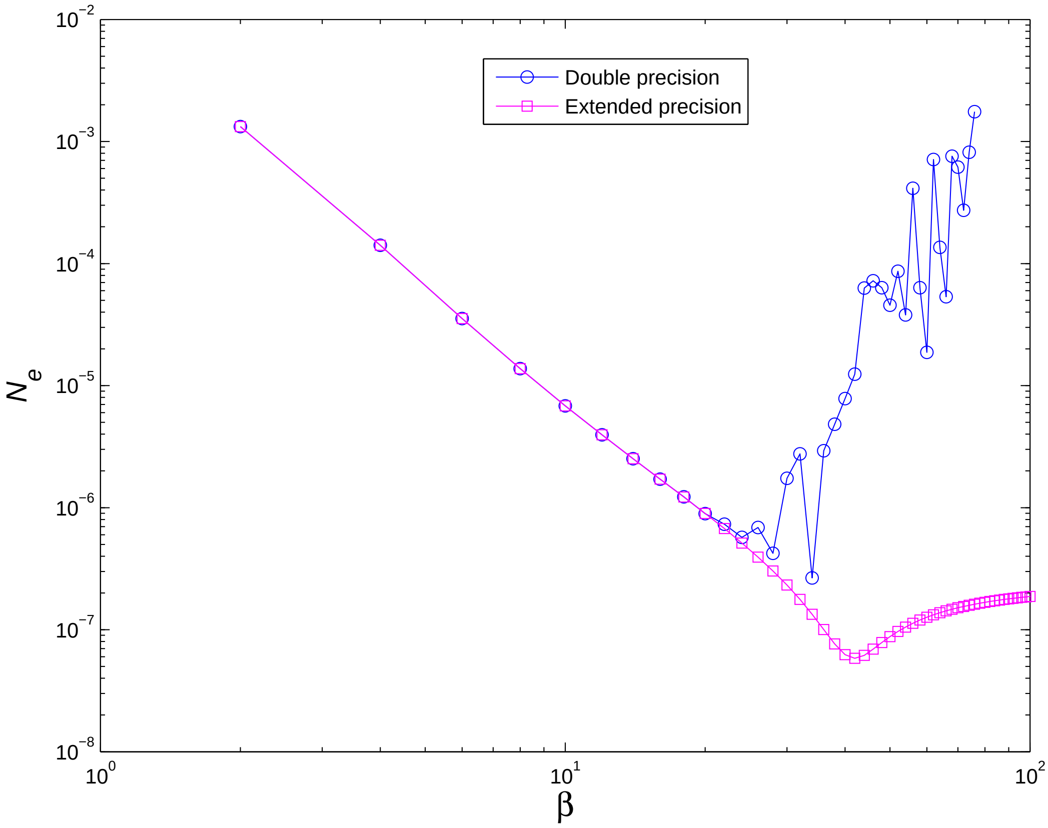

Figure 4: Second-order ODE, 3-point stencil, $0 \leq x \leq 1$, $N_x = 1201$: Solution accuracy by using double precision and extended precision (50 digits) in constructing and computing the conversion matrix.

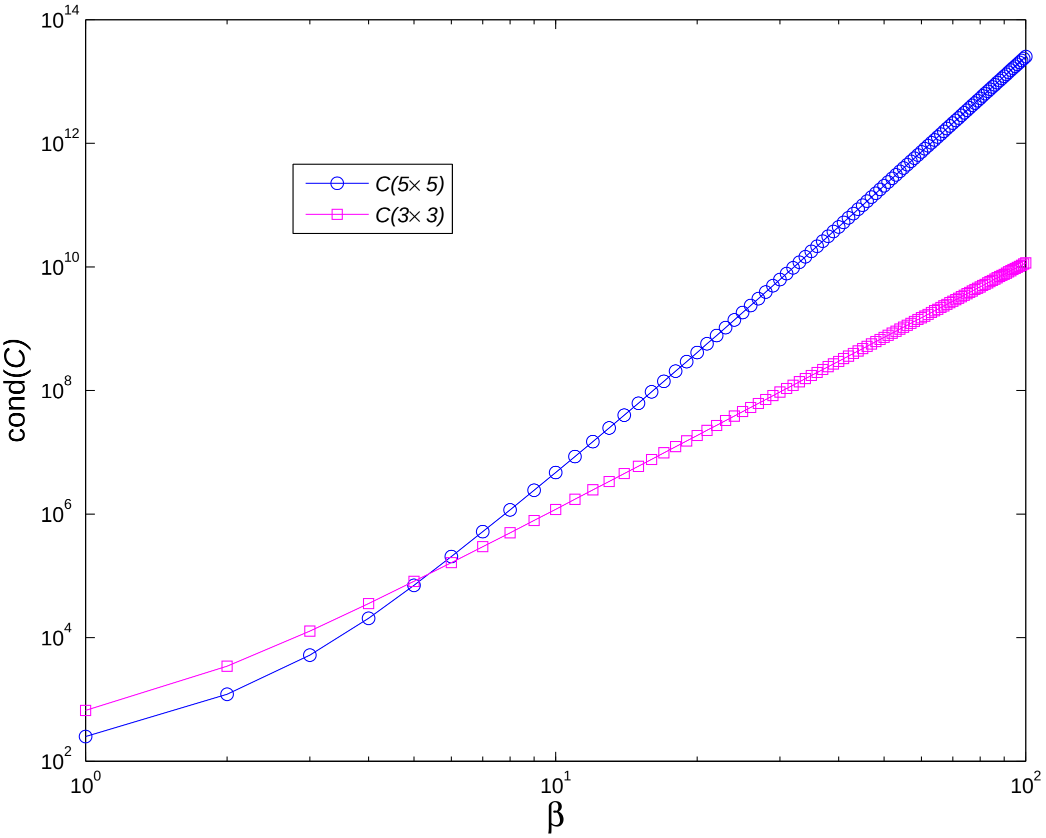

Figure 5: 3-point stencil, $N_x = 3$: Condition number of conversion matrix C computed through indefinite integrals, resulting in a matrix of 5×5 and through definite integral, resulting in a matrix of 3×3. The matrix condition number grows as $O(\beta^{6.32})$ for the former and $O(\beta^{3.88})$ for the latter.

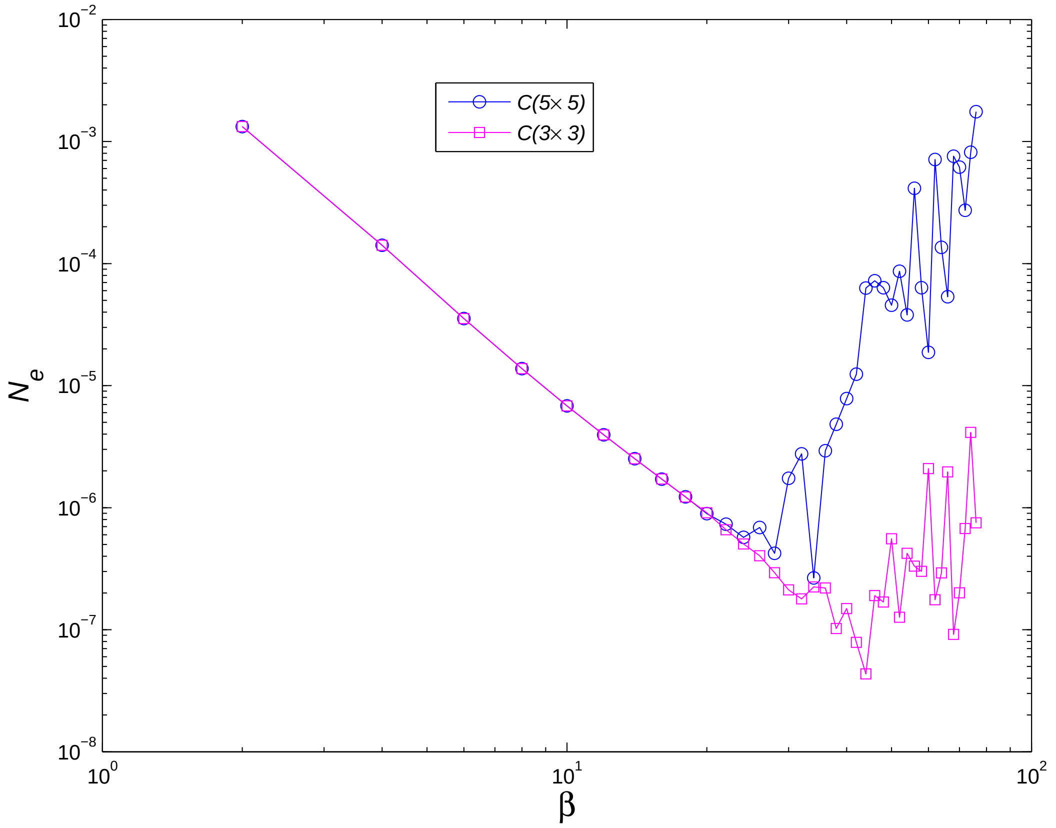

Figure 6: Second-order ODE, 3-point stencil, $0 \le x \le 1$, $N_x = 1201$: Solution accuracy by the two methods (indefinite and definite integrals) of constructing matrix C.

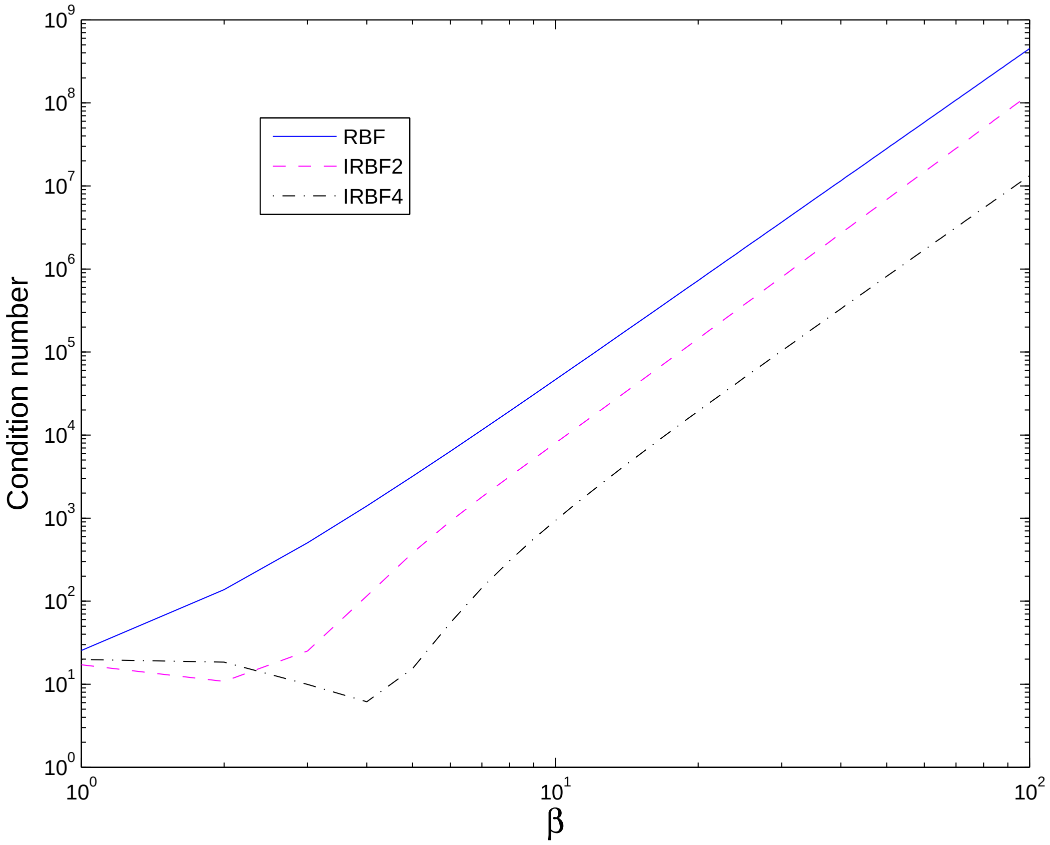

Figure 7: 3-point stencil: Condition numbers of the interpolation matrix generated by RBF, IRBF2 and IRBF4.

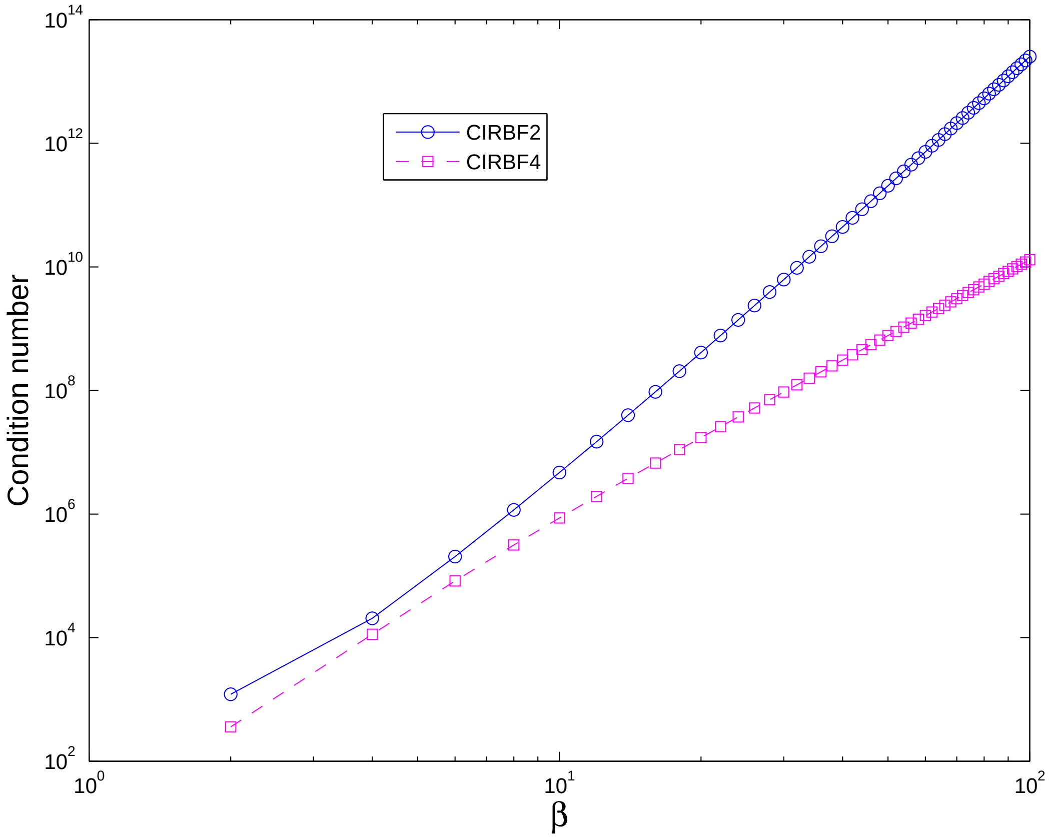

Figure 8: 3-point stencil, indefinite integral, $N_x = 3$: Condition numbers of the interpolation matrix generated by compact IRBF2 (indefinite integral) and compact IRBF4.

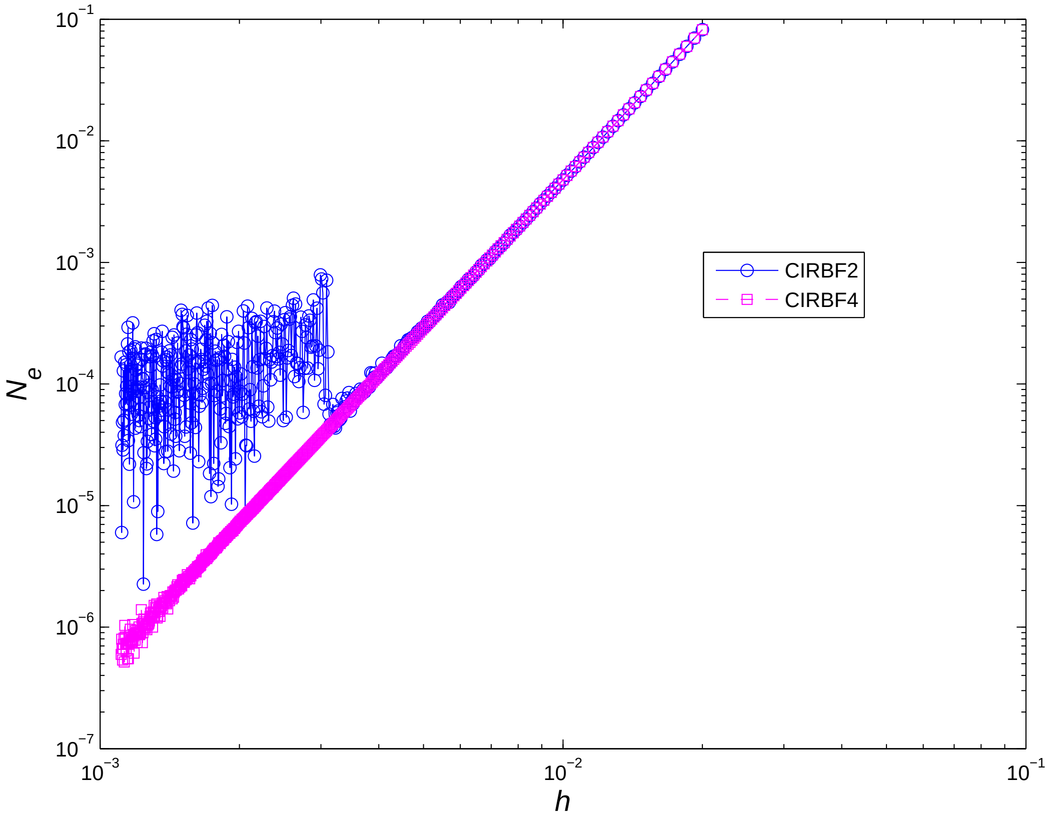

Figure 9: Second-order ODE, 3-point stencil, $0 \leq x \leq 1$, $\beta = 50$, $N_x = (51, 53, \cdots, 901)$: Solution accuracy against the grid size by CIRBF2 (indefinite integral) and CIRBF4. For the latter, the solution converges as $O(h^{4.05})$.

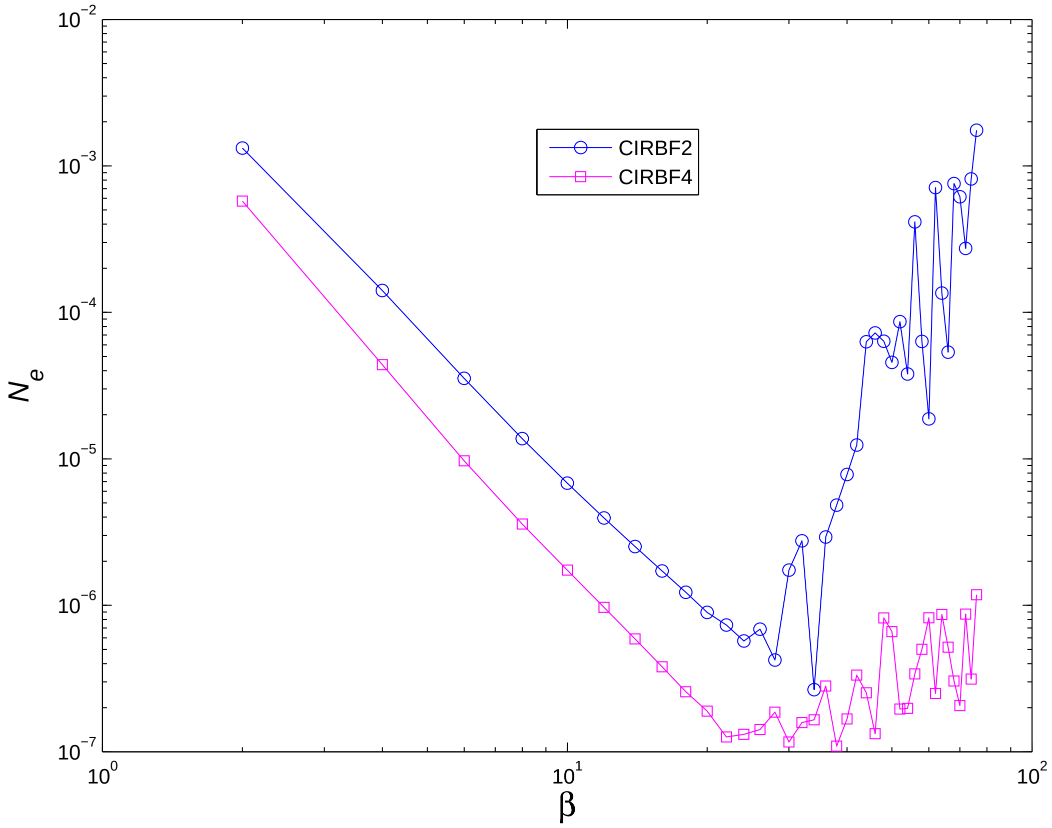

Figure 10: Second-order ODE, 3-point stencil, $0 \leq x \leq 1$, $N_x = 1201$: Solution accuracy by CIRBF2 (indefinite integral) and CIRBF4.

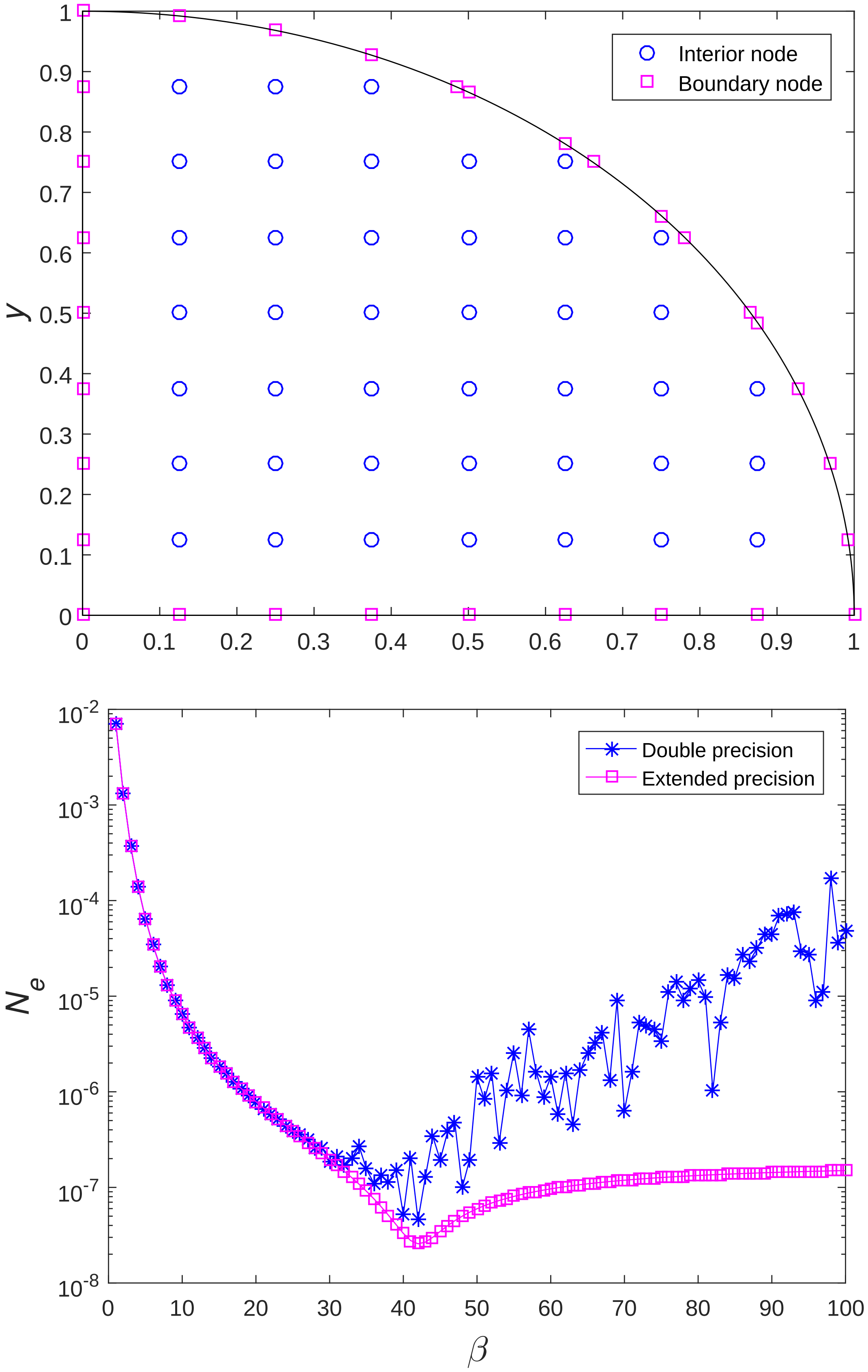

Figure 11: PDE, non-rectangular domain: Cartesian grid for non-rectangular domain, where the boundary nodes are the intersections of the grid lines and the boundary; and solution accuracy by using double precision and extended precision (50 digits) in constructing and computing the conversion matrix, where 81×81 grid lines are employed.

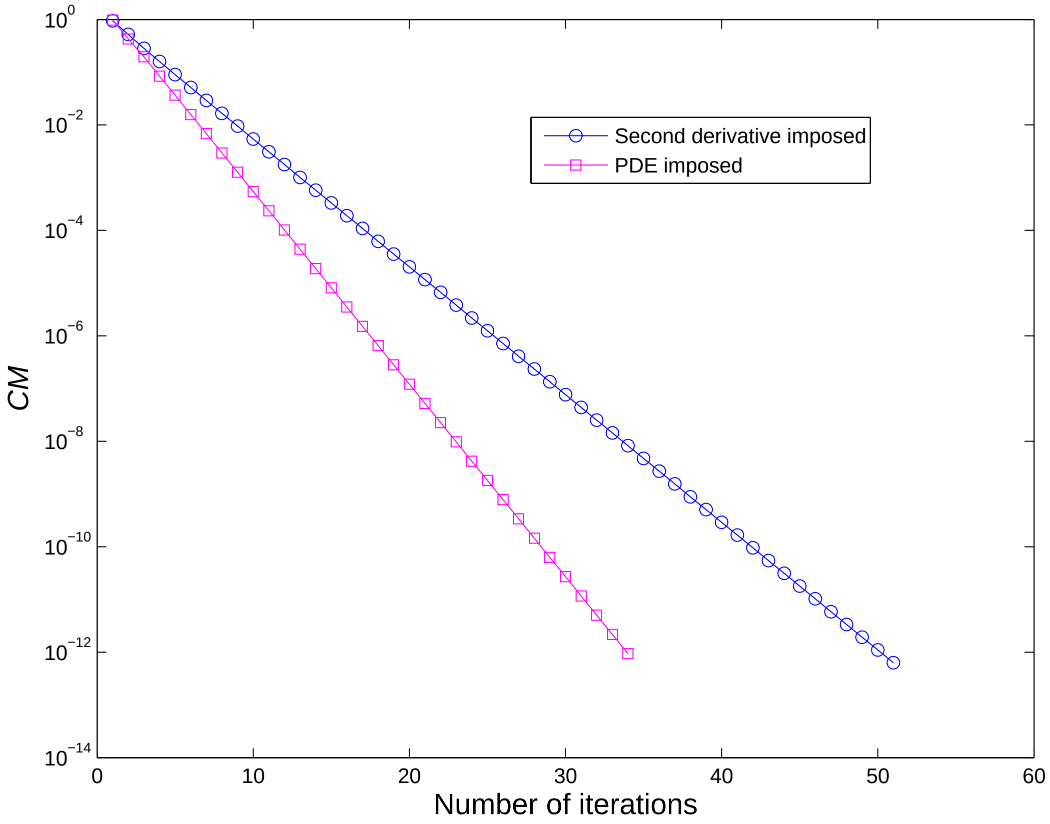

Figure 12: PDE, 37×37, $\beta = 35$, $\alpha = 0.5$, 2 derivative equations: Imposition of PDE converges faster than imposition of second derivatives.

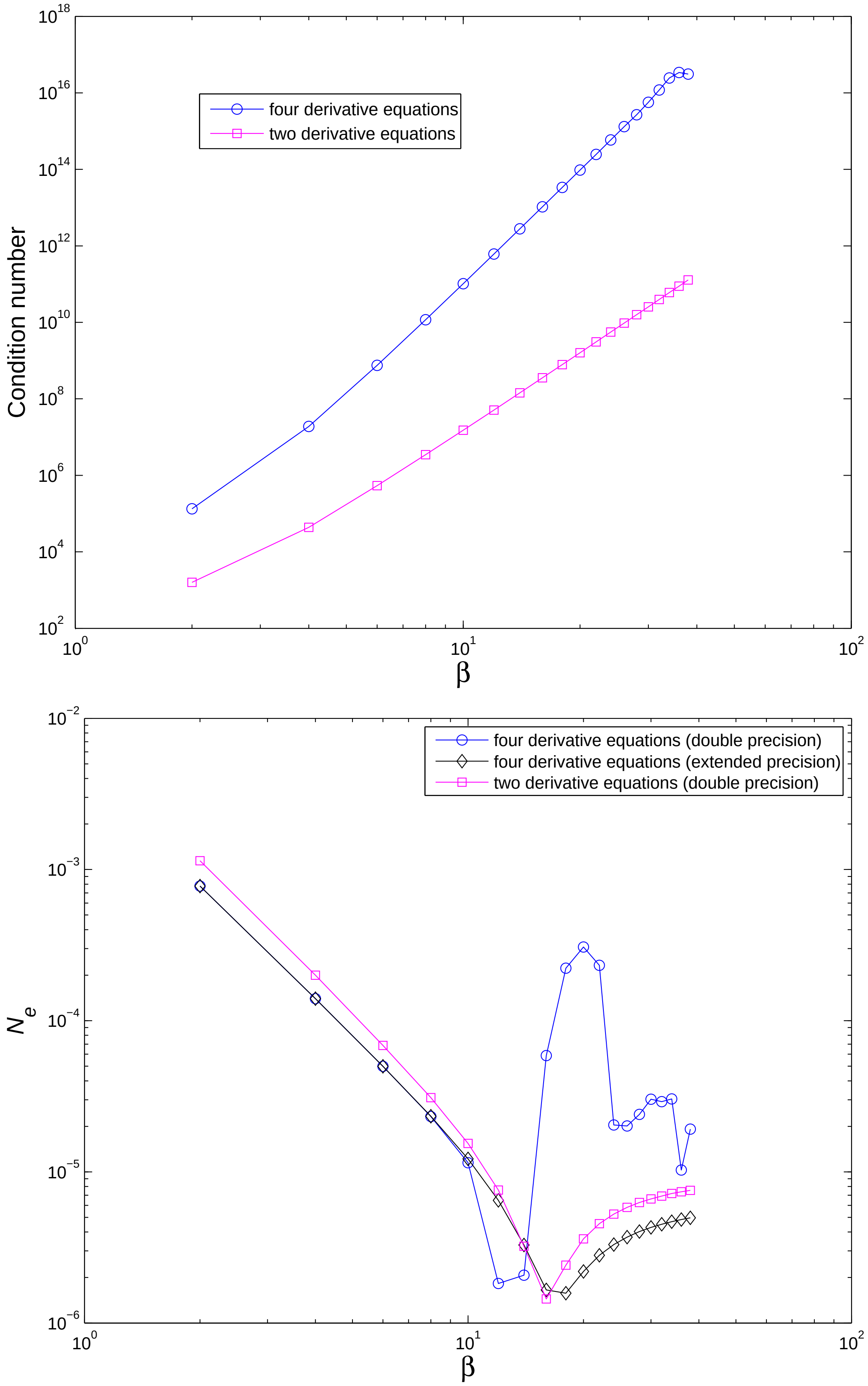

Figure 13: PDE, $\beta = (2, 4, 6, \cdots, 38)$, 31×31, $\alpha = 0.7$: Condition number of $\mathcal{C}$ and solution accuracy against the MQ width represented by β for two cases: four and two derivative equations. The 4 derivative equation case becomes unstable as β is increased. The fluctuation at large values of β is overcome by using extended precision or reducing the number of derivative equations.

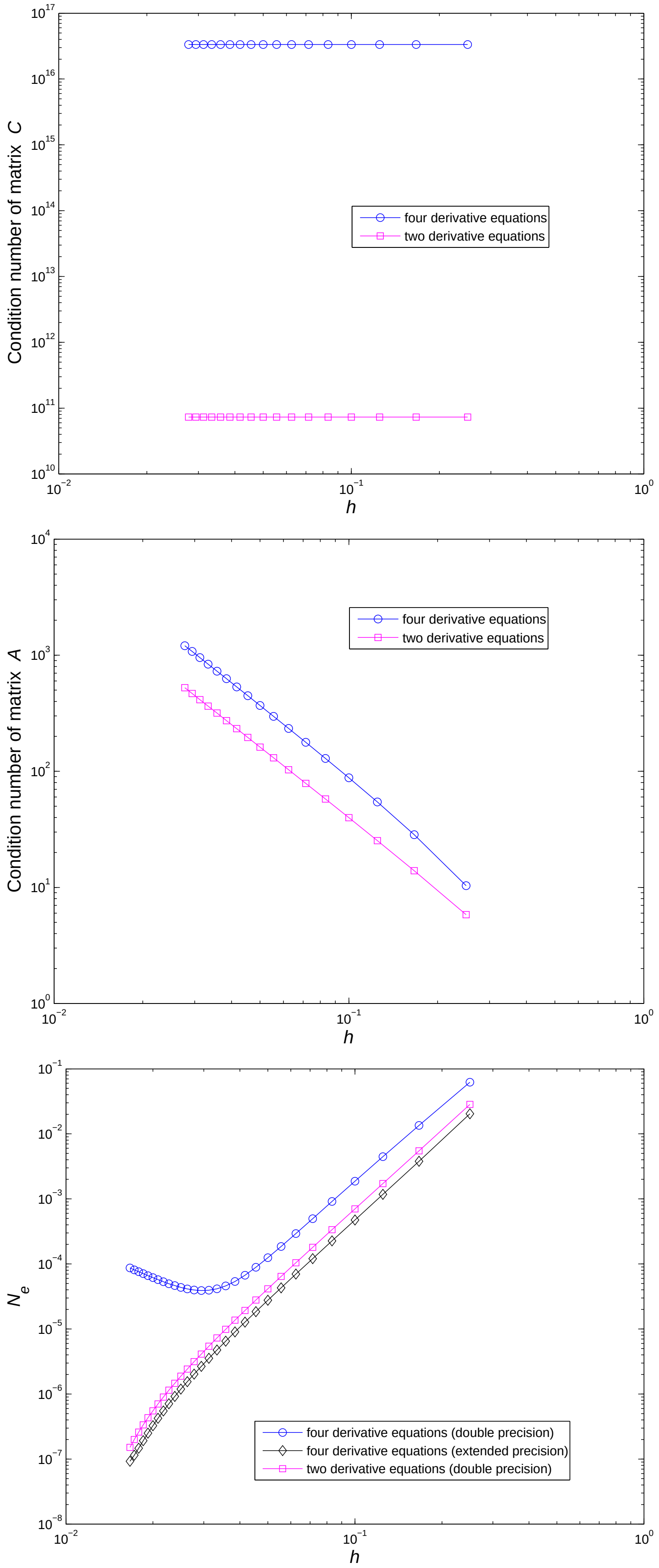

Figure 14: PDE, $5 \times 5, 7 \times 7, \cdots, 61 \times 61$, $\beta = 35$, $\alpha = 0.7$: Condition numbers of $\mathcal{C}$ and $\mathcal{A}$, and solution accuracy against grid size for two cases: four and two derivative equations. The solution converges as $O(h^{2.37})$ for the former and $O(h^{4.41})$ for the latter. The 4 derivative equation case is much less accurate due to the fact that its associated matrix $\mathcal{C}$ is ill-conditioned; by using extended precision, its performance becomes superior to the case of using 2 derivative equations.