Compact local integrated-RBF approximations for second-order elliptic differential problems

N. Mai-Duy* and T. Tran-Cong

Computational Engineering and Science Research Centre

Faculty of Engineering and Surveying

University of Southern Queensland, Toowoomba, QLD 4350, Australia

Submitted to *Journal of Computational Physics*, 30-Oct-2010; revised, 21-Feb-2011

*Corresponding author: E-mail nam.mai-duy@usq.edu.au, Telephone +61 7 4631 2748, Fax +61 7 4631 2526

Abstract This paper presents a new compact approximation method for the discretisation of second-order elliptic equations in one and two dimensions. The problem domain, which can be rectangular or non-rectangular, is represented by a Cartesian grid. On stencils, which are three nodal points for one-dimensional problems and nine nodal points for two-dimensional problems, the approximations for the field variable and its derivatives are constructed using integrated radial basis functions (IRBFs). Several pieces of information about the governing differential equation on the stencil are incorporated into the IRBF approximations by means of the constants of integration. Numerical examples indicate that the proposed technique yields a very high rate of convergence with grid refinement.

Keywords: Compact local approximations, high-order approximations, elliptic problems, integrated radial basis functions.

1 Introduction

In the context of finite differences (FDs), approximation schemes can be divided into two categories, namely standard and compact. For standard schemes, on each stencil, derivative values of the field variable u at the central grid point are expressed as linear combinations of nodal values of u only (e.g. [1]). High-order standard approximations require a relatively large stencil. For example, on a one-dimensional (1D) grid, the fourth-order scheme involves a stencil of up to 5 grid points $(x_{j-2}, x_{j-1}, x_j, x_{j+1}, x_{j+2})$. The centred approximations for the first and second derivatives of the function u are estimated as

$$u'_j = \frac{1}{h}\left(\frac{1}{12}u_{j-2} - \frac{2}{3}u_{j-1} + \frac{2}{3}u_{j+1} - \frac{1}{12}u_{j+2}\right) \tag{1}$$

$$u''_j = \frac{1}{h^2}\left(-\frac{1}{12}u_{j-2} + \frac{4}{3}u_{j-1} - \frac{5}{2}u_j + \frac{4}{3}u_{j+1} - \frac{1}{12}u_{j+2}\right) \tag{2}$$

where h is the grid size, $u_j = u(x_j)$, $u'_j = du(x_j)/dx$ and $u''_j = d^2u(x_j)/dx^2$. When large stencils are employed, the sparseness of the system matrix is reduced. In addition, special treatments are needed for grid points close to the boundary as the stencils do not entirely lie within the problem domain. For compact schemes, the FD formulas can be constructed using the Padé approximation (e.g. [2,3]). The compact approximations are based on not only nodal function values but also nodal derivative values. One can have high-order approximations on a relatively small stencil. For example, in one dimension, the fourth-order centred compact scheme requires a stencil of 3 grid points only

$$u'_{j-1} + 4u'_j + u'_{j+1} = \frac{1}{h}\left(-3u_{j-1} + 3u_{j+1}\right) \tag{3}$$

$$u''_{j-1} + 10u''_j + u''_{j+1} = \frac{1}{h^2}\left(12u_{j-1} - 24u_j + 12u_{j+1}\right) \tag{4}$$

Radial basis functions (RBFs), which are radially symmetric about their centres, are known as a powerful tool for approximation of multi-variable data and functions. An RBF is of the form $G_i(\mathbf{x}) = \phi(\|\mathbf{x}-\mathbf{c}_i\|)$, where $\mathbf{x}$ is a field point, $\mathbf{c}_i$ is a centre and $\|.\|$ denotes a norm that is usually Euclidean. Over the past two decades, RBFs have successfully been used for solving differential equations (e.g. [4]). They were first developed as a global collocation method [5]. The RBF approximations representing the field variable and its derivatives at a point involve every point within the domain of analysis. Advantages of global RBF schemes include (i) fast convergence (spectral accuracy for some RBFs such as the multiquadric and Gaussian functions); (ii) meshless nature and (iii) simple implementation. On the other hand, global RBF matrices are fully populated and tend to be much more ill-conditioned as the number of RBFs increases. Recent RBF research focused on resolving these drawbacks. One effective and popular way is to employ local RBF approximations. Works reported in this research direction include [6-13]. The RBF approximations at a point involve some surrounding points only. Local schemes lead to a sparse system which saves computer storage space and facilitates the employment of a much larger number of nodes. However, these positive properties do not come for free.

Their solution accuracy is significantly deteriorated.

Many types of RBF, e.g. multiquadric and Gaussian, involve a free shape/width parameter. Small and large values of the width make these RBFs peaked and flat, respectively. Numerical experiments show that the RBF width has a strong influence on the quality of approximation. This parameter can be used as an alternative to the mesh size in the control of the solution accuracy. Refining the mesh leads to an increase in computational and storage costs, while increasing the RBF width can be implemented without additional costs [14]. For the latter, according to the uncertainty/trade-off principle [15], high accuracy is achieved at the cost of low stability.

Integrated RBFs (IRBFs), in which the highest-order derivatives under consideration are approximated using RBFs and approximate expressions for lower-order derivatives are then obtained through integration, have several advantages over conventional differentiated RBFs [16,17]. The purposes of using integration (a smoothing operator) to construct the approximants are (i) to avoid the reduction in convergence rate caused by differentiation and (ii) to improve the numerical stability of a discrete solution. The integration process also gives rise to arbitrary constants that serve as additional expansion coefficients, and therefore facilitate the employment of some extra equations. This distinguishing feature of the integral formulation provides effective ways to overcome some difficulties associated with the conventional differential approach, namely (i) the implementation of multiple boundary conditions [18,19,20]; (ii) the imposition of the governing equation on the boundary [21] and (iii) the imposition of high-order continuity of the approximate solution across subdomain interfaces [22]. When IRBFs are implemented in local form, their solution accuracy is also reduced [23].

Compact local RBF schemes were studied in, e.g., [8,9,10]. They are based on the Hermite interpolation property of RBFs. Let $\mathcal{L}$ be some linear differential operator and let $X = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_j, \cdots, \mathbf{x}_n)$ be a stencil that corresponds to $\mathbf{x}_j$. Suppose that the values of u

and $\mathcal{L}u$ are specified at $C = (\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_p)$ and $\bar{C} = (\bar{\mathbf{c}}_1, \bar{\mathbf{c}}_2, \cdots, \bar{\mathbf{c}}_q)$, respectively. It is noted that C and $\bar{C}$ are subsets of X. The RBF approximation for u is constructed as

$$u(\mathbf{x}) = \sum_{i=1}^{p} w_i \phi(\|\mathbf{x} - \mathbf{c}_i\|) + \sum_{i=1}^{q} \lambda_i \mathcal{L}\phi(\|\mathbf{x} - \bar{\mathbf{c}}_i\|) \tag{5}$$

where $p \leq n$; $q < n$; $\{w_i\}_{i=1}^{p}$ and $\{\lambda_i\}_{i=1}^{q}$ are sets of unknown coefficients; and $\mathcal{L}$ acts on ϕ which is viewed as a function of the second variable, i.e. $\bar{\mathbf{c}}$. Approximations to derivatives of u are then obtained through differentiation. Numerical results showed that compact local RBF methods yield superior accuracy and faster convergence than standard local RBF methods.

In this paper, the use of integration to construct compact local RBF schemes is proposed. It will be shown that the constants of integration provide a natural way to include in the IRBF approximations the differential equation at some grid points on the stencil. The paper is organised as follows. A brief review of IRBFs is given in Section 2. In Section 3, the proposed technique is described. Numerical examples are presented in Section 4. Section 5 concludes the paper.

2 Integrated-RBF approximations

Consider a function $u(\mathbf{x})$ with $\mathbf{x} = (x, y)^T$. The IRBF expressions representing u and its derivatives are constructed as follows [17].

In the x direction, the second derivative of u is first decomposed into RBFs

$$\frac{\partial^2 u(\mathbf{x})}{\partial x^2} = \sum_{i=1}^{N} w_i^{[x]} G_i^{[x]}(\mathbf{x}) \tag{6}$$

where N is the number of RBFs; $\left\{w_i^{[x]}\right\}_{i=1}^{N}$ the set of weights/coefficients; and $\left\{G_i^{[x]}(\mathbf{x})\right\}_{i=1}^{N}$

the set of RBFs. Approximations to the first derivative and the function are then obtained through integration

$$\frac{\partial u(\mathbf{x})}{\partial x} = \sum_{i=1}^{N} w_i^{[x]} H_i^{[x]}(\mathbf{x}) + C_1^{[x]}(y) \tag{7}$$

$$u^{[x]}(\mathbf{x}) = \sum_{i=1}^{N} w_i^{[x]} \overline{H}_i^{[x]}(\mathbf{x}) + xC_1^{[x]}(y) + C_2^{[x]}(y) \tag{8}$$

where $H_i^{[x]}(\mathbf{x}) = \int G_i^{[x]}(\mathbf{x})dx$; $\overline{H}_i^{[x]}(\mathbf{x}) = \int H_i^{[x]}(\mathbf{x})dx$; and $C_1^{[x]}(y)$ and $C_2^{[x]}(y)$ are the constants of integration which are univariate functions of the variable y. For points that have the same y coordinate, functions $C_1^{[x]}(y)$ and $C_2^{[x]}(y)$ on the RHS of (7) and (8) will have the same value.

For the y direction, in the same way, one has

$$\frac{\partial^2 u(\mathbf{x})}{\partial y^2} = \sum_{i=1}^{N} w_i^{[y]} G_i^{[y]}(\mathbf{x}) \tag{9}$$

$$\frac{\partial u(\mathbf{x})}{\partial y} = \sum_{i=1}^{N} w_i^{[y]} H_i^{[y]}(\mathbf{x}) + C_1^{[y]}(x) \tag{10}$$

$$u^{[y]}(\mathbf{x}) = \sum_{i=1}^{N} w_i^{[y]} \overline{H}_i^{[y]}(\mathbf{x}) + yC_1^{[y]}(x) + C_2^{[y]}(x) \tag{11}$$

It can be seen that there are two approximate values of the function u at point $(\mathbf{x})$, namely $u^{[x]}(\mathbf{x})$ and $u^{[y]}(\mathbf{x})$. These two values need be forced to be identical.

3 Proposed method

Consider the following elliptic differential problem

$$\mathcal{L}u = f, \quad \mathbf{x} \in \Omega \tag{12}$$

$$u = r, \quad \mathbf{x} \in \partial\Omega_1 \tag{13}$$

$$\frac{\partial u}{\partial n} = s, \quad \mathbf{x} \in \partial\Omega_2 \tag{14}$$

where f, r and s are some prescribed functions; $\mathcal{L}$ a second-order linear differential operator, e.g. the Laplacian; $\partial\Omega_1$ and $\partial\Omega_2$ parts of the boundary of the region Ω; and n the coordinate normal to $\partial\Omega_2$. This study is concerned with the development of compact local approximations based on IRBFs for the discretisation of (12) in one and two dimensions: (i) Dirichlet and Neumann boundary conditions for 1D domains and 2D domains of rectangular shape and (ii) Dirichlet boundary conditions only for 2D domains of non-rectangular shape.

3.1 1D problems

The domain of interest, $a \leq x \leq b$, is discretised by a set of points. The IRBF expressions (6), (7) and (8) reduce to

$$\frac{d^2u(x)}{dx^2} = \sum_{i=1}^{N} w_i G_i(x) \tag{15}$$

$$\frac{du(x)}{dx} = \sum_{i=1}^{N} w_i H_i(x) + C_1 \tag{16}$$

$$u(x) = \sum_{i=1}^{N} w_i \overline{H}_i(x) + xC_1 + C_2 \tag{17}$$

where C_1 and C_2 are simply constant values. Since C_1 and C_2 are unknown, we treat them like the RBF coefficients. The length of the present coefficient vector is thus larger

than that of the conventional/differential approach, which allows the addition of two extra equations to the system that represents the conversion of the RBF space into the physical space. In this study, the two extra equations are taken to represent information about (12).

Consider an interior grid point, denoted by x_0. On its associated stencil $[x_1, x_2, x_3]$ ($x_1 < x_2 < x_3$, $x_0 \equiv x_2$), we can represent the conversion system as a matrix-vector multiplication

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ f_1 \\ f_3 \end{pmatrix} = \underbrace{\begin{bmatrix} \overline{\mathcal{H}} \\ \mathcal{K} \end{bmatrix}}_{\mathcal{C}} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ C_1 \\ C_2 \end{pmatrix} \tag{18}$$

where $u_i = u(x_i)$; $f_i = f(x_i)$; $\mathcal{C}$ is the conversion matrix; and $\overline{\mathcal{H}}$ is defined as

$$\overline{\mathcal{H}} = \begin{bmatrix} \overline{H}_1(x_1), & \overline{H}_2(x_1), & \overline{H}_3(x_1), & x_1, & 1 \\ \overline{H}_1(x_2), & \overline{H}_2(x_2), & \overline{H}_3(x_2), & x_2, & 1 \\ \overline{H}_1(x_3), & \overline{H}_2(x_3), & \overline{H}_3(x_3), & x_3, & 1 \end{bmatrix}$$

If $\mathcal{L}u$ takes the form, for instance, $\mathcal{L}u = d^2u/dx^2 + u$, one will have

$$\mathcal{K} = \begin{bmatrix} G_1(x_1) + \overline{H}_1(x_1), & G_2(x_1) + \overline{H}_2(x_1), & G_3(x_1) + \overline{H}_3(x_1), & x_1, & 1 \\ G_1(x_3) + \overline{H}_1(x_3), & G_2(x_3) + \overline{H}_2(x_3), & G_3(x_3) + \overline{H}_3(x_3), & x_3, & 1 \end{bmatrix}$$

It can be seen that $\mathcal{C}$ is an 5×5 matrix. Solving (18) yields

$$\begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ C_1 \\ C_2 \end{pmatrix} = \mathcal{C}^{-1} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ f_1 \\ f_3 \end{pmatrix} \tag{19}$$

which maps the vector of nodal values of the function and the governing equation to the vector of RBF coefficients including the two integration constants.

Approximate expressions for u and its derivatives in the physical space are obtained by substituting (19) into (17), (16) and (15)

$$u(x) = \left[\overline{H}_1(x), \overline{H}_2(x), \overline{H}_3(x), x, 1\right] \mathcal{C}^{-1} \begin{pmatrix} \widehat{u} \\ \widehat{f} \end{pmatrix} \tag{20}$$

$$\frac{du(x)}{dx} = \left[H_1(x), H_2(x), H_3(x), 1, 0\right] \mathcal{C}^{-1} \begin{pmatrix} \widehat{u} \\ \widehat{f} \end{pmatrix} \tag{21}$$

$$\frac{d^2u(x)}{dx^2} = \left[G_1(x), G_2(x), G_3(x), 0, 0\right] \mathcal{C}^{-1} \begin{pmatrix} \widehat{u} \\ \widehat{f} \end{pmatrix} \tag{22}$$

where $x_1 \leq x \leq x_3$, $\widehat{u} = (u_1, u_2, u_3)^T$ and $\widehat{f} = (f_1, f_3)^T$. They can be rewritten in the form

$$u(x) = \sum_{i=1}^{3} \varphi_i(x) u_i + \varphi_4(x) f_1 + \varphi_5(x) f_3 \tag{23}$$

$$\frac{du(x)}{dx} = \sum_{i=1}^{3} \frac{d\varphi_i(x)}{dx} u_i + \frac{d\varphi_4(x)}{dx} f_1 + \frac{d\varphi_5(x)}{dx} f_3 \tag{24}$$

$$\frac{d^2u(x)}{dx^2} = \sum_{i=1}^{3} \frac{d^2\varphi_i(x)}{dx^2} u_i + \frac{d^2\varphi_4(x)}{dx^2} f_1 + \frac{d^2\varphi_5(x)}{dx^2} f_3 \tag{25}$$

where $\{\varphi_i(x)\}_{i=1}^{5}$ is the set of IRBFs in the physical space. It can be seen from (23)-(25) that the IRBF approximations for the 3-point stencil scheme are expressed not only in terms of nodal function values but also in terms of nodal derivative values.

3.1.1 Dirichlet boundary conditions

Assume that the values of u are given at $x = a$ and $x = b$. By collocating (23)-(25) at $x = x_0$ and then substituting the obtained results into (12), we acquire an algebraic equation in terms of nodal variable values for that node. If one applies this task to every interior node and then replaces $u(a)$ and $u(b)$ with given values, this will lead to a determined system of algebraic equations for the unknown values of u at the interior nodes.

3.1.2 Dirichlet and Neumann boundary conditions

Assume that the value of $\partial u/\partial x$ and u are given at $x = a$ and $x = b$, respectively. Unlike the Dirichlet case, the value of u at $x = a$ is unknown. As a result, one needs to generate one additional algebraic equation for this unknown. The system of algebraic equations thus consists of $(N-1)$ equations for $(N-1)$ unknown values of u, where N is defined here as the number of grid nodes including the two boundary points. We study two ways of implementing the Neumann boundary condition.

Implementation 1: The Neumann boundary condition is incorporated into the IRBF approximations. The construction of a stencil associated with the grid node next to the boundary point $x = a$ needs be modified. The first extra equation in the conversion system (18) is now used to represent the Neumann boundary condition, i.e. $f_1 = du(a)/dx$ and the first row of $\mathcal{K}$ being $[H_1(a), H_2(a), H_3(a), 1, 0]$, rather than the governing equation at $x = a$. Using this stencil, we generate two algebraic equations, namely the governing

equation at the centered point of the stencil and the governing equation at the boundary point $x = a$, for the final system. The latter is an additional algebraic equation for the value of u at $x = a$. All $(N - 1)$ algebraic equations needed are thus derived from the governing equation only.

Implementation 2: The Neumann boundary condition is imposed in a direct manner. There are no modifications for the construction of a stencil associated with the grid node next to the boundary point $x = a$. The final system of equations is now added with one extra equation that is generated directly from (21) with $x = a$. There are thus $(N - 2)$ algebraic equations derived from the governing equation and one equation from the Neumann boundary condition.

Implementation 2 is more straightforward to program than Implementation 1.

3.2 2D problems

Two types of the problem domain, rectangular and non-rectangular, are considered. For the former, the compact IRBF approximations on stencils are of similar forms. For the latter, stencils associated with interior nodes close to the boundary may be cut by the boundary, and the nodal points in such stencils are defined in a different way.

3.2.1 Rectangular domain

The problem domain is represented by a Cartesian grid. Consider an interior grid point, denoted by $\mathbf{x}_0$. Its associated stencil is defined as

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

where $\mathbf{x}_0 \equiv \mathbf{x}_5$. The conversion system for this 9-point stencil is formed as

$$\begin{pmatrix} \widehat{u} \\ \widehat{o} \\ \widehat{f} \end{pmatrix} = \underbrace{\begin{bmatrix} \overline{\mathcal{H}}^{[x]}, & \mathcal{O} \\ \overline{\mathcal{H}}^{[x]}, & -\overline{\mathcal{H}}^{[y]} \\ \mathcal{K}^{[x]}, & \mathcal{K}^{[y]} \end{bmatrix}}_{\mathcal{C}} \begin{pmatrix} \widehat{w}^{[x]} \\ \widehat{w}^{[y]} \end{pmatrix} \tag{26}$$

where $\widehat{u}$ and $\widehat{o}$ are vectors of length 9; $\widehat{w}^{[x]}$ and $\widehat{w}^{[y]}$ vectors of length 15; $\mathcal{O}$, $\overline{\mathcal{H}}^{[x]}$ and $\overline{\mathcal{H}}^{[y]}$ matrices of dimensions 9×15; $\mathcal{C}$ the conversion matrix;

$$
\begin{aligned}
\widehat{u} &= (u_1, \cdots, u_9)^T \\
\widehat{w}^{[x]} &= \left(w_1^{[x]}, \cdots, w_9^{[x]}, C_1^{[x]}(y_1), C_1^{[x]}(y_2), C_1^{[x]}(y_3), C_2^{[x]}(y_1), C_2^{[x]}(y_2), C_2^{[x]}(y_3)\right)^T \\
\widehat{w}^{[y]} &= \left(w_1^{[y]}, \cdots, w_9^{[y]}, C_1^{[y]}(x_1), C_1^{[y]}(x_4), C_1^{[y]}(x_7), C_2^{[y]}(x_1), C_2^{[y]}(x_4), C_2^{[y]}(x_7)\right)^T \\
\overline{\mathcal{H}}^{[x]} &= \begin{bmatrix}
\overline{H}_1^{[x]}(\mathbf{x}_1), & \cdots, & \overline{H}_9^{[x]}(\mathbf{x}_1), & x_1, & 0, & 0, & 1, & 0, & 0 \\
\overline{H}_1^{[x]}(\mathbf{x}_2), & \cdots, & \overline{H}_9^{[x]}(\mathbf{x}_2), & 0, & x_2, & 0, & 0, & 1, & 0 \\
\overline{H}_1^{[x]}(\mathbf{x}_3), & \cdots, & \overline{H}_9^{[x]}(\mathbf{x}_3), & 0, & 0, & x_3, & 0, & 0, & 1 \\
\overline{H}_1^{[x]}(\mathbf{x}_4), & \cdots, & \overline{H}_9^{[x]}(\mathbf{x}_4), & x_4, & 0, & 0, & 1, & 0, & 0 \\
\overline{H}_1^{[x]}(\mathbf{x}_5), & \cdots, & \overline{H}_9^{[x]}(\mathbf{x}_5), & 0, & x_5, & 0, & 0, & 1, & 0 \\
\overline{H}_1^{[x]}(\mathbf{x}_6), & \cdots, & \overline{H}_9^{[x]}(\mathbf{x}_6), & 0, & 0, & x_6, & 0, & 0, & 1 \\
\overline{H}_1^{[x]}(\mathbf{x}_7), & \cdots, & \overline{H}_9^{[x]}(\mathbf{x}_7), & x_7, & 0, & 0, & 1, & 0, & 0 \\
\overline{H}_1^{[x]}(\mathbf{x}_8), & \cdots, & \overline{H}_9^{[x]}(\mathbf{x}_8), & 0, & x_8, & 0, & 0, & 1, & 0 \\
\overline{H}_1^{[x]}(\mathbf{x}_9), & \cdots, & \overline{H}_9^{[x]}(\mathbf{x}_9), & 0, & 0, & x_9, & 0, & 0, & 1
\end{bmatrix} \\
\overline{\mathcal{H}}^{[y]} &= \begin{bmatrix}
\overline{H}_1^{[y]}(\mathbf{x}_1), & \cdots, & \overline{H}_9^{[y]}(\mathbf{x}_1), & y_1, & 0, & 0, & 1, & 0, & 0 \\
\overline{H}_1^{[y]}(\mathbf{x}_2), & \cdots, & \overline{H}_9^{[y]}(\mathbf{x}_2), & y_2, & 0, & 0, & 1, & 0, & 0 \\
\overline{H}_1^{[y]}(\mathbf{x}_3), & \cdots, & \overline{H}_9^{[y]}(\mathbf{x}_3), & y_3, & 0, & 0, & 1, & 0, & 0 \\
\overline{H}_1^{[y]}(\mathbf{x}_4), & \cdots, & \overline{H}_9^{[y]}(\mathbf{x}_4), & 0, & y_4, & 0, & 0, & 1, & 0 \\
\overline{H}_1^{[y]}(\mathbf{x}_5), & \cdots, & \overline{H}_9^{[y]}(\mathbf{x}_5), & 0, & y_5, & 0, & 0, & 1, & 0 \\
\overline{H}_1^{[y]}(\mathbf{x}_6), & \cdots, & \overline{H}_9^{[y]}(\mathbf{x}_6), & 0, & y_6, & 0, & 0, & 1, & 0 \\
\overline{H}_1^{[y]}(\mathbf{x}_7), & \cdots, & \overline{H}_9^{[y]}(\mathbf{x}_7), & 0, & 0, & y_7, & 0, & 0, & 1 \\
\overline{H}_1^{[y]}(\mathbf{x}_8), & \cdots, & \overline{H}_9^{[y]}(\mathbf{x}_8), & 0, & 0, & y_8, & 0, & 0, & 1 \\
\overline{H}_1^{[y]}(\mathbf{x}_9), & \cdots, & \overline{H}_9^{[y]}(\mathbf{x}_9), & 0, & 0, & y_9, & 0, & 0, & 1
\end{bmatrix}
\end{aligned}
$$

$\widehat{o}$ and $\mathcal{O}$ the vector and matrix of zeros; and $\mathcal{K}^{[x]}$, $\mathcal{K}^{[y]}$ and $\widehat{f}$ will be defined shortly. In (26), there are three sub-systems. In the first sub-system ($\widehat{u} = \overline{\mathcal{H}}^{[x]}\widehat{w}^{[x]}$), the function

$u^{[x]}(\mathbf{x})$ is collocated at the grid nodes. In the second sub-system ($\overline{\mathcal{H}}^{[x]}\widehat{w}^{[x]} = \overline{\mathcal{H}}^{[y]}\widehat{w}^{[y]}$), the approximate nodal values of u obtained from integrating RBFs with respect to x are forced to be identical to those obtained from integrating RBFs with respect to y. The third sub-system, $\widehat{f} = \mathcal{K}^{[x]}\widehat{w}^{[x]} + \mathcal{K}^{[y]}\widehat{w}^{[y]}$, is used to enforce the governing equation (12) at some nodes. For illustrative purposes, assume that $\mathcal{L}$ is the Laplace operator. We consider the following two cases, which exploit symmetries about the central point $\mathbf{x}_5$,

Case 1: Extra equations are used to represent (12) at $(\mathbf{x}_2, \mathbf{x}_4, \mathbf{x}_6, \mathbf{x}_8)$, i.e.

$$\widehat{f} = (f_2, f_4, f_6, f_8)^T$$

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

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

If Case 1 is chosen, the present approximations will hereafter be named Compact 9-point IRBF (1) or simply Scheme 1.

Case 2: Extra equations are used to represent (12) at $(\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_7, \mathbf{x}_9)$, i.e.

$$\widehat{f} = (f_1, f_3, f_7, f_9)^T$$

$$\mathcal{K}^{[x]} = \begin{bmatrix} G_1^{[x]}(\mathbf{x}_1), & \cdots, & G_9^{[x]}(\mathbf{x}_1), & 0, & 0, & 0, & 0, & 0, & 0 \\ G_1^{[x]}(\mathbf{x}_3), & \cdots, & G_9^{[x]}(\mathbf{x}_3), & 0, & 0, & 0, & 0, & 0, & 0 \\ G_1^{[x]}(\mathbf{x}_7), & \cdots, & G_9^{[x]}(\mathbf{x}_7), & 0, & 0, & 0, & 0, & 0, & 0 \\ G_1^{[x]}(\mathbf{x}_9), & \cdots, & G_9^{[x]}(\mathbf{x}_9), & 0, & 0, & 0, & 0, & 0, & 0 \end{bmatrix}$$

$$\mathcal{K}^{[y]} = \begin{bmatrix} G_1^{[y]}(\mathbf{x}_1), & \cdots, & G_9^{[y]}(\mathbf{x}_1), & 0, & 0, & 0, & 0, & 0, & 0 \\ G_1^{[y]}(\mathbf{x}_3), & \cdots, & G_9^{[y]}(\mathbf{x}_3), & 0, & 0, & 0, & 0, & 0, & 0 \\ G_1^{[y]}(\mathbf{x}_7), & \cdots, & G_9^{[y]}(\mathbf{x}_7), & 0, & 0, & 0, & 0, & 0, & 0 \\ G_1^{[y]}(\mathbf{x}_9), & \cdots, & G_9^{[y]}(\mathbf{x}_9), & 0, & 0, & 0, & 0, & 0, & 0 \end{bmatrix}$$

If Case 2 is implemented, the present approximations will hereafter be named Compact 9-point IRBF (2) or simply Scheme 2.

For Scheme 1 and Scheme 2, the conversion matrix $\mathcal{C}$ has more columns than rows. There are an infinite number of solution $(\widehat{w}^{[x]}, \widehat{w}^{[y]})^T$ which exactly satisfy $(\widehat{u}, \widehat{o}, \widehat{f})^T = \mathcal{C}(\widehat{w}^{[x]}, \widehat{w}^{[y]})^T$. We apply the SVD technique here to find the unique solution $(\widehat{w}^{[x]}, \widehat{w}^{[y]})^T$ which minimises $\|(\widehat{w}^{[x]}, \widehat{w}^{[y]})^T\|_2$ (minimum norm solution). It is noted that the nodal values of the field variable and the governing equation on the LHS of (26) are satisfied identically. Through (26), the RBF coefficient vectors are computed as

$$\begin{pmatrix} \widehat{w}^{[x]} \\ \widehat{w}^{[y]} \end{pmatrix} = \mathcal{C}^{-1} \begin{pmatrix} \widehat{u} \\ \widehat{o} \\ \widehat{f} \end{pmatrix} \tag{27}$$

or

$$\widehat{w}^{[x]} = \mathcal{C}_{[x]}^{-1} \left(\widehat{u}, \widehat{o}, \widehat{f}\right)^T \tag{28}$$

$$\widehat{w}^{[y]} = \mathcal{C}_{[y]}^{-1} \left(\widehat{u}, \widehat{o}, \widehat{f}\right)^T \tag{29}$$

where $\mathcal{C}^{-1}$ is the pseudo inverse of the conversion matrix; and $\mathcal{C}_{[x]}^{-1}$ and $\mathcal{C}_{[y]}^{-1}$ are the first and last 15 rows of $\mathcal{C}^{-1}$, respectively.

Using (28) and (29), expressions (7), (6), (10) and (9) at $\mathbf{x} = \mathbf{x}_0$ become

$$\frac{\partial u(\mathbf{x}_0)}{\partial x} = \left[H_1^{[x]}(\mathbf{x}_0), \cdots, H_9^{[x]}(\mathbf{x}_0), 0, 1, 0, 0, 0, 0, 0\right] \mathcal{C}_{[x]}^{-1} \left(\widehat{u}, \widehat{o}, \widehat{f}\right)^T \tag{30}$$

$$\frac{\partial^2 u(\mathbf{x}_0)}{\partial x^2} = \left[G_1^{[x]}(\mathbf{x}_0), \cdots, G_9^{[x]}(\mathbf{x}_0), 0, 0, 0, 0, 0, 0, 0\right] \mathcal{C}_{[x]}^{-1} \left(\widehat{u}, \widehat{o}, \widehat{f}\right)^T \tag{31}$$

$$\frac{\partial u(\mathbf{x}_0)}{\partial y} = \left[H_1^{[y]}(\mathbf{x}_0), \cdots, H_9^{[y]}(\mathbf{x}_0), 0, 1, 0, 0, 0, 0, 0\right] \mathcal{C}_{[y]}^{-1} \left(\widehat{u}, \widehat{o}, \widehat{f}\right)^T \tag{32}$$

$$\frac{\partial^2 u(\mathbf{x}_0)}{\partial y^2} = \left[G_1^{[y]}(\mathbf{x}_0), \cdots, G_9^{[y]}(\mathbf{x}_0), 0, 0, 0, 0, 0, 0, 0\right] \mathcal{C}_{[y]}^{-1} \left(\widehat{u}, \widehat{o}, \widehat{f}\right)^T \tag{33}$$

Dirichlet boundary conditions: By collocating (12) at every interior node using (30)-(33) and then replacing the nodal values of u on the boundary with the conditions prescribed, one will obtain a determined system of algebraic equations for the unknown values of u at the interior nodes.

Dirichlet and Neumann boundary conditions: Both Implementation 1 and Implementation 2 for 1D problems are applicable here to handle Neumann boundary conditions.

3.2.2 Non-rectangular domain

The problem domain Ω is embedded in a rectangular domain which is then represented by a Cartesian grid. Grid points outside Ω are ignored and boundary nodes are generated by the intersection of the grid lines and the boundary $\partial\Omega$. It can be seen that the

preprocessing here is economical. For interior nodes whose associated stencils lie within the problem domain entirely, the compact IRBF approximations for the field variable and its derivatives have the same forms as in the rectangular-domain case. For interior nodes close to the irregular boundary, the stencils may be cut by the boundary and their shapes become non-rectangular. Consider a typical case as depicted in Figure 1. It can be seen that the boundary points generated by the x and y grid lines do not coincide with the grid nodes. We take the RBF centres as $(\mathbf{x}_1, \mathbf{x}_{2'}, \mathbf{x}_{3'}, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_8, \mathbf{x}_9)$ for the approximation of $u^{[x]}(\mathbf{x})$, $\partial u(\mathbf{x})/\partial x$ and $\partial^2 u(\mathbf{x})/\partial x^2$, and $(\mathbf{x}_1, \mathbf{x}_{2''}, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_8, \mathbf{x}_9)$ for the approximation of $u^{[y]}(\mathbf{x})$, $\partial u(\mathbf{x})/\partial y$ and $\partial^2 u(\mathbf{x})/\partial y^2$. The conversion system is constructed as follows. The function $u^{[x]}(\mathbf{x})$ is collocated at the whole set of nodal points used in the x integration process, i.e. $(\mathbf{x}_1, \mathbf{x}_{2'}, \mathbf{x}_{3'}, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_8, \mathbf{x}_9)$. The function $u^{[y]}(\mathbf{x})$ is collocated at the boundary points used in the y integration process, i.e. $(\mathbf{x}_{2''})$. The condition $u^{[x]}(\mathbf{x}) = u^{[y]}(\mathbf{x})$ is enforced at every interior point, i.e. $(\mathbf{x}_1, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_8, \mathbf{x}_9)$. The governing equation is collocated at $(\mathbf{x}_4, \mathbf{x}_6, \mathbf{x}_8)$ for Scheme 1 and $(\mathbf{x}_1, \mathbf{x}_7, \mathbf{x}_9)$ for Scheme 2. After solving the conversion system for the RBF coefficient vectors, $\widehat{w}^{[x]}$ and $\widehat{w}^{[y]}$, the approximations are expressed in terms of nodal values of the variable u and the governing equation. The remaining steps of the solution procedure are similar to those for the rectangular-domain case. It is noted that this solution procedure is restricted to problems with Dirichlet boundary conditions only.

In the case of high-order ODEs/PDEs, more integration constants arise, e.g. $\{C_1, C_2, C_3\}$ for third-order ODEs and $\{C_1, C_2, C_3, C_4\}$ for fourth-order ODEs. As a result, one is able to implement a larger number of extra equations. Since the number of extra equations used for the governing equation remain unchanged (2 for 1D problems and 4 for 2D problems), it is expected that the size of the stencil (i.e. 3 nodes for 1D problems and 9 nodes for 2D problems) is preserved.

4 Numerical examples

IRBFs are implemented with the multiquadric (MQ) function

$$G_i(\mathbf{x}) = \sqrt{(\mathbf{x} - \mathbf{c}_i)^T(\mathbf{x} - \mathbf{c}_i) + a_i} \tag{34}$$

where $\mathbf{c}_i$ and a_i are the centre and the width of the ith MQ, respectively. For each stencil, the set of nodal points is taken to be the set of MQ centres. The value of a_i is simply chosen as $a_i = \beta h_i$ in which β is a given positive number and h_i the smallest distance between the ith node and its neighbouring grid nodes. We assess the performance of the proposed method through two measures: (i) the relative discrete L_2 error defined as

$$Ne(u) = \frac{\sqrt{\sum_{i=1}^{M}(u_i - u_i^{(e)})^2}}{\sqrt{\sum_{i=1}^{M}(u_i^{(e)})^2}} \tag{35}$$

where M is the number of nodes over the whole domain and (ii) the convergence rate with respective to grid refinement defined as $O(h^\alpha)$. The latter is calculated over 2 successive grids (point(grid)-wise rate) and also over the whole set of grids used (average rate). The proposed method is validated through the solution of test problems in one and two dimensions.

4.1 Ordinary differential equations (ODEs)

4.1.1 Example 1

Consider the following second-order ODE

$$\frac{d^2u}{dx^2} = -(2\pi)^2 \sin(2\pi x), \quad 0 \leq x \leq 1 \tag{36}$$

The exact solution to this boundary value problem is chosen to be $u^{(e)}(x) = \sin(2\pi x)$.

Dirichlet boundary conditions: $u = 0$ is prescribed at $x = 0$ and $x = 1$. There are two ways to improve accuracy: (i) the grid is refined (h adaptivity) and (ii) the MQ width is increased (β adaptivity).

For h adaptivity, calculations are conducted on sets of uniformly distributed points, from 5 to 71 with an increment of 2. Results concerning the solution accuracy, the convergence rate and the matrix condition by the compact 3-point finite difference (FD) method and the compact 3-point IRBF method are presented in Table 1. Regarding the condition of the system matrix, denoted by cond(A), the two methods have similar values and they grow at the rate $O(h^{-2.02})$. In terms of accuracy, the proposed scheme converges faster. The point(grid)-wise order of accuracy is about 4 for FDs but can be up to 72.76 for IRBFs. In an average sense, the compact 3-point FD and IRBF solutions converge apparently as $O(h^{4.01})$ and $O(h^{4.78})$, respectively. At a grid of 71, the error $Ne(u)$ is 2.70×10^{-7} for FDs and 3.08×10^{-9} for IRBFs. In terms of CPU time, at a grid of 71, the elapsed times are quite small, i.e. about 0.03s for the proposed method and 0.02s for compact FDM (the computer codes were written using MATLAB and run on a Dell X86-based PC (Intel 3 GHz)). Given a grid size, the CPU time by the proposed method is greater than that by the compact FDM. However, for a prescribed accuracy, the compact FDM requires much more grid nodes than the proposed method. In this regard (accuracy), the proposed method can be more efficient than the compact FDM.

For β adaptivity, the value of β is chosen in a wide range of 2 to 100. Results calculated at a grid of 71 are shown in Figure 2. As β increases, the error $Ne(u)$ reduces significantly while the condition of the conversion matrix grows fast. Increasing the value of β appears more economical than reducing a grid size. However, at very large values of β, the solution becomes unstable. These observations are completely in accordance with the well known Schaback's "uncertainty principle" [15]. The instability phenomenon is mainly the

consequence of unbounded error propagation which appears in the solution of highly ill-conditioned system of equations. It appears that the optimal value of β is about 40 and the corresponding matrix condition number (the critical value of the condition number) is 1.39×10^{11}. On the other hand, the condition number of the system matrix $\mathcal{A}$ does not change much (about 1.98×10^3) with increasing β, probably due to the fact that the matrix $\mathcal{A}$, which is sparse, is constructed in the physical space. It is noted that, from a theoretical point of view, it is still not clear how to choose the optimal value of the MQ width. Unlike global IRBF versions ($\beta = 1$ is a preferred value), the present compact IRBF method can work well with a wide range of β. For instance, good accuracy, i.e. $Ne(u) < 1.e-6$, is obtained in a range of $20 \leq \beta \leq 100$.

Dirichlet and Neumann boundary conditions: $du/dx = 2\pi$ and $u = 0$ are prescribed at $x = 0$ and $x = 1$, respectively. Both Implementation 1, in which the Neumann boundary condition is included in the IRBF approximations, and Implementation 2, in which the Neumann boundary condition is imposed in a direct manner, are employed. Figure 3 presents the behaviours of the matrix condition number and the solution accuracy against the grid size. The two implementations produce solutions that have similar degrees of accuracy and converge apparently as $O(h^{4.45})$. This similar behaviour is probably due to the fact that the two implementations use exactly the same information. In terms of the matrix condition number, Implementation 1 is slightly more stable than Implementation 2. The former (algebraic equations derived from the governing equation only) grows at $O(h^{-1.94})$, while the latter (algebraic equations derived from the governing equation and the Neumann boundary condition) at $O(h^{-2.47})$.

Non-uniform grids: Such grids are generated here using [24]

$$x_i = \frac{1}{2} + \alpha \sinh\left(\eta\left(1 - \frac{i-1}{N-1}\right) + \theta\frac{i-1}{N-1}\right) \tag{37}$$

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 four values $\alpha = \{2, 1, 1/2, 1/3\}$ employed with $\beta = \{40, 40, 35, 30\}$, respectively (numerical studies indicate that the optimal value of β is reduced as α decreases). Figure 4 shows results obtained by the present method for Dirichlet boundary conditions. It can be seen that larger values of α (i.e. lower non-uniformity) result in slightly better accuracy and slightly lower matrix condition. It is noted that there is no steep variation in the solution under consideration. Overall, the present method yields quite similar performances over non-uniform grids. The matrix condition number grows slowly (about $O(h^{-2.0})$), while the solution converges fast (about $O(h^{4.4})$).

Comparison with other RBF methods: Results obtained by the present method are also compared with those by the global IRBF method and the compact local Hermite RBF method.

For the global IRBF method [16], the system matrix is non-symmentric and fully populated in contrast to the present scheme, where the system is symmetric and all the nonzero coefficients align themselves along three diagonals. In addition, for stable calculations of global IRBF approximations, one is allowed to use small values of β. Figure 5 shows results obtained by the global IRBF method using $\beta = 1$ and by the present method using $\beta = 40$. It can be seen that the present method performs much better than the global IRBFN method in terms of both the solution accuracy and the matrix condition.

Following the work of Wright and Fornberg [10], we also implement here a compact 3-point scheme based on Hermite MQ interpolation, which is named compact HRBF. Unlike the method in [10], a direct way of computing the interpolant is employed for compact HRBF. As used in [10], the RBF width is taken here in the form of ε (i.e. $\varepsilon = 1/a$). Decreasing ε is equivalent to increasing a. We employ uniformly distributed nodes, varying from 5 to 71 with increment of 2, to represent the domain. Figure 6 displays the error Ne against

the grid size h for three values of ε by compact HRBF and IRBF schemes. It can be seen that higher levels of accuracy are obtained with the present IRBF scheme. In addition, at a small value $\varepsilon = 1.5$, IRBF is more stable than HRBF. However, if the Contour-Padé algorithm is employed, the latter is able to work for small values of ε and one can thus obtain the optimal solution over a full range of ε, i.e. $\varepsilon \geq 0$. This point will be illustrated in Section 4.2.2.

4.1.2 Example 2

Find u such that

$$\frac{d^2u}{dx^2} + \frac{du}{dx} + u = -\exp(-5x)\left[9979\sin(100x) + 900\cos(100x)\right], \quad 0 \leq x \leq 1 \tag{38}$$

$$u(0) = 0 \tag{39}$$

$$u(1) = \sin(100)\exp(-5) \tag{40}$$

The exact solution can be verified to be

$$u^{(e)}(x) = \sin(100x)\exp(-5x) \tag{41}$$

which is highly oscillatory as shown in Figure 7.

It is different from Example 1 that the differential equation here involves three terms and its exact solution has much more complex shape. For the latter, a large number of nodes is required for an accurate simulation.

Figure 8 compares the matrix condition and the solution accuracy for various values of h between the local 3-point IRBF method and the present compact 3-point IRBF method. Including information about (38) in the IRBF approximations leads to a significant improvement in accuracy and rate of convergence. The solution converges at an average

rate of 1.94 by the local IRBF method and of 4.70 with local rates being up to 44.44 by the present compact method. The matrix condition grows as $O(h^{-2.00})$ for both methods. Results concerning β adaptivity are presented in Figures 9; remarks here are similar to those in Example 1.

4.2 Partial differential equations (PDEs)

4.2.1 Example 1

The PDE under consideration here is taken in the form

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -2\pi^2 \sin(\pi x) \sin(\pi y) \tag{42}$$

The domain of interest is a unit square $0 \leq x, y \leq 1$. The exact solution to (42) is $u^{(e)}(x, y) = \sin(\pi x) \sin(\pi y)$. The local 9-point IRBF method and two schemes of the present compact 9-point IRBF method are employed.

Dirichlet boundary conditions: $u = 0$ is specified on the entire boundary.

To study the effect of h adaptivity, we use grids of $(5 \times 5, 7 \times 7, \cdots, 37 \times 37)$ and $\beta = 45$. The obtained results are shown in Figure 10, which plots the matrix condition and the error against h. For the local 9-point IRBF method, Scheme 1 and Scheme 2, the solution converges apparently as $O(h^{2.05})$, $O(h^{3.80})$ and $O(h^{4.13})$, respectively, while the matrix condition grows as $O(h^{-2.05})$, $O(h^{-1.99})$ and $O(h^{-1.99})$, respectively. Scheme 1 and Scheme 2 produce much more accurate results and better condition numbers of the system matrix than the local 9-point IRBF method. However, CPU times consumed by Scheme 1 and Scheme 2 are slightly greater than that by the local 9-point IRBF method. For example, at a grid of 37×37, the elapsed time is about 4.7s for Scheme 1, 4.7s for Scheme 2 and 4.6s for the local 9-point IRBF method (the computer codes were written using MATLAB

and run on a Dell X86-based PC (Intel 3 GHz)).

To study the effect of β adaptivity, a wide range of β is employed. Figure 11 plots $Ne(u)$ against β. Both schemes can work at large values of β. The error $Ne(u)$ is a decreasing function of β up to the value of 60 for Scheme 1 and 25 for Scheme 2.

Scheme 1 outperforms Scheme 2 regarding accuracy for both h and β adaptivities.

Dirichlet and Neumann boundary conditions: Dirichlet boundary conditions are prescribed on $x = 0$ and $x = 1$ with $0 \leq y \leq 1$, while Neumann boundary conditions are specified on $y = 0$ and $y = 1$ with $0 < x < 1$. As shown in Section 4.1.1, the two implementations of a Neumann boundary condition produce similar degrees of accuracy. We employ Implementation 2 here. The matrix condition and solution accuracy by Scheme 1 and Scheme 2 are shown in Figure 12. Scheme 1 yields slightly larger condition numbers but much more accurate results than Scheme 2. The compact local IRBFN solution converges as $O(h^{3.90})$ for Scheme 1 and $O(h^{4.04})$ for Scheme 2. It should be pointed out that these rates are as high as those for the case of Dirichlet boundary conditions only.

4.2.2 Example 2

Example 2 is similar to Example 1, except that the exact solution now takes the form $u^{(e)}(x, y) = \exp\left(-(x - 1/4)^2 - (y - 1/2)^2\right) \sin(\pi x) \cos(2\pi y)$. This problem was solved using the RBF-FD (finite difference formulas generated by RBFs) and RBF-HFD (compact FD formulas generated by Hermite RBFs) schemes [10]. Solutions to this problem by RBF-FD and RBF-HFD were obtained with MQs and equispaced grids. The stencils used in Scheme 1 and RBF-HFD are of the same size and they use the same set of nodes for incorporating the governing PDE. For comparison purposes, the MQ's free parameter is used here in the form of ε ($\varepsilon = 1/a$) and the solution accuracy is measured by means of the maximum norm error.

For RBF-FD and RBF-HFD, the Contour-Padé algorithm is employed, which allows stable calculations of the interpolant for a full range of ε, i.e. $\varepsilon \geq 0$. In contrast, by using the direct way of computing the IRBF interpolant (i.e. directly solving (26)), Scheme 1 is limited to large values of ε.

Figure 13 displays the solution accuracy against the grid size by RBF-FD, RBF-HFD and Scheme 1 for several values of ε. It can be seen that, for large values of ε, i.e. 2.0 and 1.6, Scheme 1 is more accurate than RBF-FD and RBF-HFD. Decreasing ε can improve accuracy significantly. As shown in [25], the highest accuracy is often found for values of ε that cause the direct computation of the interpolant to suffer from severe ill-conditioning. Numerical experiments showed that the optimal value of ε is about 1.6 for RBF-FD ([10], Table 3), 0.85 for RBF-HFD ([10], Table 4) and 1.6 for Scheme 1. The first two are found over a full range of ε, while the last one is found over $\varepsilon-$values that make the direct way of computing the interpolant stable. RBF-HFD yields better accuracy than Scheme 1 for the optimal value of ε. It is expected that overcoming a severely ill-conditioned problem of Scheme 1 will extend its range of ε and thus improve its highest accuracy.

4.2.3 Example 3

Consider the following PDE

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 4(1-\pi^2)\sin(2\pi x)\sinh(2y) + 16(1-\pi^2)\cosh(4x)\cos(4\pi y) \qquad (43)$$

The problem domain is of circular shape of radius 1/2. Its centre is located at the origin. The exact solution is $u^{(e)}(x,y) = \sin(2\pi x)\sinh(2y) + \cosh(4x)\cos(4\pi y)$ from which the boundary conditions for u are easily derived. Figure 14 shows the variation of the exact solution over an extended domain $-1/2 \leq x, y \leq 1/2$. Both Scheme 1 and Scheme 2 are used. Calculations are carried out with grids of $(8\times 8, 10\times 10, 12\times 12, \cdots, 70\times 70)$ and $\beta = (20, 30, 40)$. Results obtained are presented in Figure 15, which plots the matrix

condition and the solution accuracy against the grid size for several values of β.

In the case of curved boundary, there may be interior points that are very close the boundary. Since their associated compact stencils employ the boundary nodes as the RBF centres, there are sudden changes in the distance between the nodal points in these stencils. They can be seen to be a random function of grid density. Figure 15 shows that the matrix condition behaviour has some fluctuation. For Scheme 1, the matrix condition grows as $O(h^{-2.85})$, $O(h^{-2.85})$ and $O(h^{-2.85})$ for the three values of β. For Scheme 2, it grows as $O(h^{-2.79})$, $O(h^{-2.79})$ and $O(h^{-2.79})$ for the three values of β. These rates are higher than those in the rectangular-domain case.

In spite of having some fluctuation in the matrix condition behaviour, the compact local IRBF solution still converges very fast and quite smoothly (Figure 15). Using $\beta = (20, 30, 40)$, the rates are $O(h^{4.03})$, $O(h^{4.02})$ $O(h^{4.02})$ for Scheme 1 and $O(h^{3.84})$, $O(h^{3.83})$ $O(h^{3.83})$ for Scheme 2.

Each scheme of the proposed method has thus similar performances in terms of the matrix condition and the solution accuracy for different values of β, which makes the choice of β easy in practice. Like in the rectangular case, Scheme 1 outperforms Scheme 2. The former is recommended for use.

Implementation notes:

Compact local 3-point and 9-point IRBF schemes lead to a system matrix that is sparse and yield a solution that is high-order accurate. These positive properties come at the expense of more computational effort required for the construction of compact stencils. Fortunately, compact stencils involve the handling of matrices and vectors of small sizes. In addition, some generic stencils can be built. For example, in Example 1 of PDEs, (30)-(33) need to be constructed only once and then used for every interior node.

In the 2D case, the present discretisation formulations are based on Cartesian grids. Numerical results indicate that high rates of convergence are obtained for both rectangular and non-rectangular domains. The present method is simpler in implementation (no fourth-order derivatives involved) but less flexible in discretisation (structured grids required) than the RBF-HFD method.

As shown in Figures 10 and 12, the way of incorporating the governing PDE into the approximations has a strong influence on the solution accuracy. Our various numerical studies of the number and location of additional nodes used for the PDE on the stencil indicate that Scheme 1 yields the most accurate results.

5 Conclusions

In this paper, a new compact local approximation method is proposed for the discretisation of second-order elliptic problems defined on rectangular and non-rectangular domains. The preprocessing is economical as Cartesian grids are used to represent the problem domain. The present IRBF approximations are constructed locally and each local construction has its own set of extra coefficients (a set of integration constants) which are exploited to impose the governing equation. They utilise more information about the governing equation than standard IRBF approximations, leading to a significant improvement in accuracy. This study further demonstrates the usefulness of the integration constants. The proposed method is validated successfully through a series of test problems in one and two dimensions. Very accurate results are obtained using relatively coarse grids.

Appendix

Analytic forms of the integrated MQ basis functions used are given below

$$H_i^{[x]}(\mathbf{x}) = \frac{(x - x_i^\dagger)}{2} Q + \frac{S^{[x]}}{2} R^{[x]} \tag{44}$$

$$H_i^{[y]}(\mathbf{x}) = \frac{(y - y_i^\dagger)}{2} Q + \frac{S^{[y]}}{2} R^{[y]} \tag{45}$$

$$\overline{H}_i^{[x]}(\mathbf{x}) = \left(\frac{(x - x_i^\dagger)^2}{6} - \frac{S^{[x]}}{3} \right) Q + \frac{S^{[x]}(x - x_i^\dagger)}{2} R^{[x]} \tag{46}$$

$$\overline{H}_i^{[y]}(\mathbf{x}) = \left(\frac{(y - y_i^\dagger)^2}{6} - \frac{S^{[y]}}{3} \right) Q + \frac{S^{[y]}(y - y_i^\dagger)}{2} R^{[y]} \tag{47}$$

where $\mathbf{x} = (x, y)^T$; $\mathbf{c}_i = (x_i^\dagger, y_i^\dagger)^T$; $r = \|\mathbf{x} - \mathbf{c}_i\|$;

$$Q = \sqrt{r^2 + a_i^2} \tag{48}$$

$$R^{[x]} = \ln\Big((x - x_i^\dagger) + Q \Big) \tag{49}$$

$$R^{[y]} = \ln\Big((y - y_i^\dagger) + Q \Big) \tag{50}$$

$$S^{[x]} = r^2 - (x - x_i^\dagger)^2 + a_i^2 \tag{51}$$

$$S^{[y]} = r^2 - (y - y_i^\dagger)^2 + a_i^2 \tag{52}$$

Acknowledgements

This work is supported by the Australian Research Council. The authors would like to thanks the referees for their helpful comments.

References

1. P.J. Roache, Fundamentals of Computational Fluid Dynamics, Hermosa Publishers, Albuquerque, 1998.

2. R.S. Hirsh, Higher order accurate difference solutions of fluid mechanics problems by a compact differencing technique, Journal of Computational Physics 19(1) (1975) 90-109.

3. S.K. Lele, Compact finite difference schemes with spectral-like resolution, Journal of Computational Physics 103(1) (1992) 16–42.

4. G.E. Fasshauer, Meshfree Approximation Methods With Matlab (Interdisciplinary Mathematical Sciences - Vol. 6), World Scientific Publishers, Singapore, 2007.

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

6. C.K. Lee, X. Liu, S.C. Fan, Local multiquadric approximation for solving boundary value problems, Computational Mechanics 30(5–6) (2003) 396–409.

7. C. Shu, H. Ding, K.S. Yeo, Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier-Stokes equations, Computer Methods in Applied Mechanics and Engineering 192 (2003) 941–954.

8. A.I. Tolstykh, D.A. Shirobokov, On using radial basis functions in a finite difference mode with applications to elasticity problems, Computational Mechanics 33(1) (2003) 68–79.

9. A.I. Tolstykh, D.A. Shirobokov, Using radial basis functions in a "finite difference mode", CMES: Computer Modeling in Engineering & Sciences 7(2) (2005) 207–222.

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

11. P. Orsini, H. Power, H. Morvan, Improving Volume Element Methods by Meshless Radial Basis Function Techniques, CMES: Computer Modeling in Engineering & Sciences 23(2) (2008) 187–208.

12. Y.V.S.S. Sanyasiraju, G. Chandhini, Local radial basis function based gridfree scheme for unsteady incompressible viscous flows, Journal of Computational Physics 227(20) (2008) 8922–8948.

13. R. Vertnik, B. Sarler, Solution of incompressible turbulent flow by a mesh-free method, CMES: Computer Modeling in Engineering & Sciences 44(1) (2009) 65–96.

14. A.H.-D. Cheng, M.A. Golberg, E.J. Kansa, G. Zammito, Exponential convergence and H-c multiquadric collocation method for partial differential equations, Numerical Methods for Partial Differential Equations 19 (2003) 571–594.

15. R. Schaback, Error estimates and condition numbers for radial basis function interpolation, Advances in Computational Mathematics 3 (1995) 251–264.

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

17. N. Mai-Duy, T. Tran-Cong, Approximation of function and its derivatives using radial basis function networks, Applied Mathematical Modelling 27 (2003) 197–220.

18. N. Mai-Duy, Solving high order ordinary differential equations with radial basis function networks, International Journal for Numerical Methods in Engineering 62 (2005) 824–852.

19. N. Mai-Duy, R.I. Tanner, Solving high order partial differential equations with indirect radial basis function networks, International Journal for Numerical Methods in Engineering 63 (2005) 1636–1654.

20. N. Mai-Duy, T. Tran-Cong, Solving biharmonic problems with scattered-point discretisation using indirect radial-basis-function networks, Engineering Analysis with Boundary Elements 30(2) (2006) 77–87.

21. N. Mai-Duy, R.I. Tanner, A collocation method based on one-dimensional RBF interpolation scheme for solving PDEs, International Journal of Numerical Methods

for Heat & Fluid Flow 17(2) (2007) 165–186.

22. N. Mai-Duy, T. Tran-Cong, A multidomain integrated radial basis function collocation method for elliptic problems, Numerical Methods for Partial Differential Equations 24 (2008) 1301–1320.

23. N. Mai-Duy, T. Tran-Cong, A Cartesian-grid discretisation scheme based on local integrated RBFNs for two-dimensional elliptic problems, CMES: Computer Modeling in Engineering & Sciences 51(3) (2009) 213–238.

24. D. Tavella, C. Randall, Pricing Financial Instruments: The Finite Difference Method, John Wiley & Sons, New York, 2000.

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

Table 1: ODE, Example 1, Dirichlet boundary conditions, $N = (5, 7, \cdots, 71)$: Condition numbers of the system and relative L_2 errors of the approximate solution u for various values of h by the compact 3-point finite-difference method and the compact 3-point IRBF method ($\beta = 40$).

	Compact 3-point FD			Compact 3-point IRBF		
h	cond(A)	$Ne(u)$	Rate	cond(A)	$Ne(u)$	Rate
2.50e-1	5.82e+0	2.73e-2	-	5.82e+0	2.72e-2	-
1.66e-1	1.39e+1	5.21e-3	4.08	1.39e+1	5.17e-3	4.09
1.25e-1	2.52e+1	1.62e-3	4.05	2.52e+1	1.60e-3	4.07
1.00e-1	3.98e+1	6.59e-4	4.03	3.98e+1	6.47e-4	4.06
8.33e-2	5.76e+1	3.16e-4	4.02	5.76e+1	3.07e-4	4.07
7.14e-2	7.87e+1	1.70e-4	4.01	7.87e+1	1.64e-4	4.08
6.25e-2	1.03e+2	9.96e-5	4.01	1.03e+2	9.48e-5	4.10
5.55e-2	1.30e+2	6.21e-5	4.01	1.30e+2	5.83e-5	4.12
5.00e-2	1.61e+2	4.07e-5	4.00	1.61e+2	3.76e-5	4.14
4.54e-2	1.95e+2	2.78e-5	4.00	1.95e+2	2.52e-5	4.19
4.16e-2	2.32e+2	1.96e-5	4.00	2.32e+2	1.74e-5	4.25
3.84e-2	2.73e+2	1.42e-5	4.00	2.73e+2	1.24e-5	4.25
3.57e-2	3.17e+2	1.05e-5	4.00	3.17e+2	9.02e-6	4.30
3.33e-2	3.64e+2	8.03e-6	4.00	3.64e+2	6.63e-6	4.46
3.12e-2	4.14e+2	6.20e-6	4.00	4.14e+2	4.98e-6	4.41
2.94e-2	4.67e+2	4.86e-6	4.00	4.67e+2	3.78e-6	4.53
2.77e-2	5.24e+2	3.87e-6	4.00	5.24e+2	2.90e-6	4.63
2.63e-2	5.84e+2	3.11e-6	4.00	5.84e+2	2.23e-6	4.88
2.50e-2	6.47e+2	2.53e-6	4.00	6.47e+2	1.76e-6	4.59
2.38e-2	7.14e+2	2.08e-6	4.00	7.14e+2	1.38e-6	4.97
2.27e-2	7.83e+2	1.73e-6	4.00	7.83e+2	1.08e-6	5.20
2.17e-2	8.56e+2	1.45e-6	4.00	8.56e+2	8.53e-7	5.44
2.08e-2	9.33e+2	1.22e-6	4.00	9.33e+2	6.71e-7	5.62
2.00e-2	1.01e+3	1.03e-6	4.00	1.01e+3	5.36e-7	5.50
1.92e-2	1.09e+3	8.88e-7	4.00	1.09e+3	4.21e-7	6.16
1.85e-2	1.18e+3	7.64e-7	4.00	1.18e+3	3.29e-7	6.52
1.78e-2	1.27e+3	6.60e-7	4.00	1.27e+3	2.59e-7	6.51
1.72e-2	1.36e+3	5.74e-7	4.00	1.36e+3	1.96e-7	7.96
1.66e-2	1.45e+3	5.01e-7	4.00	1.45e+3	1.43e-7	9.19
1.61e-2	1.55e+3	4.39e-7	4.00	1.55e+3	1.03e-7	10.05
1.56e-2	1.65e+3	3.87e-7	4.00	1.65e+3	6.92e-8	12.63
1.51e-2	1.76e+3	3.42e-7	4.00	1.76e+3	4.72e-8	12.44
1.47e-2	1.87e+3	3.03e-7	4.00	1.87e+3	2.54e-8	20.75
1.42e-2	1.98e+3	2.70e-7	4.00	1.98e+3	3.08e-9	72.76
	$O(h^{-2.02})$	$O(h^{4.01})$		$O(h^{-2.02})$	$O(h^{4.78})$	

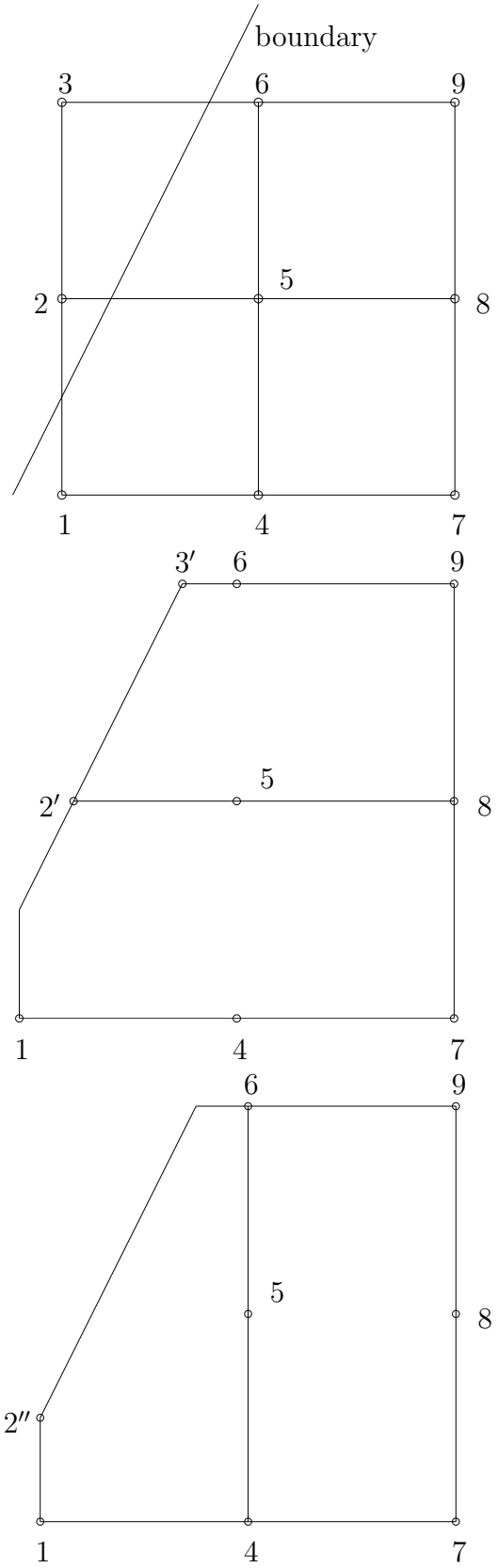

Figure 1: PDE, non-rectangular domain: a stencil cut by the boundary (top), RBF centres used for the x approximation (middle) and RBF centres used for the y approximation (bottom).

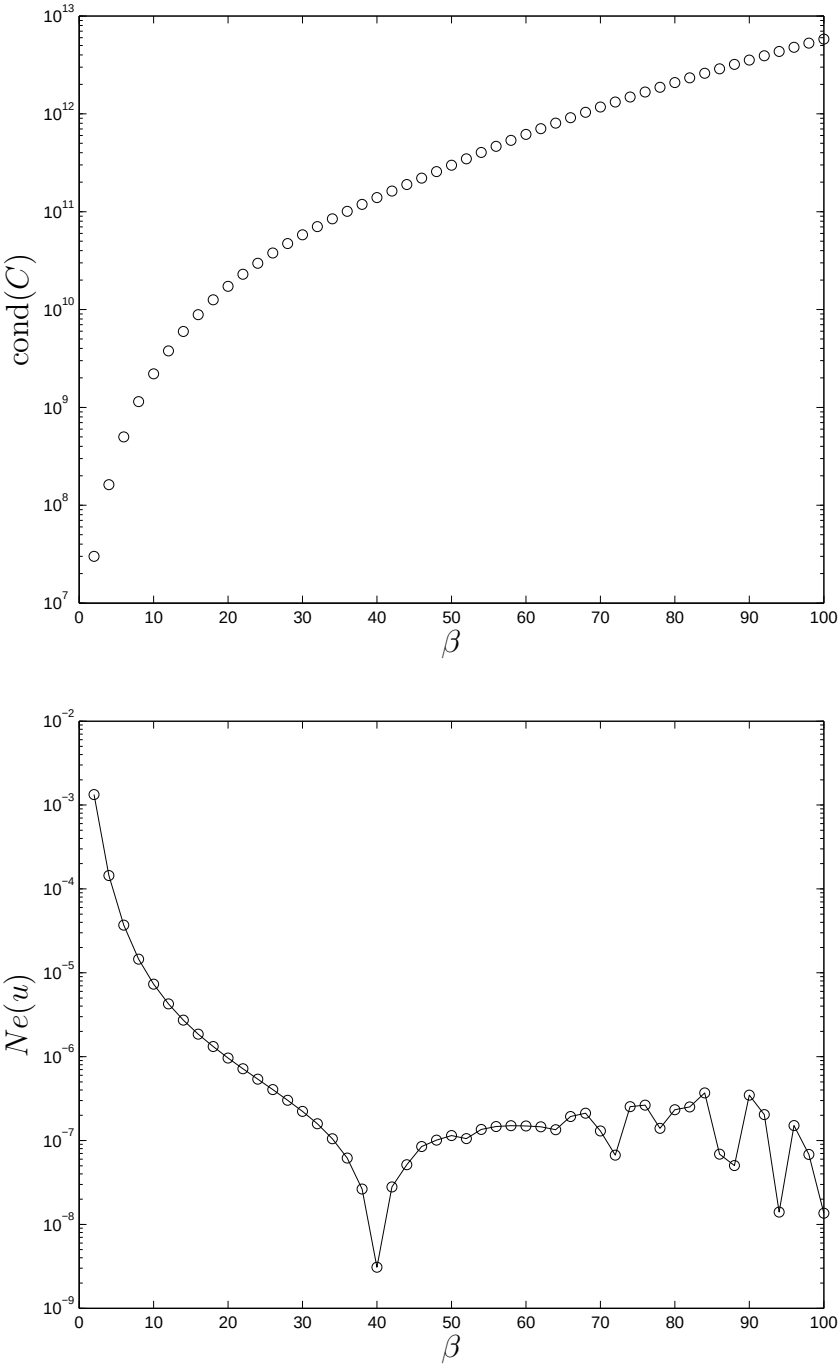

Figure 2: ODE, Example 1, Dirichlet boundary conditions, $N = 71$: $\beta-$adaptivity study.

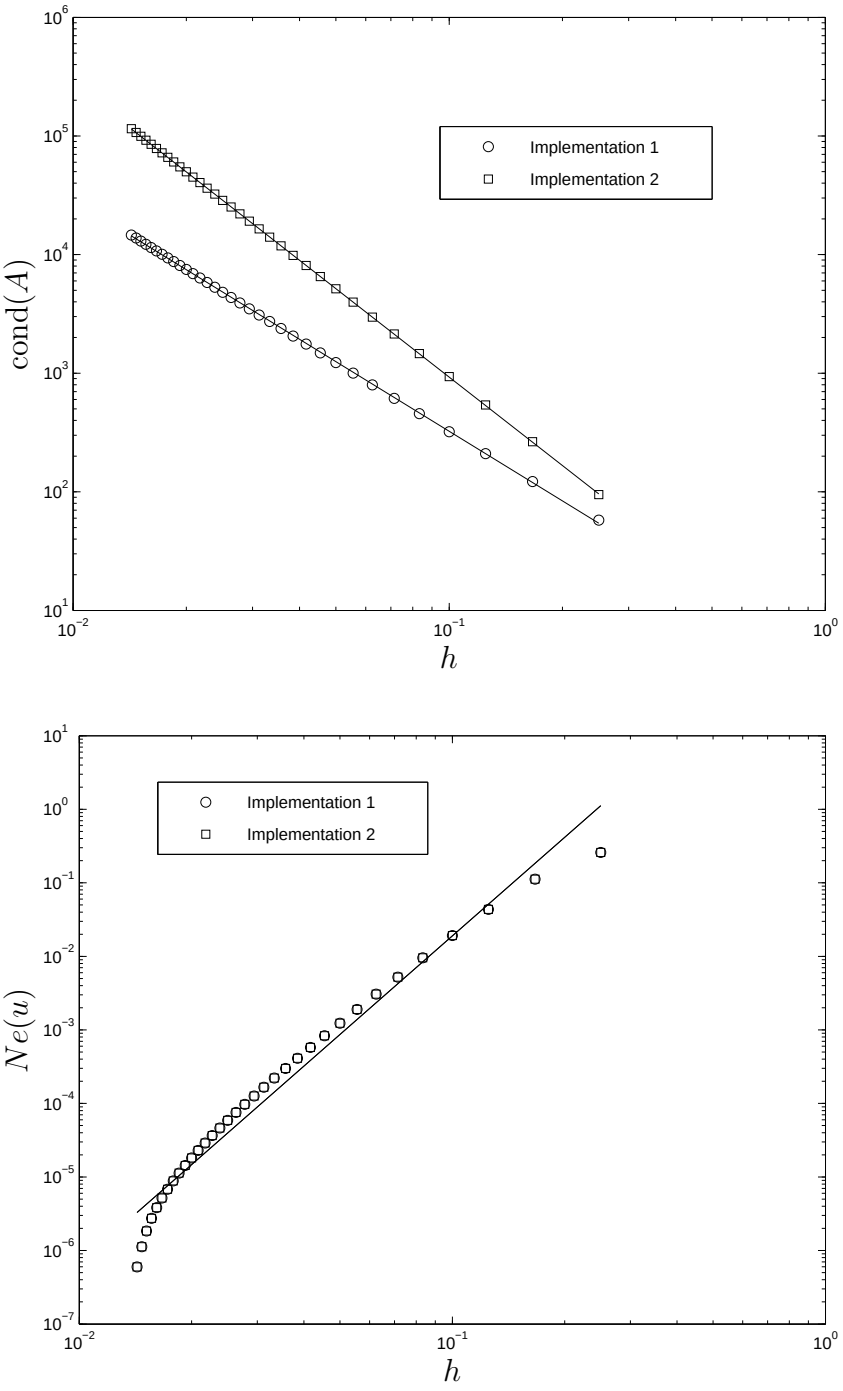

Figure 3: ODE, Example 1, $N = (5, 7, 9, \cdots, 71)$, $\beta = 25$: The Neumann boundary condition is imposed through the conversion process (Implementation 1) and through the final algebraic system (Implementation 2). Both implementation schemes have similar levels of accuracy which converge at $O(h^{4.45})$, but different condition numbers of the system matrix which grow at $O(h^{-1.94})$ for Implementation 1 and $O(h^{-2.47})$ for Implementation 2.

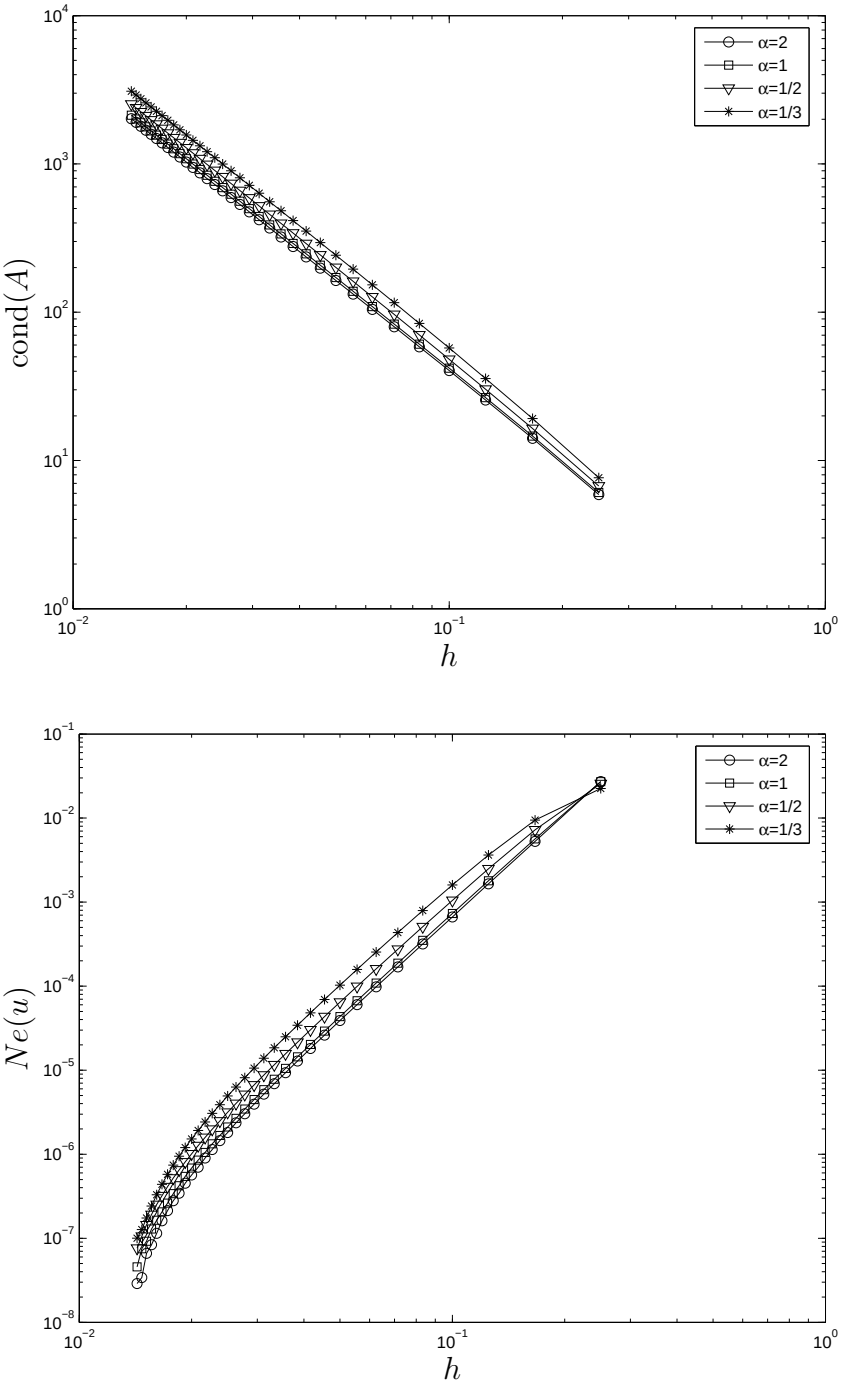

Figure 4: ODE, Example 1, Dirichlet boundary conditions, $N = (5, 7, 9, \cdots, 71)$: Performance of the present technique over several non-uniform grids. It is noted that the smaller the value of α the higher degree of non-uniformity the grid will be. For $\alpha = \{2, 1, 1/2, 1/3\}$, the matrix condition number grows as $\{O(h^{-2.02}), O(h^{-2.03}), O(h^{-2.05}), O(h^{-2.07})\}$, respectively and the solution converges as $\{O(h^{4.62}), O(h^{4.50}), O(h^{4.41}), O(h^{4.39})\}$, respectively.

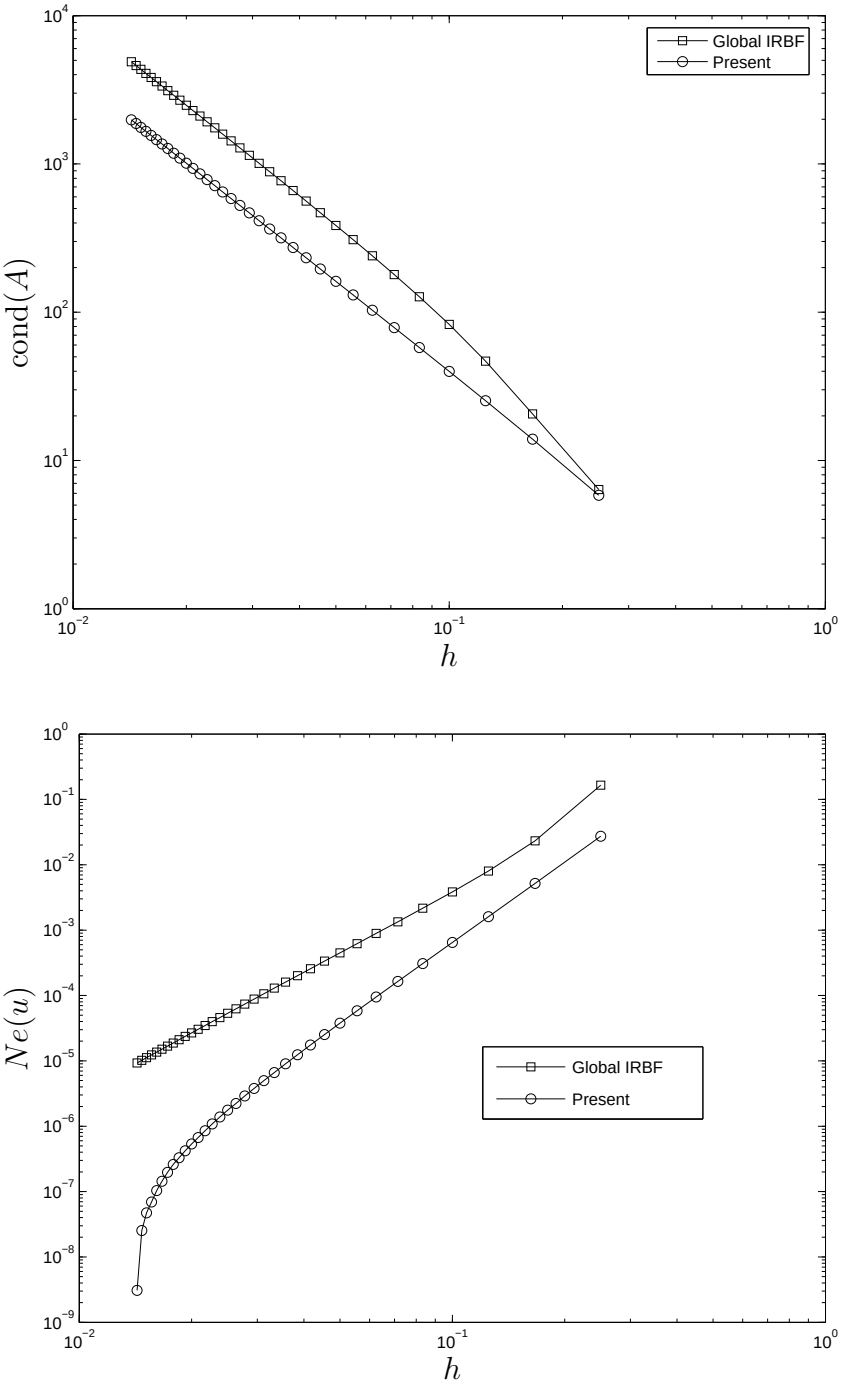

Figure 5: ODE, Example 1, Dirichlet boundary conditions, $N = (5, 7, 9, \cdots, 71)$: The matrix condition (top) and solution accuracy (bottom) against the grid size by the global IRBF method and the present method. The matrix condition grows as $O(h^{-2.02})$ for the present method and $O(h^{-2.19})$ for the global IRBF method while the solution converges as $O(h^{4.78})$ for the present method and $O(h^{3.21})$ for the global IRBF method.

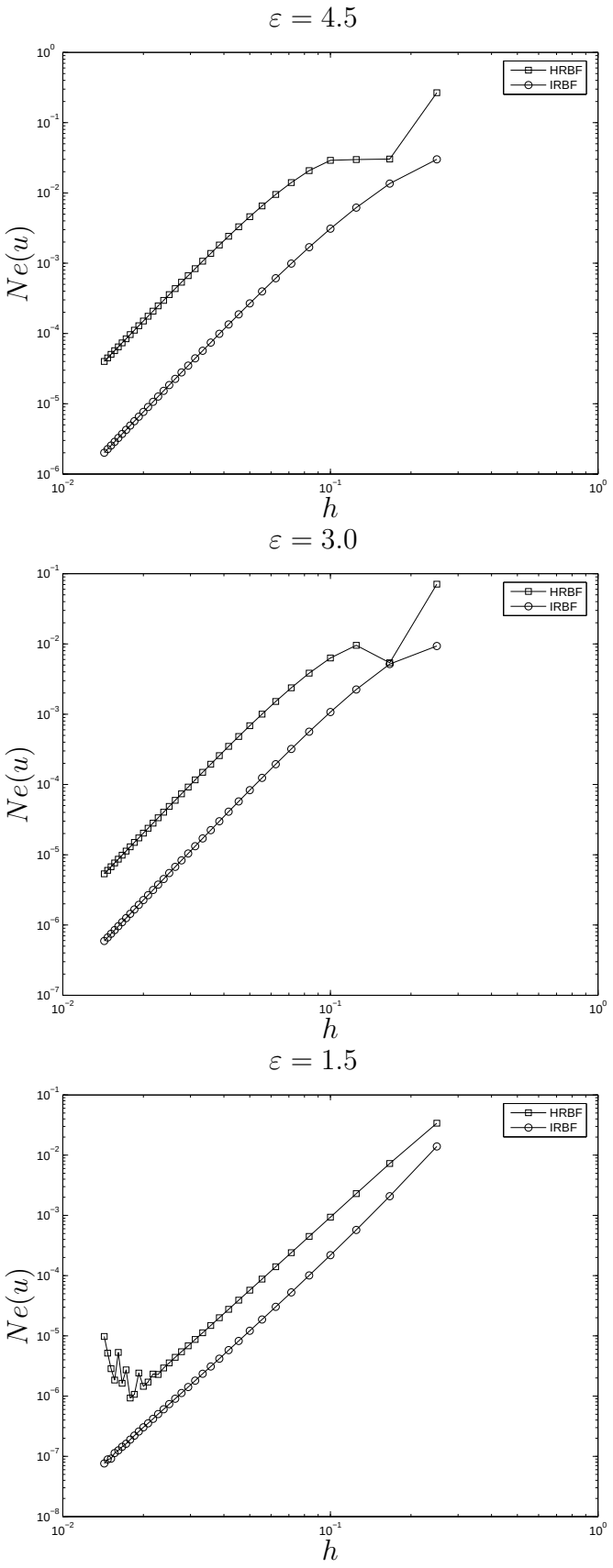

Figure 6: ODE, Example 1, Dirichlet boundary conditions, $N = (5, 7, 9, \cdots, 71)$: Errors by compact 3-point HRBF and IRBF schemes for several values of ε.

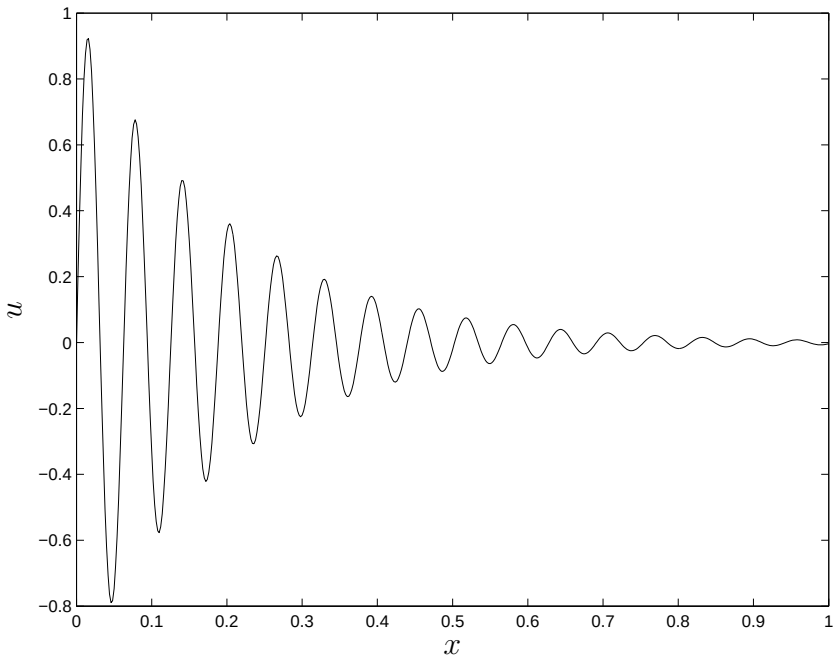

Figure 7: ODE, Example 2: exact solution.

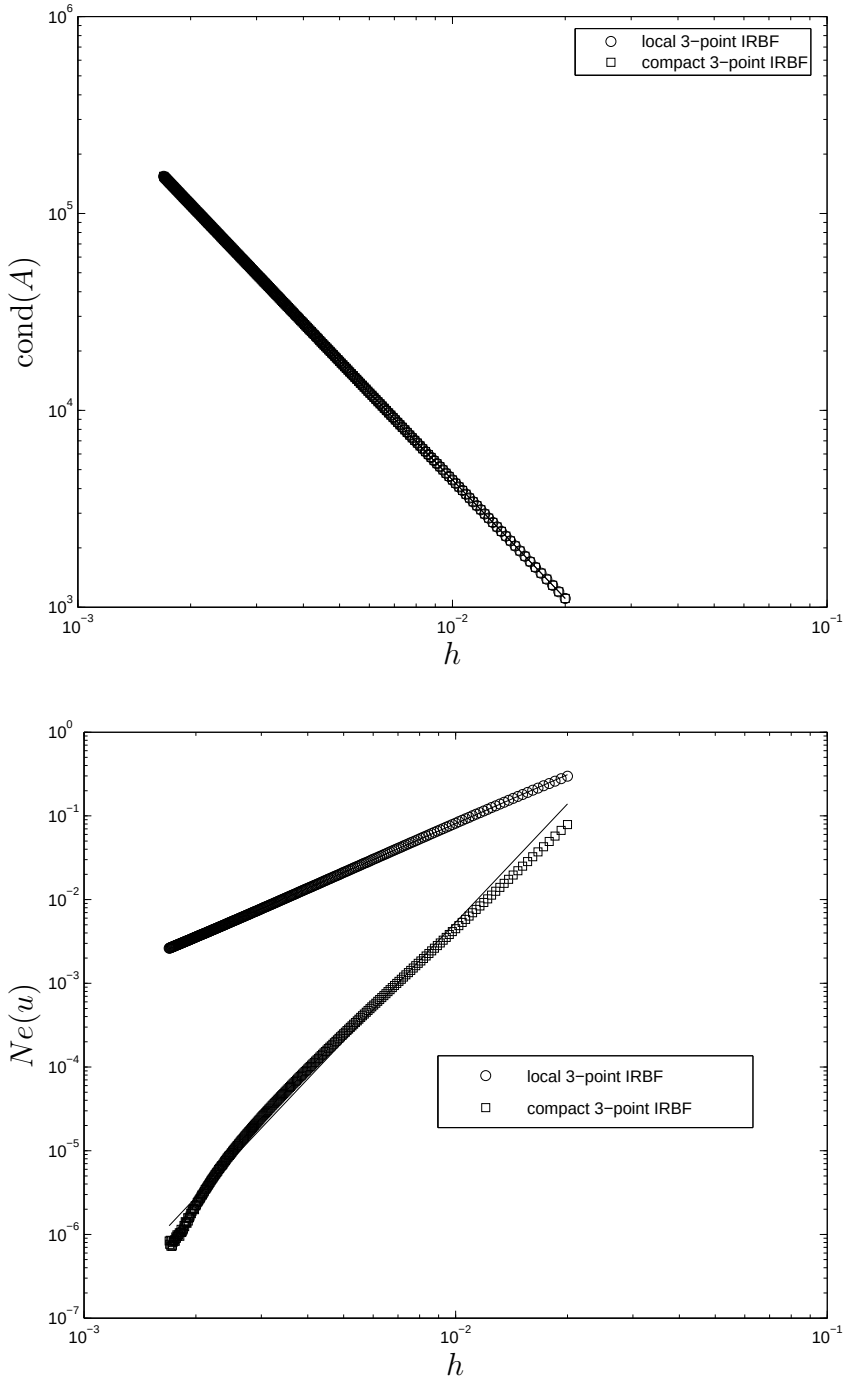

Figure 8: ODE, Example 2, Dirichlet boundary conditions, $N = (51, 53, 55, \cdots, 591)$, $\beta = 20$: $h-$adaptivity studies. The local 3-point IRBF method and the present compact 3-point IRBF schemes have similar condition numbers growing as $O(h^{-2.00})$ and their solutions converge apparently as $O(h^{1.94})$ and $O(h^{4.70})$, respectively.

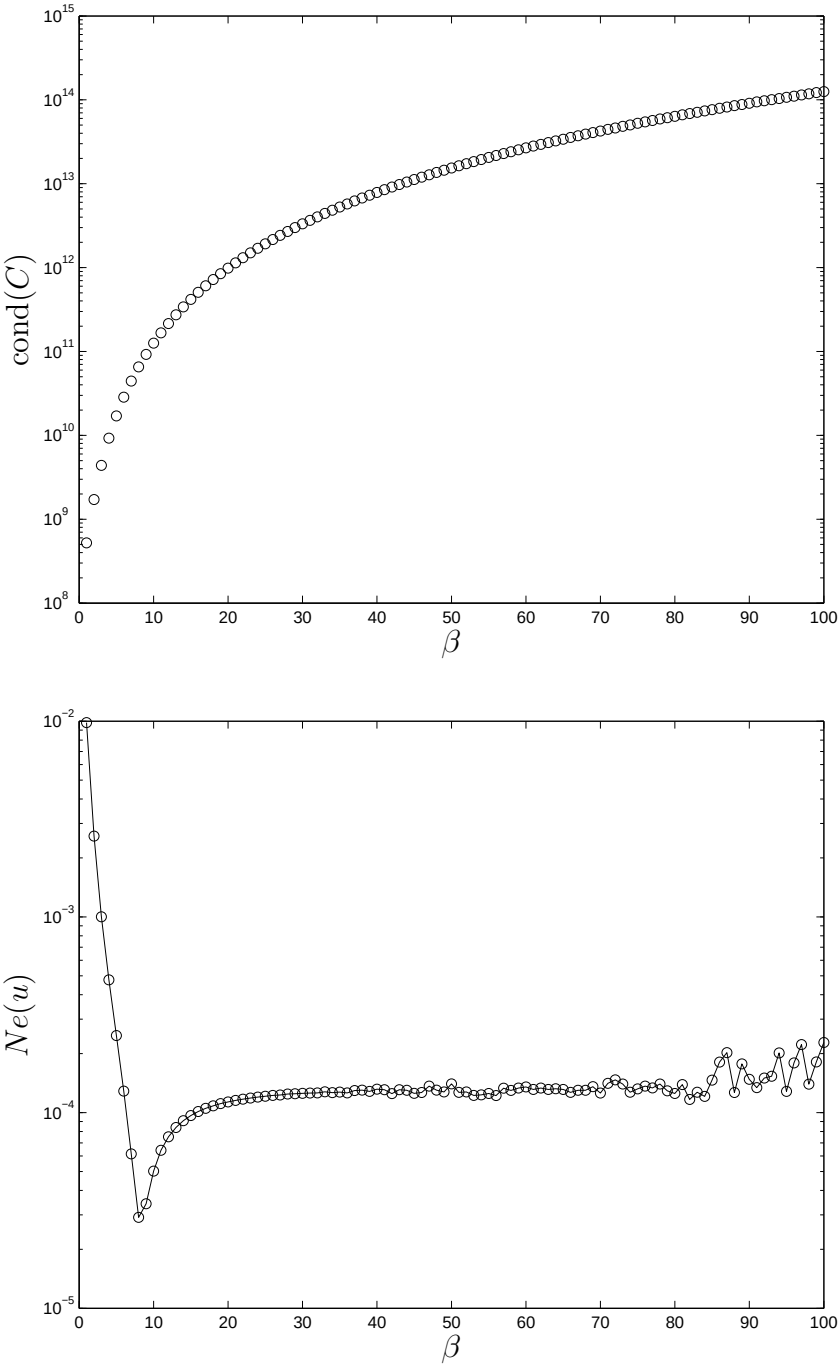

Figure 9: ODE, Example 2, Dirichlet boundary conditions, $N = 241$: $\beta-$adaptivity study.

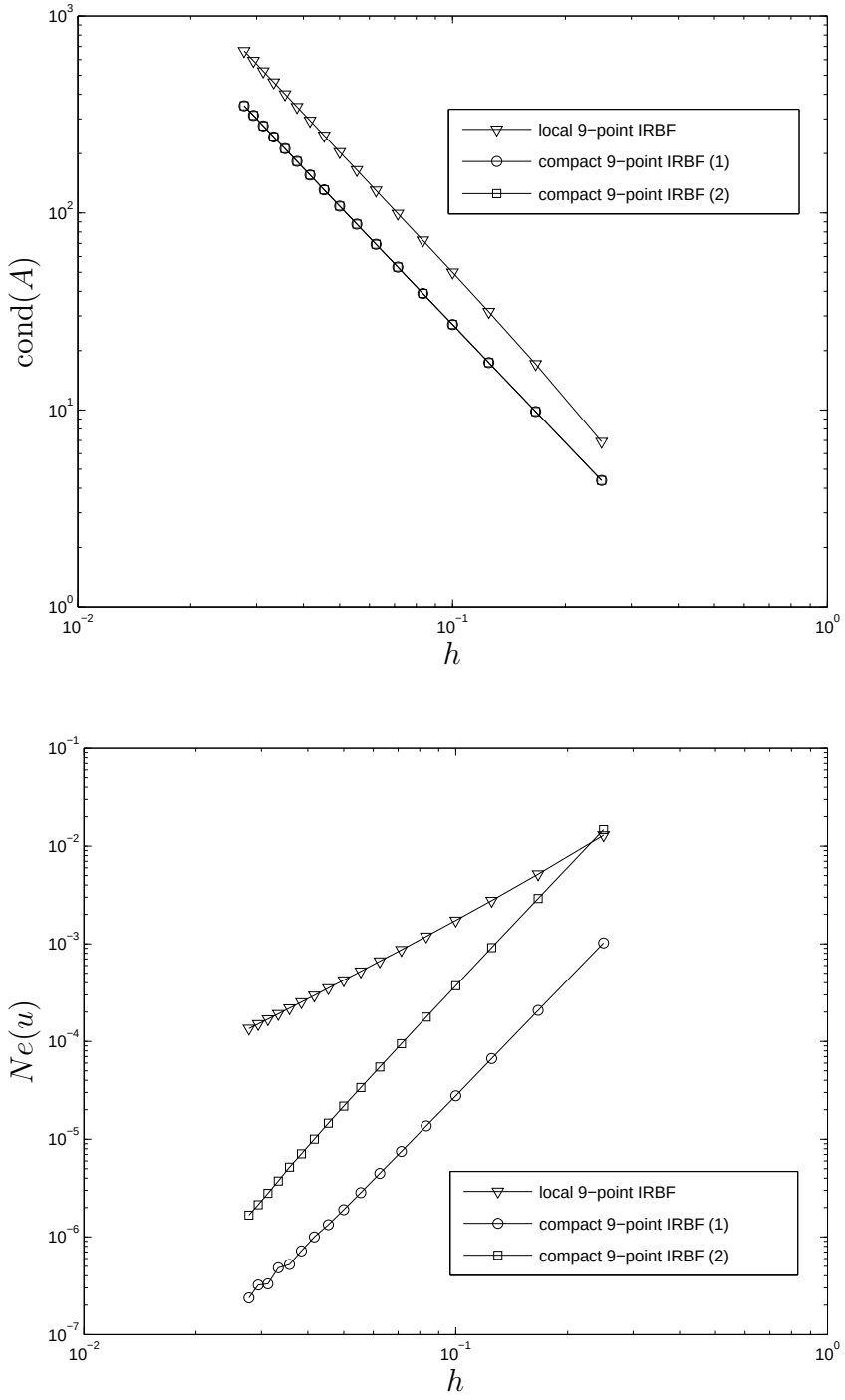

Figure 10: PDE, Example 1, rectangular domain, $(5 \times 5, 7 \times 7, \cdots, 37 \times 37)$, $\beta = 45$: $h-$adaptivity studies. For the local 9-point IRBF method, Scheme 1 and Scheme 2, the solution converges apparently as $O(h^{2.05})$, $O(h^{3.80})$ and $O(h^{4.13})$, while the matrix condition grows as $O(h^{-2.05})$, $O(h^{-1.99})$ and $O(h^{-1.99})$, respectively.

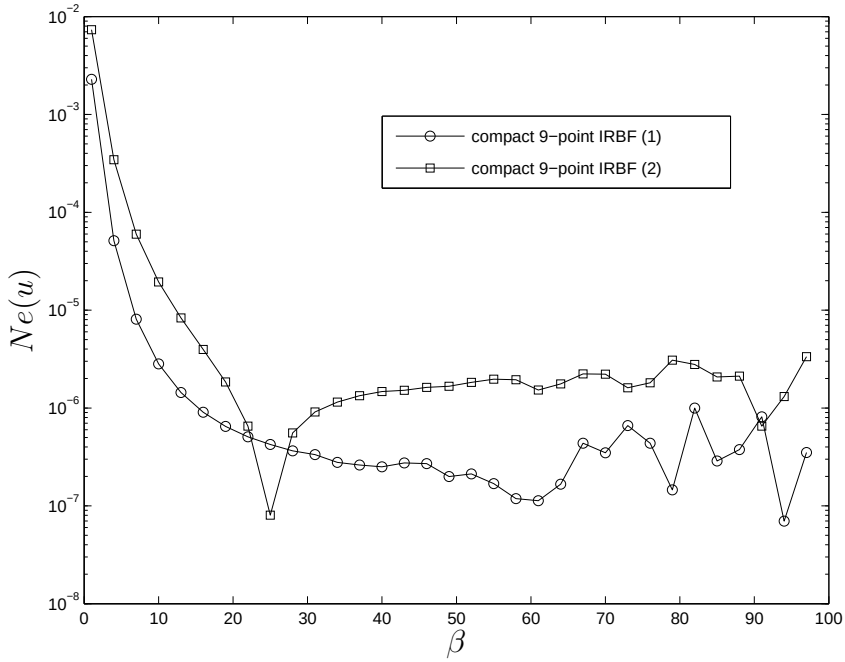

Figure 11: PDE, Example 1, rectangular domain, 37×37: $\beta-$adaptivity studies.

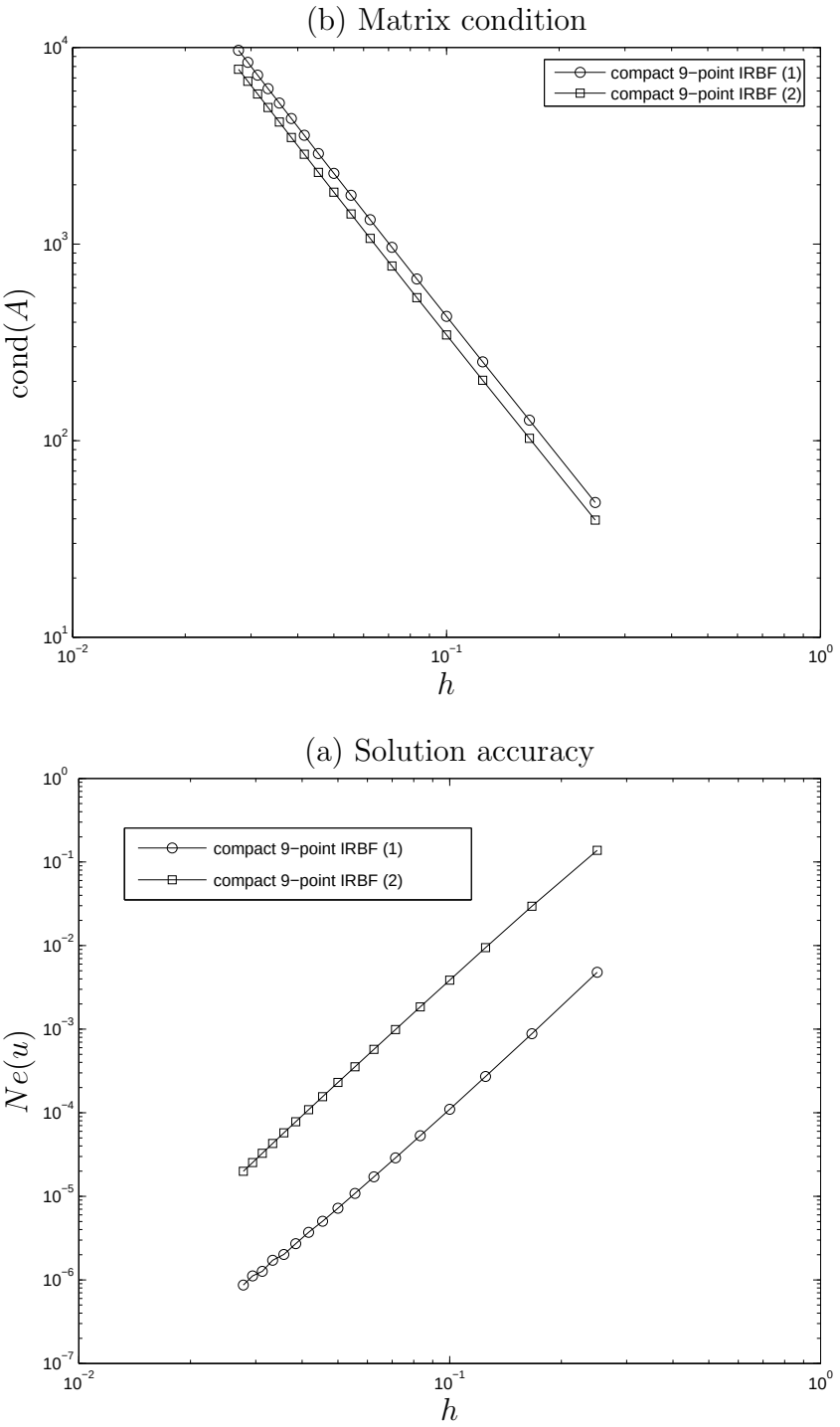

Figure 12: PDE, Example 1, rectangular domain, $(5\times 5, 7\times 7, \cdots, 37\times 37)$, $\beta = 45$: Plots of the matrix condition number (top) and accuracy (bottom) against the grid size for the case of Neumann boundary conditions. The solution converges apparently as $O(h^{3.90})$ for Scheme 1 and $O(h^{4.04})$ for Scheme 2, while the matrix condition grows as $O(h^{-2.41})$ for Scheme 1 and $O(h^{-2.40})$ for Scheme 2.

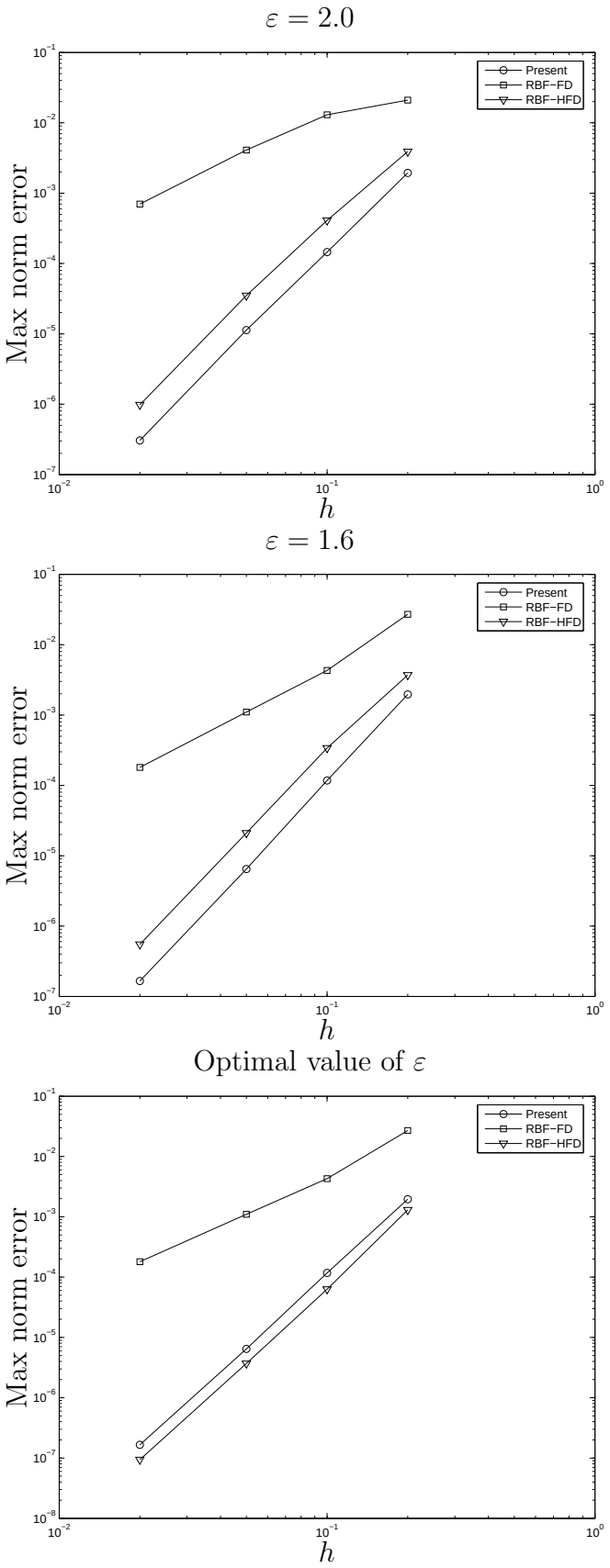

Figure 13: PDE, Example 2, rectangular domain: Errors by Scheme 1, RBF-FD and RBF-HFD for several values of ε. Results by RBF-FD and RBF-HFD are taken from [10].

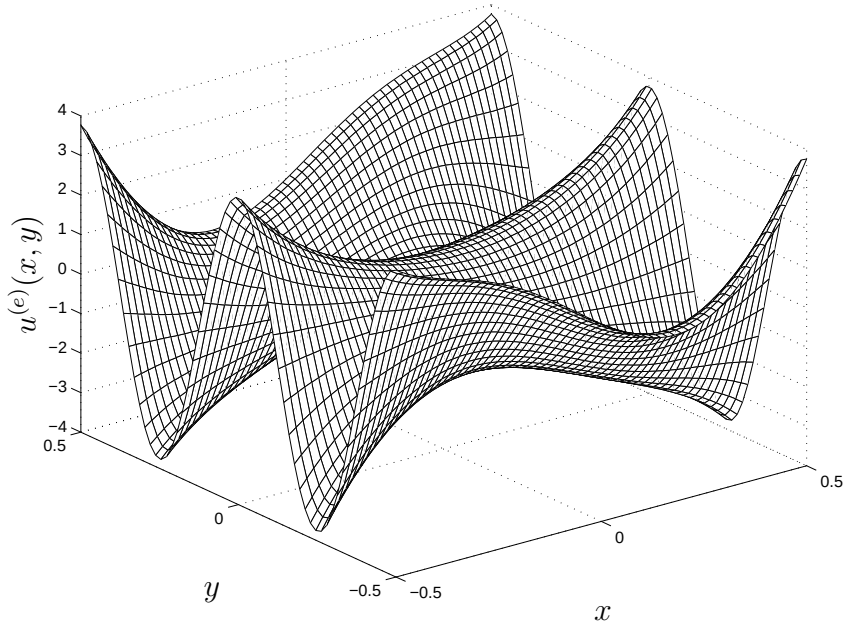

Figure 14: PDE, Example 3, circular domain: Exact solution over an extended domain.

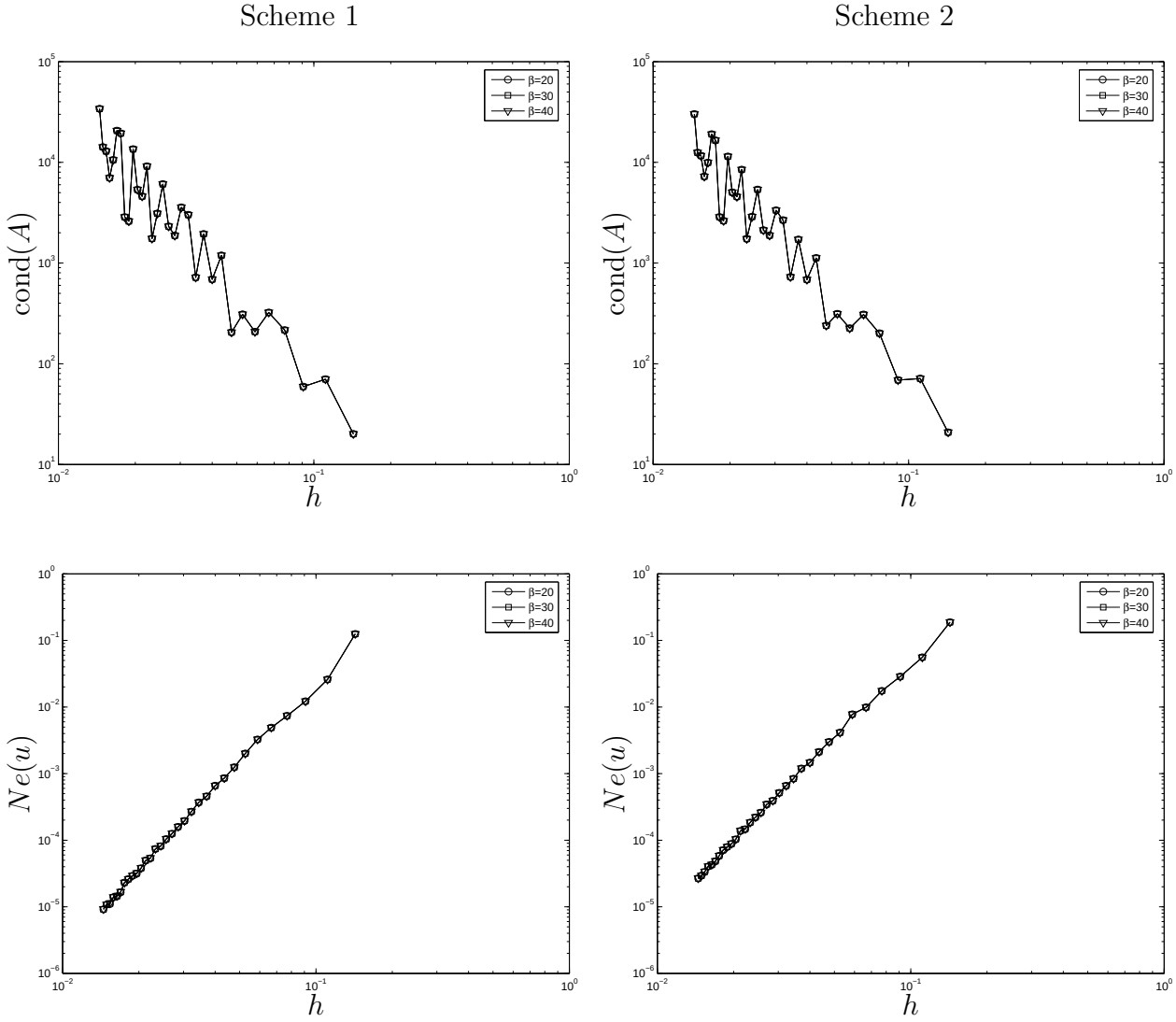

Figure 15: PDE, Example 3, circular domain, $(8 \times 8, 10 \times 10, \cdots, 70 \times 70)$: The matrix condition (top) and solution accuracy (bottom) against the grid size for $\beta = (20, 30, 40)$ by Scheme 1 (left) and Scheme 2 (right). Plots have the same scaling. Consider $\beta = 20$. The solution converges as $O(h^{4.03})$ for Scheme 1 and $O(h^{3.84})$ for Scheme 2, while the matrix condition grows as $O(h^{-2.85})$ for Scheme 1 and $O(h^{-2.79})$ for Scheme 2.