A multidomain integrated-radial-basis-function collocation method for elliptic problems

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

Faculty of Engineering and Surveying,

The University of Southern Queensland, Toowoomba, QLD 4350, Australia

Submitted to *Numerical Methods for Partial Differential Equations*,
11-Jul-2007; revised, 11-Oct-2007

[*] Corresponding author: Telephone +61 7 4631 1324, Fax +61 7 4631 2526, E-mail maiduy@usq.edu.au

Abstract This paper is concerned with the use of integrated radial-basis-function networks (IRBFNs) and non-overlapping domain decompositions (DDs) for numerically solving one- and two-dimensional elliptic problems. A substructuring technique is adopted, where subproblems are discretized by means of one-dimensional IRBFNs. A distinguishing feature of the present DD technique is that the continuity of the RBF solution across the interfaces is enforced with one order higher than with conventional DD techniques. Several test problems governed by second- and fourth-order differential equations are considered to investigate the accuracy of the proposed technique.

KEY WORDS: non-overlapping domain decomposition; radial basis function; collocation technique; high-order differential equations

1 Introduction

The basic idea of a physical DD technique (cf. [1]) is to divide the problem domain into a number of subdomains, on which the governing differential equations (DEs) are solved with transmission conditions at the interfaces. The main advantage of DD techniques lies in their ability to handle large-scale problems. Given a spatial discretization, the size of matrices obtained with DD techniques is much smaller than that with a one-domain technique. With the recent emergence of parallel computers, DD techniques have become more attractive because they allow the parallel computations of solutions to subdomains. The disadvantage of DD techniques is that their solution is not as smooth as that on a single domain.

The basic part of any DD technique lies in the method of matching the computed solutions on contiguous regions. In the context of non-overlapping decomposition, the continuity conditions of the field variable and its normal derivative up to a certain order are enforced across the subdomain interfaces. For second-order problems, the enforcement is applied to the solution and its first-order normal derivative, while for fourth-order problems, one imposes the continuity for the solution and its normal derivatives of order up to three. The DD approach thus provides an approximate solution that is a C^1 function across the interfaces for second-order problems and C^3 function for fourth-order problems. It is noted that there are relatively few papers on DDs for the solution of fourth-order differential problems.

RBF collocation methods for solving DEs appeared in the early 1990s. Kansa [2] and Fasshauer [3] were the first to suggest the so-called non-symmetric and symmetric RBF collocation methods, respectively. The RBF methods are easy to implement and have the capability to provide a very accurate solution using relatively low numbers of points. When handling large-scale problems, like other discretization techniques, the RBF methods need to be combined with DD techniques for an efficient solution or one needs to construct the RBF approximations locally. A number of RBF papers discussing these topics have been reported, see, e.g. [4-8] for the use of DDs and [9-11] for the use of local approximations. A drawback of the DD approach is that it requires a domain partition, i.e. some sort of meshing. For these one- and multi-domain RBF collocation techniques, the construction of the RBF approximations is based on differentiation (DRBFNs).

In this paper, we discuss a multidomain IRBFN collocation technique for the solution of second- and fourth-order elliptic problems in one and two dimensions. The IRBFN discretization scheme, which is applied on subdomains here, is based on the use of a Cartesian grid and a one-dimensional (1D) IRBFN interpolation scheme that has recently been reported in [12,13]. The incorporation of 1D-IRBFNs into the substructuring technique leads to an approximate solution that is a C^p function, instead of the usual C^{p-1}, across the subdomain interfaces, where p is the order of the DE. It is expected that this achievement of higher-order smoothness of the approximate solution alleviates the deterioration in accuracy caused by the domain division.

An outline of the paper is as follows. Brief reviews of 1D-IRBFNs for the approximation of functions and the solution of DEs are given in Sections 2 and 3, respectively. The proposed DD technique is described in Section 4. Numerical results are presented in Section 5. Section 6 concludes the paper.

2 A brief review of 1D-IRBFNs

Consider a univariate function $f(x)$. The basic idea of the integral RBF scheme [14] is to decompose a pth-order derivative of the function f into RBFs

$$\frac{d^p f(x)}{dx^p} = \sum_{i=1}^{n} w_i \varphi_i(x) = \sum_{i=1}^{n} w_i I_i^{(p)}(x), \tag{1}$$

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

$$\frac{d^{p-1} f(x)}{dx^{p-1}} = \sum_{i=1}^{n} w_i I_i^{(p-1)}(x) + c_1, \tag{2}$$

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

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

$$\frac{df(x)}{dx} = \sum_{i=1}^{n} w_i I_i^{(1)}(x) + c_1 \frac{x^{p-2}}{(p-2)!} + c_2 \frac{x^{p-3}}{(p-3)!} + \cdots + c_{p-2} x + c_{p-1}, \tag{4}$$

$$f(x) = \sum_{i=1}^{n} w_i I_i^{(0)}(x) + c_1 \frac{x^{p-1}}{(p-1)!} + c_2 \frac{x^{p-2}}{(p-2)!} + \cdots + c_{p-1} x + c_p, \tag{5}$$

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

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

The evaluation of (1)-(5) at a set of collocation points $\{x_j\}_{j=1}^n$ leads to

$$\frac{\widehat{d^p f}}{dx^p} = \widehat{\mathcal{I}}_{[p]}^{(p)} \widehat{\alpha}, \tag{6}$$

$$\frac{\widehat{d^{p-1} f}}{dx^{p-1}} = \widehat{\mathcal{I}}_{[p]}^{(p-1)} \widehat{\alpha}, \tag{7}$$

$$\cdots\cdots\cdots$$

$$\frac{\widehat{df}}{dx} = \widehat{\mathcal{I}}_{[p]}^{(1)} \widehat{\alpha}, \tag{8}$$

$$\widehat{f} = \widehat{\mathcal{I}}_{[p]}^{(0)} \widehat{\alpha}, \tag{9}$$

where the subscript [.] and superscript (.) are used to denote the order of the IRBFN scheme and the order of a derivative function, respectively;

$$\widehat{\mathcal{I}}_{[p]}^{(p)} = \begin{bmatrix} I_1^{(p)}(x_1), & I_2^{(p)}(x_1), & \cdots, & I_n^{(p)}(x_1), & 0, & 0, & \cdots, & 0, & 0 \\ I_1^{(p)}(x_2), & I_2^{(p)}(x_2), & \cdots, & I_n^{(p)}(x_2), & 0, & 0, & \cdots, & 0, & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ I_1^{(p)}(x_n), & I_2^{(p)}(x_n), & \cdots, & I_n^{(p)}(x_n), & 0, & 0, & \cdots, & 0, & 0 \end{bmatrix},$$

$$\widehat{\mathcal{I}}_{[p]}^{(p-1)} = \begin{bmatrix} I_1^{(p-1)}(x_1), & I_2^{(p-1)}(x_1), & \cdots, & I_n^{(p-1)}(x_1), & 1, & 0, & \cdots, & 0, & 0 \\ I_1^{(p-1)}(x_2), & I_2^{(p-1)}(x_2), & \cdots, & I_n^{(p-1)}(x_2), & 1, & 0, & \cdots, & 0, & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ I_1^{(p-1)}(x_n), & I_2^{(p-1)}(x_n), & \cdots, & I_n^{(p-1)}(x_n), & 1, & 0, & \cdots, & 0, & 0 \end{bmatrix},$$

$\cdots\cdots,$

$$\widehat{\mathcal{I}}^{(0)}_{[p]} = \left[\begin{array}{ccccccccc} I_1^{(0)}(x_1), & I_2^{(0)}(x_1), & \cdots, & I_n^{(0)}(x_1), & \frac{x_1^{p-1}}{(p-1)!}, & \frac{x_1^{p-2}}{(p-2)!}, & \cdots, & x_1, & 1 \\ I_1^{(0)}(x_2), & I_2^{(0)}(x_2), & \cdots, & I_n^{(0)}(x_2), & \frac{x_2^{p-1}}{(p-1)!}, & \frac{x_2^{p-2}}{(p-2)!}, & \cdots, & x_2, & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ I_1^{(0)}(x_n), & I_2^{(0)}(x_n), & \cdots, & I_n^{(0)}(x_n), & \frac{x_n^{p-1}}{(p-1)!}, & \frac{x_n^{p-2}}{(p-2)!}, & \cdots, & x_n, & 1 \end{array} \right];$$

$$\widehat{\alpha} = (w_1, w_2, \cdots, w_n, c_1, c_2, \cdots, c_p)^T;$$

and

$$\widehat{\frac{d^k f}{dx^k}} = \left(\frac{d^k f_1}{dx^k}, \frac{d^k f_2}{dx^k}, \cdots, \frac{d^k f_n}{dx^k} \right)^T, \quad k = (1, 2, \cdots, p),$$
$$\widehat{f} = (f_1, f_2, \cdots, f_n)^T,$$

in which $d^k f_j/dx^k = d^k f(x_j)/dx^k$ and $f_j = f(x_j)$ with $j = (1, 2, \cdots, n)$.

The use of integrated basis functions is expected to avoid the problem of reduction of convergence rate caused by differentiation [15]. Numerical studies (e.g. [16,17]) have shown that the integral collocation approach is more accurate than the differential one. Recently, theoretical studies [18] have confirmed superior accuracy of IRBFNs over DRBFNs. Moreover, there are additional weights (integration constants) in the integral collocation formulation, and they have been found to be extremely useful for handling the multiple boundary conditions [19-21]. This study further exploits the constants of integration for the purpose of improving the order of continuity of the approximate RBF solution across the subdomain interfaces.

3 A brief review of 1D-IRBFNs for solving DEs on a single domain

In the remainder of the paper, we will use

- the notation $\widehat{[]}$ for a vector/matrix $[]$ that is associated with a grid line,
- the notation $\widetilde{[]}$ for a vector/matrix $[]$ that is associated with the whole set of grid lines,
- the notation $\overline{[]}$ for a vector/matrix $[]$ that is associated with the boundaries of the domain,
- the notation $[]_{(\eta,\theta)}$ to denote selected rows η and columns θ of the matrix $[]$,
- the notation $[]_{(\eta)}$ to pick out selected components η of the vector $[]$,
- the notation $[]_{(:,\theta)}$ to denote all rows of the matrix $[]$, and
- the notation $[]_{(\eta,:)}$ to denote all columns of the matrix $[]$.

3.1 1D elliptic problems

Consider a 1D boundary-value problem governed by the pth-order ODE

$$F(u,\frac{du}{dx},\frac{d^2u}{dx^2},\cdots,\frac{d^pu}{dx^p}) = b(x), \quad r \leq x \leq s, \tag{10}$$

where F and b are prescribed functions, together with boundary conditions for u, du/dx, ..., and $d^{p/2-1}u/dx^{p/2-1}$ at $x = r$ and $x = s$.

The continuous domain of interest is replaced by a set of discrete points $\{x_j\}_{j=1}^{n}$ with $x_1 = r$ and $x_n = s$. The integral scheme of order p (IRBFN-p) is employed here to approximate the field variable and its derivatives in the ODE and the boundary conditions. Owing to the presence of p integration constants in the integral formulation, one can add p extra equations to the discrete system. These extra equations can be utilized to represent the ODE and the values of the derivative boundary conditions at both ends of the domain. The governing equation (10) and the boundary conditions can be transformed into the following discrete form

$$\widehat{\mathcal{A}}\,\widehat{\alpha} = \widehat{f}, \tag{11}$$

where $\widehat{\mathcal{A}}$ is the system matrix of size $(n+p)\times(n+p)$ defined as

$$\widehat{\mathcal{A}} = \left[\begin{array}{c} F\left(\widehat{\mathcal{I}}^{(0)}_{[p](1,:)}, \widehat{\mathcal{I}}^{(1)}_{[p](1,:)}, \widehat{\mathcal{I}}^{(2)}_{[p](1,:)}, \cdots, \widehat{\mathcal{I}}^{(p)}_{[p](1,:)}\right) \\ F\left(\widehat{\mathcal{I}}^{(0)}_{[p](2,:)}, \widehat{\mathcal{I}}^{(1)}_{[p](2,:)}, \widehat{\mathcal{I}}^{(2)}_{[p](2,:)}, \cdots, \widehat{\mathcal{I}}^{(p)}_{[p](2,:)}\right) \\ \cdots \\ F\left(\widehat{\mathcal{I}}^{(0)}_{[p](n,:)}, \widehat{\mathcal{I}}^{(1)}_{[p](n,:)}, \widehat{\mathcal{I}}^{(2)}_{[p](n,:)}, \cdots, \widehat{\mathcal{I}}^{(p)}_{[p](n,:)}\right) \\ \widehat{\mathcal{I}}^{(0)}_{[p]([1,n],:)} \\ \widehat{\mathcal{I}}^{(1)}_{[p]([1,n],:)} \\ \cdots \\ \widehat{\mathcal{I}}^{(p/2-1)}_{[p]([1,n],:)} \end{array}\right],$$

$$\widehat{\alpha} = (w_1, w_2, \cdots, w_n, c_1, c_2, \cdots, c_p)^T, \quad \text{and}$$

$$\widehat{f} = (b_1, b_2, \cdots, b_n, u_r, u_s, \frac{du_r}{dx}, \frac{du_s}{dx}, \cdots, \frac{d^{p/2-1}u_r}{dx^{p/2-1}}, \frac{d^{p/2-1}u_s}{dx^{p/2-1}})^T.$$

In (11), the ODE is collocated at the whole set of grid points including the two boundary points $x = r$ and $x = s$. Solving (11) yields

$$\widehat{\alpha} = \widehat{\mathcal{A}}^{-1}\widehat{f}, \tag{12}$$

from which one is able to obtain the values of u and its derivatives at the grid points via (6)-(9). More details can be found in [19].

3.2 2D elliptic problems

Consider a 2D boundary value problem governed by the pth-order PDE

$$F(u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \cdots, \frac{\partial^k u}{\partial x^i \partial y^j}, \cdots, \frac{\partial^p u}{\partial x^p}, \frac{\partial^p u}{\partial y^p}) = b(x, y), \quad (x, y) \in \Omega, \tag{13}$$

where Ω is a rectangular domain, and subject to the prescribed conditions for u, $\partial u/\partial n$, ..., and $\partial^{p/2-1}u/\partial n^{p/2-1}$ on the boundaries of Ω ($n-$the direction normal to the boundary).

The 1D-IRBFN-based Cartesian-grid technique approximates the solution in terms of nodal variable values rather than network weights. On a grid line, the pth-order integral scheme (IRBFN-p) is employed. Along a horizontal grid line, the relationships between the network-weight space and the physical space can be described by

$$\begin{pmatrix} \widehat{u} \\ \widehat{v} \end{pmatrix} = \widehat{\mathcal{C}}_{[p]} \, \widehat{\alpha}, \tag{14}$$

where $\widehat{\mathcal{C}}_{[p]}$ is the conversion matrix of dimension $(n_x + p) \times (n_x + p)$

$$\widehat{\mathcal{C}}_{[p]} = \begin{bmatrix} \widehat{\mathcal{I}}^{(0)}_{[p]} \\ \widehat{\mathcal{I}}^{(1)}_{[p]([1,n_x],:)} \\ \cdots \\ \widehat{\mathcal{I}}^{(p/2-1)}_{[p]([1,n_x],:)} \\ \widehat{\mathcal{I}}^{(p)}_{[p]([1,n_x],:)} \end{bmatrix},$$

$$\widehat{u} = (u_1, u_2, \cdots, u_{n_x})^T,$$

$$\widehat{v} = \left(\frac{\partial u_1}{\partial x}, \frac{\partial u_{n_x}}{\partial x}, \cdots, \frac{\partial^{p/2-1} u_1}{\partial x^{p/2-1}}, \frac{\partial^{p/2-1} u_{n_x}}{\partial x^{p/2-1}}, \frac{\partial^p u_1}{\partial x^p}, \frac{\partial^p u_{n_x}}{\partial x^p} \right)^T,$$

$$\widehat{\alpha} = (w_1, w_2, \cdots, w_{n_x}, c_1, c_2, \cdots, c_p)^T,$$

and n_x is the number of grid points on the line. It can be seen that (14) takes into account information about the values of u at the grid points (the first n_x equations), the derivative boundary conditions (the next $(p-2)$ equations) and the PDE at the boundary points (the last two equations). Solving (14), one will obtain a map from the physical space to the network-weight space

$$\widehat{\alpha} = \widehat{\mathcal{C}}_{[p]}^{-1} \begin{pmatrix} \widehat{u} \\ \widehat{v} \end{pmatrix}. \tag{15}$$

Substitution of (15) into (6)-(8) yields

$$\widehat{\frac{\partial^k u}{\partial x^k}} = \widehat{\mathcal{I}}_{[p]}^{(k)} \widehat{\mathcal{C}}_{[p]}^{-1} \begin{pmatrix} \widehat{u} \\ \widehat{v} \end{pmatrix}, \quad k = (1, 2, \cdots, p), \tag{16}$$

where $\widehat{\frac{\partial^k u}{\partial x^k}} = \left(\frac{\partial^k u_1}{\partial x^k}, \frac{\partial^k u_2}{\partial x^k}, \cdots, \frac{\partial^k u_{n_x}}{\partial x^k} \right)^T$. Approximate expressions for derivatives of u with respect to x over the whole domain can then be conveniently constructed by means of Kronecker tensor products. The process of constructing the 1D-IRBFN approximations for $\partial^k u/\partial y^k$ is similar to that for $\partial^k u/\partial x^k$.

Moreover, mixed derivatives of u can be computed via the following relation

$$\frac{\partial^k u}{\partial x^i \partial y^j} = \frac{1}{2} \left[\frac{\partial^i}{\partial x^i} \left(\frac{\partial^j u}{\partial y^j} \right) + \frac{\partial^j}{\partial y^j} \left(\frac{\partial^i u}{\partial x^i} \right) \right], \quad \text{with} \quad k = i + j, \tag{17}$$

which reduces the computation of mixed derivatives to that of lower-order pure derivatives for which IRBFNs involve integration with respect to x or y only. In computing (17), lower-order integral schemes are employed, and hence one only needs to take derivative boundary conditions (not information about the PDE at the boundary points) as extra information ($\widehat{v}$), making the computation significantly simpler.

Let n, n_{ip} and n_{bp} be the total number of collocation points, the number of interior points and the number of boundary points, respectively. Collocating (13) at the interior points results in

$$\widetilde{\mathcal{A}} \left(\begin{array}{c} \widetilde{u} \\ \overline{\frac{\partial u}{\partial n}} \\ \cdots \\ \overline{\frac{\partial^{p/2-1} u}{\partial n^{p/2-1}}} \\ \overline{\frac{\partial^p u}{\partial n^p}} \end{array} \right) = \widetilde{b}, \tag{18}$$

where $\widetilde{\mathcal{A}}$ is a known matrix of size $n_{ip} \times (n + (p/2)n_{bp})$; $\widetilde{b}$ represents the values of b in (13) at the interior points; $\widetilde{u} = (u_1, u_2, \cdots, u_n)^T$; and $\overline{\frac{\partial u}{\partial n}}, \cdots, \overline{\frac{\partial^{p/2-1} u}{\partial n^{p/2-1}}}$ and $\overline{\frac{\partial^p u}{\partial n^p}}$ represent the values of $\frac{\partial u}{\partial n}, \cdots, \frac{\partial^{p/2-1} u}{\partial n^{p/2-1}}$ and $\frac{\partial^p u}{\partial n^p}$ at the boundary points, respectively.

The equation set (18) can be rewritten as

$$\widetilde{\mathcal{A}}_{(:,ip)}\, \widetilde{u}_{(ip)} = \widetilde{b} - \overline{\mathcal{B}}_0 \overline{u} - \overline{\mathcal{B}}_1 \overline{\frac{\partial u}{\partial n}} + \cdots - \overline{\mathcal{B}}_{p/2-1} \overline{\frac{\partial^{p/2-1} u}{\partial n^{p/2-1}}} - \overline{\mathcal{B}}_p \overline{\frac{\partial^p u}{\partial n^p}}, \tag{19}$$

where

$$\overline{u} = \widetilde{u}_{(bp)},\ \overline{\mathcal{B}}_0 = \widetilde{\mathcal{A}}_{(:,bp)},$$
$$\overline{\mathcal{B}}_1 = \widetilde{\mathcal{A}}_{(:,n+bp)}, \cdots,\ \overline{\mathcal{B}}_{p/2-1} = \widetilde{\mathcal{A}}_{(:,n+n_{bp}(p/2-2)+bp)},\ \overline{\mathcal{B}}_p = \widetilde{\mathcal{A}}_{(:,n+n_{bp}(p/2-1)+bp)},$$

and the notations ip and bp refer to the indices of the rows/columns that are associated with the interior and boundary points, respectively. Given the PDE and the boundary conditions, the right-hand side of (19) can reduce to a known vector. In forming (19), the IRBFN discretization does not involve the four corners of the domain.

More details can be found in [12]. In the case of irregular domains, there are some modifications required and they were reported in [13].

4 The proposed multidomain(MD) IRBFN method

The present DD method is based on the use of non-overlapping subdomains and 1D-IRBFNs. The substructuring technique is employed to construct a separated problem that involves the unknowns relative to the subdomain interfaces only (the Schur complement system). Subdomains are handled here with the 1D-IRBFN-based Cartesian-grid method.

4.1 1D elliptic equations

For simplicity, the present DD technique is described for the case of the biharmonic equation and 2 non-overlapping subdomains, namely I and II. The values of u and du/dx at the interface $\breve{x}$ are selected to be the interface unknowns

$$u_I(\breve{x}) = u_{II}(\breve{x}) = \breve{u}, \tag{20}$$

$$\frac{du_I}{dx}(\breve{x}) = \frac{du_{II}}{dx}(\breve{x}) = \frac{d\breve{u}}{dx}, \tag{21}$$

and these unknowns are then determined by solving the equations of continuity in second- and third-order derivatives

$$\frac{d^2u_I}{dx^2}(\breve{x}) = \frac{d^2u_{II}}{dx^2}(\breve{x}), \tag{22}$$

$$\frac{d^3u_I}{dx^3}(\breve{x}) = \frac{d^3u_{II}}{dx^3}(\breve{x}). \tag{23}$$

Expressions for d^2u/dx^2 and d^3u/dx^3 in the Schur complement system, (22) and (23), are constructed using the subdomain solver ((10)-(12))

$$\frac{d^2u}{dx^2}(\breve{x}) = \widehat{\mathcal{I}}^{(2)}_{[4](n,:)}\widehat{\alpha} = \widehat{\mathcal{I}}^{(2)}_{[4](n,:)}\widehat{\mathcal{A}}^{-1}\widehat{f}, \tag{24}$$

$$\frac{d^3u}{dx^3}(\breve{x}) = \widehat{\mathcal{I}}^{(3)}_{[4](n,:)}\widehat{\alpha} = \widehat{\mathcal{I}}^{(3)}_{[4](n,:)}\widehat{\mathcal{A}}^{-1}\widehat{f}, \tag{25}$$

in which $\widehat{f} = \left(b_1, b_2, \cdots, b_n, u_r, \breve{u}, \frac{du_r}{dx}, \frac{d\breve{u}}{dx}\right)^T$ for subdomain I, and

$$\frac{d^2u}{dx^2}(\breve{x}) = \widehat{\mathcal{I}}^{(2)}_{[4](1,:)}\widehat{\alpha} = \widehat{\mathcal{I}}^{(2)}_{[4](1,:)}\widehat{\mathcal{A}}^{-1}\widehat{f}, \tag{26}$$

$$\frac{d^3u}{dx^3}(\breve{x}) = \widehat{\mathcal{I}}^{(3)}_{[4](1,:)}\widehat{\alpha} = \widehat{\mathcal{I}}^{(3)}_{[4](1,:)}\widehat{\mathcal{A}}^{-1}\widehat{f}, \tag{27}$$

in which $\widehat{f} = \left(b_1, b_2, \cdots, b_n, \breve{u}, u_s, \frac{d\breve{u}}{dx}, \frac{du_s}{dx}\right)^T$ for subdomain II. In (24)-(27), the subscripts I and II are dropped out for brevity.

Substitution of (24)-(27) into (22)-(23) leads to a set of two algebraic equations for the two unknowns $\breve{u}$ and $\frac{d\breve{u}}{dx}$. Once these unknowns are determined, the solutions to subdomains will be obtained through (12) and (6)-(9).

A distinguishing feature of the present DD scheme is that, in solving a subproblem, the ODE is forced to be satisfied exactly at the interface (the nth and 1st rows in (11) for subdomains I and II, respectively):

$$\frac{d^4u_I}{dx^4}(\breve{x}) + \frac{d^2u_I}{dx^2}(\breve{x}) = b(\breve{x}) \quad \text{and} \tag{28}$$

$$\frac{d^4u_{II}}{dx^4}(\breve{x}) + \frac{d^2u_{II}}{dx^2}(\breve{x}) = b(\breve{x}). \tag{29}$$

Since the field variable u and its first three derivatives are enforced to be continuous at the interface ((20)-(23)), equations (28) and (29) lead to

$$\frac{d^4u_I}{dx^4}(\breve{x}) = \frac{d^4u_{II}}{dx^4}(\breve{x}). \tag{30}$$

The present MD-IRBFN technique thus achieves a C^4 solution at the interface.

4.2 2D elliptic equations

The MD-IRBFN method in the previous section is extended to the case of two-dimensional problems. The problem domain, which can be of regular and irregular shape, is partitioned into a number of subdomains, for which the interfaces are required to run parallel to the x and y axes and the grid points on the interface of two adjoining subdomains are chosen to be the same. The MD-IRBFN method is described in detail for the Poisson and biharmonic equations with Dirichlet boundary conditions. Special attention is given to the treatment for the continuity of the approximate solution at the interior corner points.

4.2.1 Poisson equation

Consider a typical subdomain (i.e. all boundaries are the interfaces) (Figure 1). The proposed DD technique gives different treatments for the interior corner points (the intersection points) and the interior points on the interfaces (the interface points), which achieves a C^2 solution across the interfaces. The present Schur complement system is constructed as follows. The unknown values on the interfaces are chosen to be the values of u at the interface points and the values of u, $\partial^2 u/\partial x^2$ and $\partial^2 u/\partial y^2$ at the intersection points. The equations used for determining these unknowns are based on the continuity of $\partial u/\partial n$ at the interface points, and the continuity of $\partial u/\partial x$, $\partial u/\partial y$ together with the satisfaction of the PDE at the intersection points.

Interface points: The values of $\partial u/\partial n$ at these nodes are computed through the subproblem solver discussed earlier, which involves two steps.

First, one needs to express the values of u at the interior points of a subdomain in terms of the interface unknowns only. For the Poisson equation, the discrete system (19) reduces to

$$\widetilde{\mathcal{A}}_{(:,ip)}\ \widetilde{u}_{(ip)} = \widetilde{b} - \overline{\mathcal{B}}_0\overline{u} - \overline{\mathcal{B}}_2\overline{\frac{\partial^2 u}{\partial n^2}}. \tag{31}$$

The second-order normal derivative vector on the right-hand side of (31) can be replaced by

$$\overline{\frac{\partial^2 u}{\partial n^2}} = \overline{b} - \overline{\frac{\partial^2 u}{\partial t^2}}, \tag{32}$$

where t is the direction tangent to the boundary.

We employs the second-order integral scheme (IRBFN-2) to express tangent derivatives in (32) in terms of the interface unknowns. For an interface of the subdomain, one has

$$\widehat{\frac{\partial^2 u}{\partial t^2}} = \widehat{\mathcal{I}}^{(2)}_{[2]}\widehat{\mathcal{C}}^{-1}_{[2]}\begin{pmatrix} \widehat{u} \\ \widehat{v} \end{pmatrix}, \tag{33}$$

where $\widehat{u}$ represents the values of u at the grid points on the interface, and $\widehat{v}$ the values of $\partial^2 u/\partial t^2$ at the two extreme points of the interface (i.e. the intersection points).

Second, by taking (31), (32) and (33) into account, one is able to derive the values of $\partial u/\partial n$ at the interface points in terms of the interface unknowns only.

Intersection points: The values of $\partial u/\partial x$ and $\partial u/\partial y$ at these nodes are simply computed via function approximation.

Applying the IRBFN-2 scheme to the interfaces that run parallel to the x axis, one obtains the values of $\partial u/\partial x$ at the intersection points

$$\begin{pmatrix} \frac{\partial u_1}{\partial x} \\ \frac{\partial u_{n_x}}{\partial x} \end{pmatrix} = \widehat{\mathcal{I}}^{(1)}_{[2]([1,n_x],:)} \widehat{\mathcal{C}}^{-1}_{[2]} \begin{pmatrix} \widehat{u} \\ \frac{\partial^2 u_1}{\partial x^2} \\ \frac{\partial^2 u_{n_x}}{\partial x^2} \end{pmatrix}, \tag{34}$$

where $\widehat{u}$ represents the values of u at the grid points on the interface.

Similarly, for the interfaces that run parallel to the y axis, one has

$$\begin{pmatrix} \frac{\partial u_1}{\partial y} \\ \frac{\partial u_{n_y}}{\partial x} \end{pmatrix} = \widehat{\mathcal{I}}^{(1)}_{[2]([1,n_y],:)} \widehat{\mathcal{C}}^{-1}_{[2]} \begin{pmatrix} \widehat{u} \\ \frac{\partial^2 u_1}{\partial y^2} \\ \frac{\partial^2 u_{n_y}}{\partial y^2} \end{pmatrix}. \tag{35}$$

The computations here are relatively simple because the approximations used involve information about the interfaces only.

The third equation in a set of three equations, which is employed at each intersection point, is the PDE in its original form, i.e.

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = b. \tag{36}$$

Owing to the fact that the PDE is not collocated at the four corners of the domain in the solution of subdomains, equation (36) is independent from continuity equations for $\partial u/\partial n$.

Continuity order: The present DD technique enforces the continuity of the solution across the interfaces with one order higher than conventional DD techniques because (i) the PDE is forced to be satisfied at the interface points in the subdomain solutions, and (ii) the four subdomains associated with an intersection point are forced to have the same values of $\partial^2 u/\partial x^2$ and $\partial^2 u/\partial y^2$ at that point.

4.2.2 Biharmonic equation

The MD-IRBFN method for the biharmonic equation is similar to that for the Poisson equation. However, the Schur complement system is larger. The unknown vector consists of the values of u and $\partial u/\partial n$ at the interface points, and the values of u, $\partial u/\partial x$, $\partial u/\partial y$, $\partial^4 u/\partial x^4$ and $\partial^4 u/\partial y^4$ at the intersection points. The construction of the interface system is based on the use of a set of two equations, namely $\partial^2 u/\partial n^2$ and $\partial^3 u/\partial n^3$, at an interface point, and a set of 5 equations, $\partial^2 u/\partial x^2$, $\partial^2 u/\partial y^2$, $\partial^3 u/\partial x^3$, $\partial^3 u/\partial y^3$ and the PDE, at an intersection point.

The values of $\partial^2 u/\partial n^2$ and $\partial^3 u/\partial n^3$ at an interface point are computed using the subdomain solver and the IRBFN-4 scheme. The equations that correspond to (31), (32) and (33) are, respectively,

$$\widetilde{\mathcal{A}}_{(:,ip)}\;\widetilde{u}_{(ip)} = \widetilde{b} - \overline{\mathcal{B}}_0\overline{u} - \overline{\mathcal{B}}_1\overline{\frac{\partial u}{\partial n}} - \overline{\mathcal{B}}_4\overline{\frac{\partial^4 u}{\partial n^4}}, \tag{37}$$

$$\overline{\frac{\partial^4 u}{\partial n^4}} = \overline{b} - 2\overline{\frac{\partial^4 u}{\partial x^2\partial y^2}} - \overline{\frac{\partial^4 u}{\partial t^4}}, \tag{38}$$

$$\widehat{\frac{\partial^4 u}{\partial t^4}} = \widehat{\mathcal{I}}^{(4)}_{[4]}\widehat{\mathcal{C}}^{-1}_{[4]}\begin{pmatrix}\widehat{u}\\ \\ \widehat{v}\end{pmatrix}, \tag{39}$$

where $\widehat{v}$ consists of the values of $\partial u/\partial t$ and $\partial^4 u/\partial t^4$ at the intersection points. After solving (37), any derivative functions of u, including the mixed fourth-order one, can be expressed in terms of the interface unknowns only.

On the other hand, the computation of $\partial^2 u/\partial x^2$, $\partial^2 u/\partial y^2$, $\partial^3 u/\partial x^3$ and $\partial^3 u/\partial y^3$ at an intersection point is based on the use of the fourth-order 1D-IRBFN scheme only. For a horizontal interface, the values of $\partial^2 u/\partial x^2$ and $\partial^3 u/\partial x^3$ at the two extreme points of the interface are given below

$$\begin{pmatrix} \frac{\partial^2 u_1}{\partial x^2} \\ \frac{\partial^2 u_{n_x}}{\partial x^2} \end{pmatrix} = \widehat{\mathcal{I}}^{(2)}_{[4]([1,n_x],:)} \widehat{\mathcal{C}}^{-1}_{[4]} \begin{pmatrix} \widehat{u} \\ \frac{\partial u_1}{\partial x} \\ \frac{\partial u_{n_x}}{\partial x} \\ \frac{\partial^4 u_1}{\partial x^4} \\ \frac{\partial^4 u_{n_x}}{\partial x^4} \end{pmatrix} \quad \text{and} \tag{40}$$

$$\begin{pmatrix} \frac{\partial^3 u_1}{\partial x^3} \\ \frac{\partial^3 u_{n_x}}{\partial x^3} \end{pmatrix} = \widehat{\mathcal{I}}^{(3)}_{[4]([1,n_x],:)} \widehat{\mathcal{C}}^{-1}_{[4]} \begin{pmatrix} \widehat{u} \\ \frac{\partial u_1}{\partial x} \\ \frac{\partial u_{n_x}}{\partial x} \\ \frac{\partial^4 u_1}{\partial x^4} \\ \frac{\partial^4 u_{n_x}}{\partial x^4} \end{pmatrix}, \tag{41}$$

where $\widehat{u}$ is the vector containing the values of u at the grid points on the interface.

Continuity order: Consider an interface (Γ) and its two associated subdomains (Ω_1 and Ω_2). As shown earlier, the present DD technique imposes the continuity for the solution u and its normal derivatives of order up to four (i.e. $\partial u/\partial n$, $\partial^2 u/\partial n^2$, $\partial^3 u/\partial n^3$ and $\partial^4 u/\partial n^4$) at every point on the interface Γ. The 1D-IRBFN approximations for derivatives of these interface values with respect to the tangent direction (i.e. mixed derivatives) from subdomain Ω_1 can be seen to be identical to those taken from subdomain Ω_2. In other words, one also has the continuity across the interface Γ for mixed derivatives (e.g. $\partial^4 u/\partial t^2 \partial n^2$). The present DD technique achieves a C^4 solution across the interface Γ.

5 Numerical examples

Several examples are presented here to demonstrate the attractiveness of the proposed DD technique.

It has generally been accepted that, among RBFNs, the multiquadric (MQ)-based interpolation scheme tends to result in the most accurate results. The present 1D-IRBFN schemes are implemented with the MQ function whose form is

$$\varphi_i(x) = \sqrt{(x - c_i)^2 + a_i^2}, \tag{42}$$

where c_i and a_i are the centre and the width of the ith MQ function, respectively. The centre points are selected to coincide with the collocation points. The MQ widths are known to have a profound effect on the performance of MQ-RBFNs. However, there is still a lack of mathematical theories for specifying their optimal values. For all numerical examples taken here, the MQ widths are simply computed by

$$a_i = \beta d_i, \quad i = (1, 2, \cdots, n), \tag{43}$$

in which β is a factor and d_i is the distance between the the ith centre and its closest neighbour. The reader is referred to [16,19] for a discussion about the effect of β on accuracy of the RBFN solutions.

To assess the accuracy of the MD-IRBFN method, the one-domain IRBFN and MD-DRBFN methods are considered. It is noted that the present MD-DRBFN method is also based on the substructuring technique and the multiquadric functions. Both multidomain RBF methods employ the same formulation of interface conditions.

5.1 1D second-order problem

Consider the following second-order ODE

$$\frac{d^2u}{dx^2} + \frac{du}{dx} + u = -\exp(-5x)\left[9979\sin(100x) + 900\cos(100x)\right], \quad 0 \leq x \leq 1, \quad (44)$$

with Dirichlet boundary conditions $u(0) = 0$ and $u(1) = \sin(100)\exp(-5)$. The exact solution can be verified to be

$$u_e(x) = \sin(100x)\exp(-5x), \quad (45)$$

which is highly oscillatory as shown in Figure 2.

The domain is partitioned into two subdomains of the same size, and each subdomain is discretized with uniformly distributed points. Twenty grids are considered with their densities varying from 11 to 201 in increment of 10. The accuracy of a numerical scheme is measured by means of the discrete relative L_2 norm of the solution (N_e). A test set of 201 uniform points is used to compute the error N_e.

The approximate solution is a C^1 function across the interface for MD-DRBFNs and C^2 for MD-IRBFNs.

Table 1 presents N_es of MD-DRBFNs and MD-IRBFNs obtained with $\beta = 1$. The performance of the latter is far superior to that of the former in both accuracy and convergence rate. The MD-DRBFN results are relatively poor. As mentioned earlier, the RBF widths strongly affect the accuracy of RBFNs. Figure 3 shows a significant improvement in accuracy for MD-DRBFNs with increasing RBF widths. However, the MD-DRBFN results with higher values of β are still less accurate than the MD-IRBFN results with $\beta = 1$. The process of increasing β to get a better solution needs to be conducted with great care. For example, the convergence behaviour becomes unstable for $\beta = 9$ and there is no grid convergence observed for $\beta = 11$. We also investigate how the domain partition affects the local error. Figure 4 shows the variation of the absolute error of the MD-IRBFN solution over the domain. It can be seen that the distribution of the error is quite uniform.

5.2 1D fourth-order problem

This test problem is governed by

$$\frac{d^4u}{dx^4} + \frac{d^2u}{dx^2} = \exp(-5x)\left[98490650\sin(100x) + 19949000\cos(100x)\right], \quad -1 \leq x \leq 1, \tag{46}$$

with Dirichlet boundary conditions

$$\begin{aligned} &u(0) = 0, \\ &\frac{du}{dx}(0) = 100, \\ &u(1) = \sin(100)\exp(-5), \\ &\frac{du}{dx}(1) = 100\cos(100)\exp(-5) - 5\sin(100)\exp(-5), \end{aligned}$$

for which the exact solution is also given in the form of (45). Four subdomains are considered. Grid convergence is studied using various sets of collocation points, namely $(11, 21, 31, \cdots, 701)$ points/subdomain. A test set of 201 uniformly distributed points is employed to compute the error N_e.

The continuity of the approximate solution at the interfaces is imposed up to third-order derivative for MD-DRBFNs and fourth-order derivative for MD-IRBFNs. Figure 5 shows accuracy against grid refinement by the two multidomain RBF techniques, indicating a superior performance of MD-IRBFNs over MD-DRBFNs. Unlike second-order problems, the increase in β for the case of MD-DRBFNs does not lead to better accuracy, probably due to the effect of implementing multiple boundary conditions. The ODE on each subdomain is collocated at $(n-4)$ grid points for MD-DRBFNs and n points for MD-IRBFNs. To investigate this issue separately, we solve the above problem by the one-domain DRBFN method. The same grid sizes are used. Two boundary points and two adjacent interior points are set to implement the double boundary conditions. Results obtained are shown in Figure 6. For $\beta = 1$, the convergence behaviour is smooth, but its rate is very slow. For higher values of β, the technique gives a fast rate of convergence over coarse grids. However, the accuracy of the solution then deteriorates rapidly over fine grids. It can be seen that the grid-convergence behaviour of the one-domain DRBFN method is not stable, which may cause a poor performance of the MD-DRBFN technique here.

5.3 2D second-order problem

Consider the Poisson equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = (1-\pi^2)\sin(\pi x)\sinh(y) + 4(1-\pi^2)\cosh(2x)\cos(2\pi y), \tag{47}$$

on the domain $-1 \leq x, y \leq 1$, subject to Dirichlet boundary conditions

$$u = \cosh(\pm 2)\cos(2\pi y), \quad x = \pm 1, \quad \text{and}$$

$$u = \sin(\pi x)\sinh(\pm 1) + \cosh(2x), \quad y = \pm 1.$$

The exact solution of this problem is given by

$$u_e(x, y) = \sin(\pi x)\sinh(y) + \cosh(2x)\cos(2\pi y). \tag{48}$$

Figure 7 shows the variation of (48) over the domain of interest. Only IRBFNs are applied here to solve this problem. The relative L_2 norm of the approximate solution (N_e) is computed at the grid points. Results concerning N_e for two cases, single domain and 16 subdomains, are presented in Table 2. Values of the condition number of the IRBFN system matrix vary from 1.0e0 to 2.4e3 for grids of 3×3 to 51×51/subdomain, respectively. It can be seen that the MD-IRBFN technique is able to employ much larger numbers of collocation points (e.g., up to 40,400 grid points used here without suffering from the problem of ill-conditioned matrices). The achievement of higher-order smoothness of the present DD solution across the subdomain interfaces is expected to alleviate the deterioration in accuracy caused by the domain division. Numerical results show that the convergence rate is reduced from $O(h^{3.51})$ (single domain) to $O(h^{3.07})$ (16 subdomains). This amount of reduction appears to be relatively small. Owing to smaller grid sizes used, the MD-IRBFN results are more accurate than the one-domain IRBFN results. Furthermore, the MD-IRBFN method only needs to handle a set of smaller matrices (i.e. a matrix for the unknowns on the interfaces and matrices for the interior values of the field variable on subdomains), leading to an improvement in computational efficiency over the one-domain IRBFN method.

5.4 2D fourth-order problem

This test problem is governed by the biharmonic equation with Dirichlet boundary conditions (u and $\partial u/\partial n$). The domain of interest is a square region of size 2×2 that is centered at the origin. The exact solution is chosen to be the same as that of the previous problem, i.e. (48), from which one can easily derive expressions for the forcing function b and the boundary conditions. The problem domain is decomposed into 16 subdomains of the same size, and each subdomain is discretized using a number of grids, namely 3×3, $5\times5, \cdots$, and 15×15. Relative L_2 errors (N_e) of the solution u obtained by the one-domain and multidomain IRBFN methods are presented in Table 3. Grid densities used for both methods are the same. The size of MD-IRBFN matrices is thus comparable with that of one-domain IRBFN matrices. However, the grid spacing of the former is 4 times smaller than that of the latter. For grids employed here, the condition numbers of the IRBFN system matrix are in the range of 1.0e0 to 4.5e4. As expected, the rate of convergence of the MD-IRBFN solution is reduced relative to the case of single domain. However, the DD method still yields a fast convergence rate up to $O(h^{4.42})$, resulting in very accurate solutions (e.g. $N_e(u)$=2.2e-5 is obtained with a density of 15×15). Given the same exact solution and the same number of subdomains, Tables 2 and 3 show that the MD-IRBFN technique performs better for the fourth-order PDE than for the second-order PDE.

6 Concluding remarks

In this paper, one-dimensional IRBFN interpolation schemes are incorporated into the substructuring technique for the solution of second- and fourth-order elliptic problems. The constants of integration in IRBFNs are exploited to improve the continuity order of the approximate solution across the subdomain interfaces. Additional enforcements are applied at the intersection points, which allows the solution at these points to be continuous across the subdomain interface with the same order as that at the interface points. Numerical results show that the proposed domain-decomposition technique yields a high level of accuracy and a fast rate of convergence. The multidomain IRBFN method thus appears to be more attractive than the one-domain IRBFN method for solving differential problems with large computational domains, where the use of large numbers of points is necessary.

Acknowledgement

This work is supported by the Australian Research Council. We would like to thank the referees for their helpful comments.

References

1. B.F. Smith, P.E. Bjorstad and W.D. Gropp, Domain Decomposition Parallel Multilevel Methods for Elliptic Partial Differential Equations, New York, Cambridge University Press, 1996.

2. 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.

3. G.E. Fasshauer, "Solving partial differential equations by collocation with radial basis functions", A. LeMhaut, C. Rabut and L.L. Schumaker, editors, Surface Fitting and Multiresolution Methods, Vanderbilt University Press, Nashville, TN, 1997, pp. 131–138.

4. M.R. Dubal, Domain decomposition and local refinement for multiquadric approximations. I: Second-order equations in one-dimension, Journal of Applied Science and Computation 1(1) (1994), 146–171.

5. E.J. Kansa and Y.C. Hon, Circumventing the ill-conditioning problem with multiquadric radial basis functions: applications to elliptic partial differential equations, Computers and Mathematics with Applications 39 (2000), 123–137.

6. J. Li and Y.C. Hon, Domain decomposition for radial basis meshless methods, Numerical Methods for Partial Differential Equations 20 (2004), 450–462.

7. M.S. Ingber, C.S. Chen and J.A. Tanski, A mesh free approach using radial basis functions and parallel domain decomposition for solving three-dimensional diffusion equations, International Journal for Numerical Methods in Engineering 60(13) (2004), 2183-2201.

8. E. Divo and A. Kassab, Iterative domain decomposition meshless method modeling of incompressible viscous flows and conjugate heat transfer, Engineering Analysis with Boundary Elements 30(6) (2006), 465–478.

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

10. C. Shu, H. Ding and 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.

11. B. Sarler and R. Vertnik, Meshfree explicit local radial basis function collocation method for diffusion problems, Computers & Mathematics with Applications 51(8) (2006), 1269–1282.

12. N. Mai-Duy and 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.

13. N. Mai-Duy and T. Tran-Cong, A Cartesian-grid collocation method based on radial-basis-function networks for solving PDEs in irregular domains, Numerical Methods for Partial Differential Equations 23 (2007), 1192–1210.

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

15. E.J. Kansa, H. Power, G.E. Fasshauer and L. Ling, A volumetric integral radial basis function method for time-dependent partial differential equations: I. Formulation, Engineering Analysis with Boundary Elements 28 (2004), 1191–1206.

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

17. L. Ling and M.R. Trummer, Multiquadric collocation method with integral formulation for boundary layer problems, Computers & Mathematics with Applications 48(5–6) (2004), 927–941.

18. S.A. Sarra, Integrated multiquadric radial basis function approximation methods, Computers & Mathematics with Applications 51(8) (2006), 1283–1296.

19. 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.

20. N. Mai-Duy and 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.

21. N. Mai-Duy and 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.

Table 1: 1D second-order problem, 2 subdomains, $\beta = 1$: Relative L_2 errors and their orders by the MD-DRBFN and MD-IRBFN methods

n/subdomain	MD-DRBFNs	MD-IRBFNs
11	1.8993e+1	2.2079e+1
21	1.7530e+0	2.7877e-1
31	1.1146e+0	7.3596e-2
41	8.3960e-1	3.4584e-2
51	6.7971e-1	1.8495e-2
61	5.7443e-1	1.0932e-2
71	4.9945e-1	6.9591e-3
81	4.4310e-1	4.6839e-3
91	3.9907e-1	3.2914e-3
101	3.6363e-1	2.3938e-3
111	3.3448e-1	1.7907e-3
121	3.1002e-1	1.3710e-3
131	2.8919e-1	1.0703e-3
141	2.7123e-1	8.4952e-4
151	2.5558e-1	6.8402e-4
161	2.4181e-1	5.5768e-4
171	2.2959e-1	4.5973e-4
181	2.1868e-1	3.8272e-4
191	2.0888e-1	3.2143e-4
201	2.0001e-1	2.7206e-4
	$O(h^{1.32})$	$O(h^{3.37})$

Table 2: 2D second-order problem: Relative L_2 errors and their orders by the one-domain and multidomain IRBFN methods

Single domain		Sixteen subdomains	
$n_x \times n_y$	$N_e(u)$	$n_x \times n_y$/subdomain	$N_e(u)$
3×3	1.0275e+0	3×3	9.0074e-2
7×7	2.6783e-2	7×7	2.6275e-3
11×11	4.1082e-3	11×11	5.7330e-4
15×15	1.1806e-3	15×15	2.0881e-4
19×19	4.6597e-4	19×19	9.7596e-5
23×23	2.2276e-4	23×23	5.2946e-5
27×27	1.2160e-4	27×27	3.1731e-5
31×31	7.3428e-5	31×31	2.0427e-5
35×35	4.8094e-5	35×35	1.3881e-5
39×39	3.3691e-5	39×39	9.8421e-6
43×43	2.4953e-5	43×43	7.2226e-6
47×47	1.9339e-5	47×47	5.4544e-6
51×51	1.5545e-5	51×51	4.2197e-6
	$O(h^{3.51})$		$O(h^{3.07})$

Table 3: 2D fourth-order problem: Relative L_2 errors and their orders by the one-domain and multidomain IRBFN methods

Single domain		Sixteen subdomains	
$n_x \times n_y$	$N_e(u)$	$n_x \times n_y$/subdomain	$N_e(u)$
3×3	3.6342e+1	3×3	1.4087e-001
5×5	2.9080e-1	5×5	2.3727e-003
7×7	4.0895e-2	7×7	6.9952e-004
9×9	7.1937e-3	9×9	2.3024e-004
11×11	2.3320e-3	11×11	7.9883e-005
13×13	1.0353e-3	13×13	3.0488e-005
15×15	5.5344e-4	15×15	2.2203e-005
	$O(h^{5.67})$		$O(h^{4.42})$

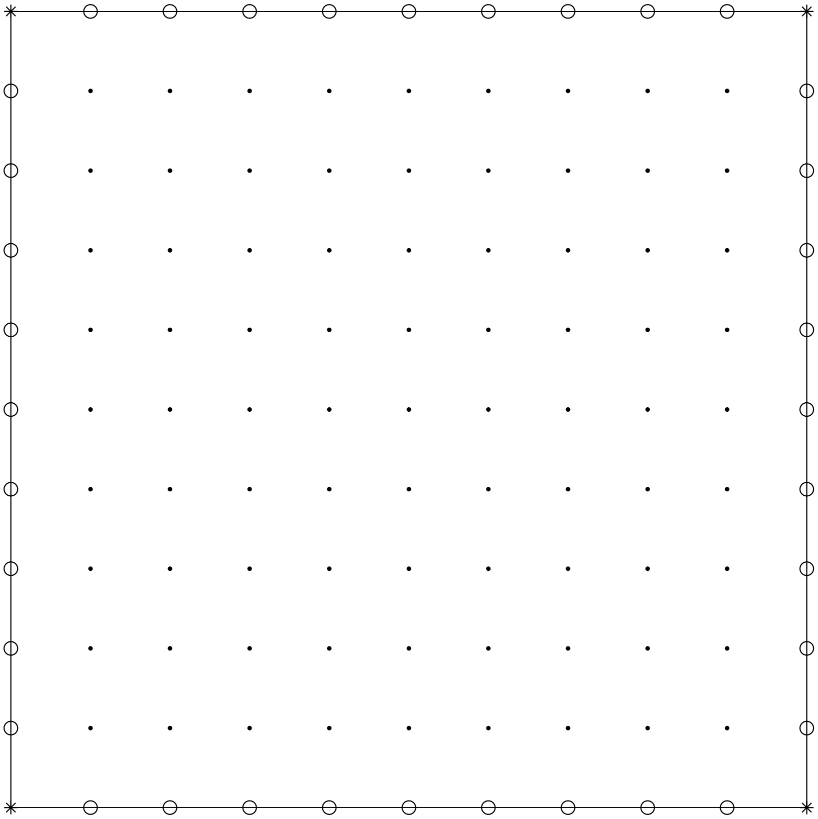

Figure 1: A typical subdomain. Legends $*$, $\circ$ and $\cdot$ are used to denote the intersection points (the corner points), the interior points on the interfaces (the interface points) and the interior points of the subdomain, respectively.

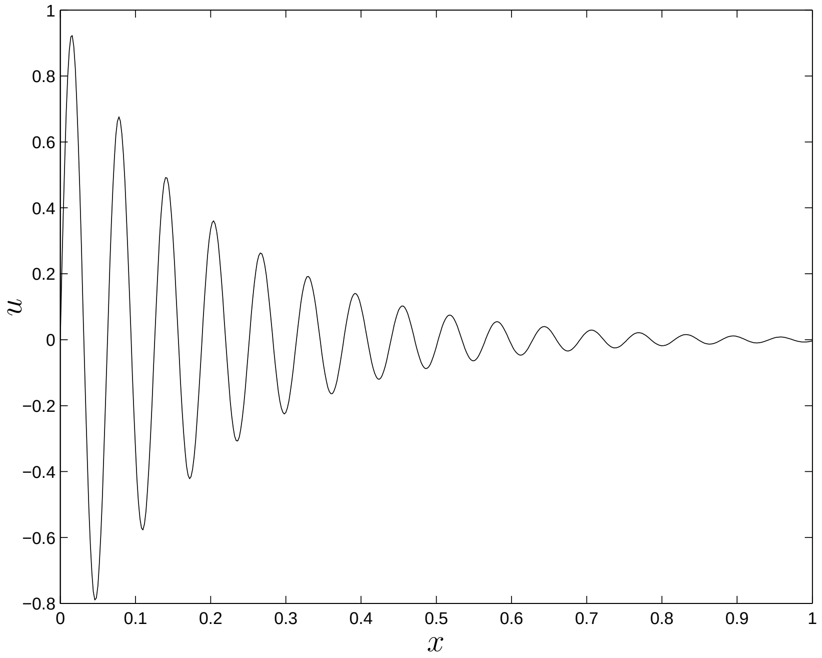

Figure 2: 1D problems: Exact solution.

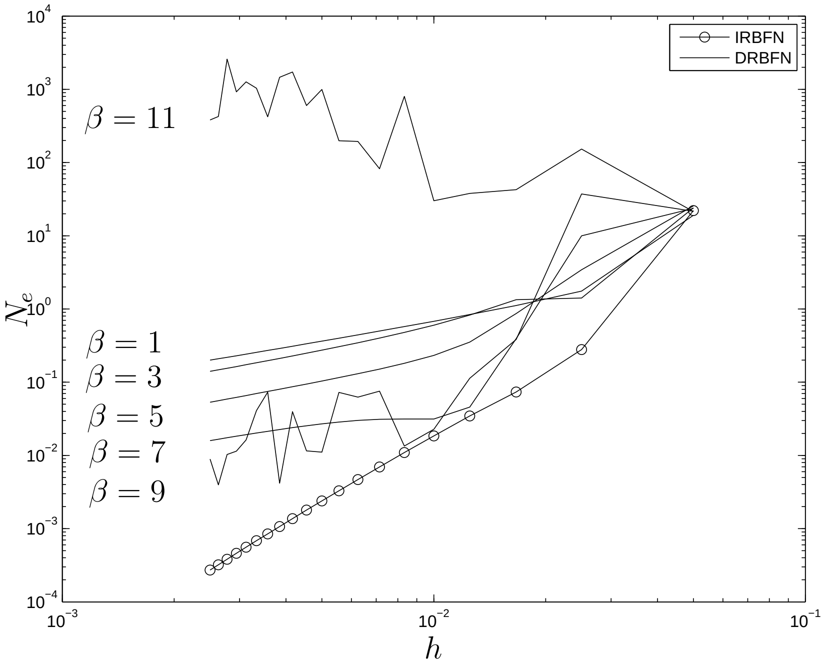

Figure 3: 1D second-order problem, 2 subdomains: Error N_e versus the grid spacing h by the MD-IRBFN ($\beta = 1$) and MD-DRBFN ($\beta = \{1, 3, 5, 7, 9, 11\}$) methods.

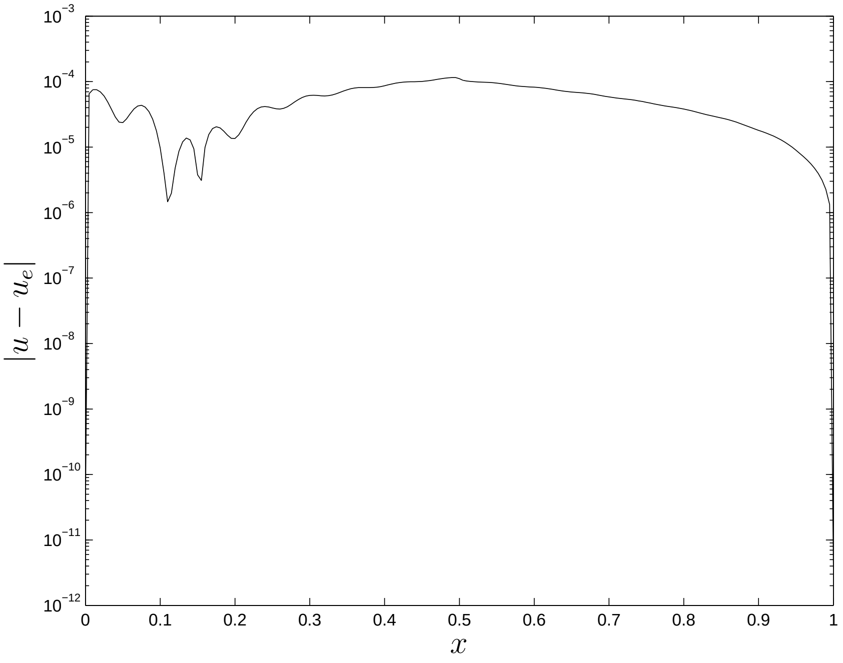

Figure 4: 1D second-order problem, 2 subdomains, 201 points/subdomain: Spatial distribution of the absolute error of the IRBFN solution.

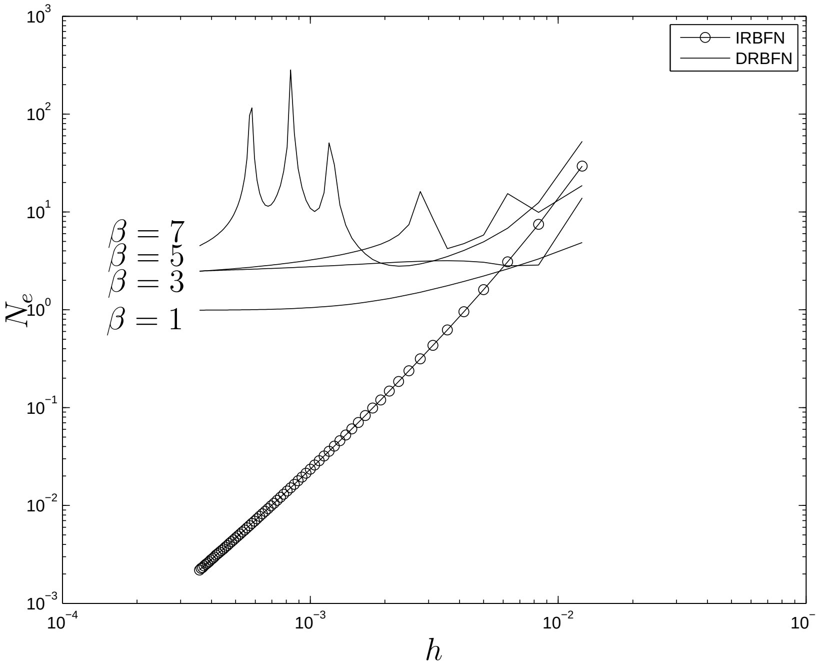

Figure 5: 1D fourth-order problem, 4 subdomains: Error N_e versus the grid spacing h by the MD-IRBFN ($\beta = 1$) and MD-DRBFN ($\beta = \{1, 3, 5, 7\}$) methods.

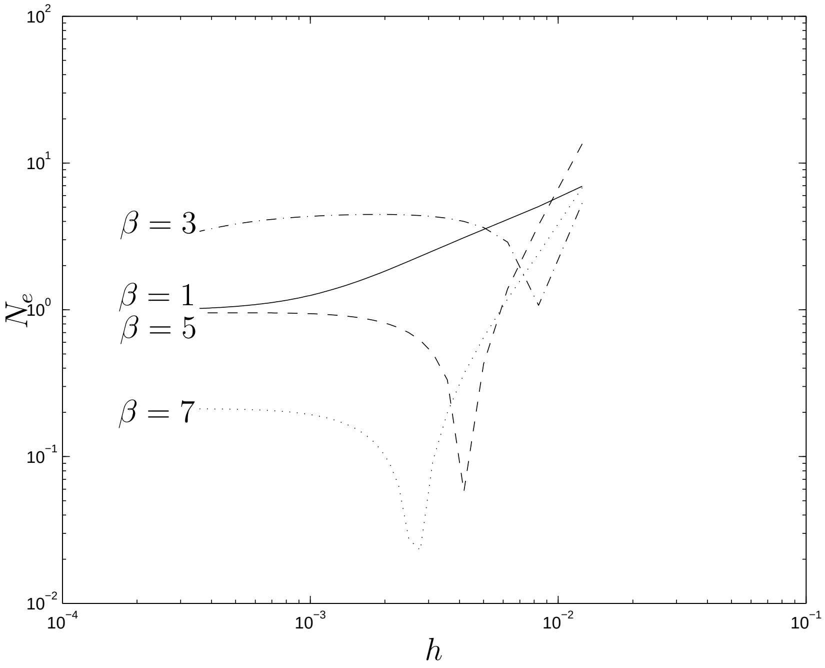

Figure 6: 1D fourth-order problem: Error N_e versus the grid spacing h by the one-domain DRBFN method for various values of β.

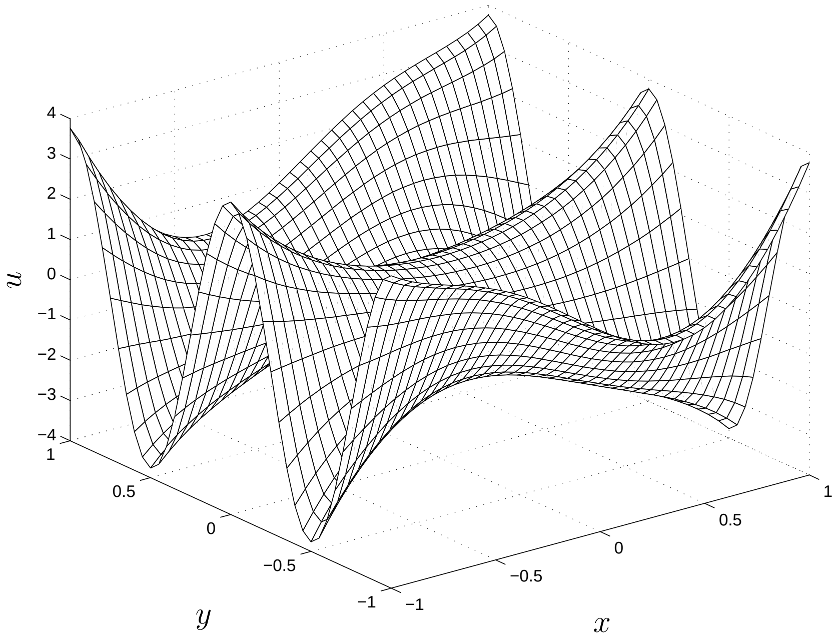

Figure 7: 2D problems: Exact solution.