

# On the use of integrated RBFs in Galerkin approximation for elliptic problems

N. Mai-Duy & T. Tran-Cong

*Computational Engineering and Science Research Centre,  
University of Southern Queensland, Toowoomba, QLD 4350, Australia*

## Abstract

This paper presents a new radial-basis-function (RBF) technique for solving elliptic differential equations (DEs). Integrated RBF networks (IRBFN) are constructed to satisfy (a) the boundary conditions in a local sense using the point-collocation formulation, and (b) the DE in a global sense using the Galerkin formulation. Several numerical examples are presented to demonstrate the attractiveness of the present implementation.

*Keywords: integrated RBFs, Galerkin formulation, Neumann boundary conditions, multiple boundary conditions, domain decomposition.*

## 1 Introduction

Many discretisation techniques can be viewed as variants of the method of weighted residuals that can be stated in three well-known formulations, namely the strong, weak and inverse statements [1]. By means of weighting functions in a statement, the residuals for the DE and boundary conditions are made small in some senses. Two popular ways used are (i) the point-collocation approach, where the residuals are zero at certain points and (ii) the Galerkin-type approach, in which the residuals are zero in the mean. Each approach has some advantages in certain areas of application. The former is cost-effective as no integrations are required, while the latter has a smoothing capability owing to its integral nature.

RBF collocation methods are considered as a powerful tool for the interpolation of scattered data as well as for the solution of differential problems [2]. The methods are capable of approximating arbitrarily-well continuous functions. A number of RBFs such as multiquadric and Gaussian functions have spectral approximation power. However, the condition number of the RBF interpolation matrix also grows



rapidly with respect to (a) the decrease in distance between the RBF centres and (b) the increase in the RBF width. RBF collocation methods thus, in practice, suffer from a trade-off between accuracy and stability. Moreover, there is a gap in accuracy between the RBF solutions to Neumann- and Dirichlet-type boundary-value problems. To improve the numerical stability of a RBF solution, there have been a number of schemes proposed in the literature: for example, (a) preconditioning the system matrix; (b) constructing the approximations locally; (c) using compactly-supported RBFs; and (d) employment of domain decompositions. Recently, an approximation scheme, which is based on point collocation, Cartesian grids and one-dimensional IRBFNs, has been proposed in [3, 4]. A problem domain, which can be regular or irregular, is discretised by a Cartesian grid. Along grid lines, 1D-IRBFNs are constructed to satisfy the governing DE together with boundary conditions in an exact manner. This scheme allows a larger number of nodes to be employed.

There are very few papers on the use of RBFs in the context of Galerkin approximation [2]. In this study, we present a new numerical scheme, which is based on the Galerkin formulation and 1D-IRBFNs, for solving elliptic problems. From a Galerkin-approach point of view, it will be shown that the present derivative boundary conditions are imposed in an exact manner. From a RBF-approach point of view, it will be shown that (a) the proposed method is able to yield almost the same levels of accuracy for the solutions of Neumann- and Dirichlet-type problems and that (b) its accuracy is considerably better than that of the 1D-IRBFN collocation method. An additional attractiveness of the proposed technique is that it facilitates a higher-order continuity of the solution across the subdomain interfaces.

The paper is organised as follows. Brief reviews of the Galerkin formulation and 1D-IRBFNs are given in Sections 2 and 3, respectively. The Galerkin 1D-IRBFN method is presented in Section 4, followed by several numerical examples in Section 5. Section 6 concludes the paper.

## 2 Galerkin approach

The Galerkin-type approach is well documented in the literature (e.g. [1]). A brief review of this approach is given below.

Consider a boundary-value problem defined by a linear DE and its boundary conditions

$$L(\bar{u}) = 0, \quad \mathbf{x} \in \Omega, \quad B(\bar{u}) = 0, \quad \mathbf{x} \in \Gamma, \quad (1)$$

where  $\bar{u}$  is the field/dependent variable (the overbar denotes the exact solution),  $L$  and  $B$  prescribed known operators,  $\Omega$  the domain of interest and  $\Gamma$  the boundaries of the domain  $\Omega$ .

An approximate solution, denoted by  $u$ , to (1) can be sought in the form

$$\bar{u}(\mathbf{x}) \approx u(\mathbf{x}) = \sum_{i=1}^N \alpha_i \phi_i(\mathbf{x}), \quad (2)$$



where  $\{\alpha_i\}_{i=1}^N$  is the set of unknown coefficients and  $\{\phi_i(\mathbf{x})\}_{i=1}^N$  the set of linearly-independent functions. The terms  $\phi_i$  are usually referred to as the trial/basis/approximating functions. Assume that a function  $u$  is constructed to satisfy the DE at every point on the domain  $\Omega$ , it leads to

$$\int_{\Omega} wL(u)d\Omega = 0, \quad (3)$$

for any function  $w$  that is bounded on  $\Omega$ . Similarly, assume that the approximate solution  $u$  also satisfies the boundary conditions, it follows that

$$\int_{\Gamma} \tilde{w}B(u)d\Gamma = 0, \quad (4)$$

for any bounded function  $\tilde{w}$ . The functions  $w$  and  $\tilde{w}$  are often referred to as the weighting/test functions. It can be seen that the system defined by (1) is equivalent to the following integral statement

$$\int_{\Omega} wL(u)d\Omega + \int_{\Gamma} \tilde{w}B(u)d\Gamma = 0, \quad (5)$$

that is satisfied for all bounded functions  $w$  and  $\tilde{w}$ .

If the weighting functions  $w$  and  $\tilde{w}$  have sufficient degrees of continuity, integrations by parts can be applied to derivative terms in (5), leading to other integral statements, namely the weak and inverse forms, that can be expressed as

$$\int_{\Omega} C(w)D(u)d\Omega + \int_{\Gamma} E(\tilde{w})F(u)d\Gamma = 0, \quad (6)$$

where the order of continuity required for the  $u$  solution is reduced. One can thus use either (5) or (6) to determine the approximate solution  $u$ . For the Galerkin-type approach, the weighting functions are chosen from the same set of functions as the trial functions.

### 3 One-dimensional integrated RBFNs

Consider a univariate function  $f(x)$ . The basic idea of the integral RBF scheme [3, 4] is to decompose a  $p$ th-order derivative of the function  $f$  into RBFs

$$\frac{d^p f(x)}{dx^p} = \sum_{i=1}^N w_i g_i(x) = \sum_{i=1}^N w_i I_i^{(p)}(x), \quad (7)$$

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

$$\frac{d^{p-1} f(x)}{dx^{p-1}} = \sum_{i=1}^N w_i I_i^{(p-1)}(x) + c_1, \dots, \quad (8)$$



$$f(x) = \sum_{i=1}^N w_i I_i^{(0)}(x) + c_1 \frac{x^{p-1}}{(p-1)!} + c_2 \frac{x^{p-2}}{(p-2)!} + \dots + c_{p-1}x + c_p, \quad (9)$$

where  $I_i^{(p-1)}(x) = \int I_i^{(p)}(x)dx, \dots, I_i^{(0)}(x) = \int I_i^{(1)}(x)dx$ , and  $\{c_1, c_2, \dots, c_p\}$  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  $p$ th-order derivative is taken as the starting point.

Evaluation of (7)–(9) at a set of collocation points  $\{x_j\}_{j=1}^N$  leads to

$$\frac{\widehat{d^p f}}{dx^p} = \widehat{\mathcal{I}}_{[p]}^{(p)} \widehat{\alpha}, \dots, \widehat{f} = \widehat{\mathcal{I}}_{[p]}^{(0)} \widehat{\alpha},$$

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

$$\widehat{\mathcal{I}}_{[p]}^{(p)} = \begin{bmatrix} I_1^{(p)}(x_1), & \dots, & I_N^{(p)}(x_1), & 0, & 0, & \dots, & 0, & 0 \\ I_1^{(p)}(x_2), & \dots, & I_N^{(p)}(x_2), & 0, & 0, & \dots, & 0, & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ I_1^{(p)}(x_N), & \dots, & I_N^{(p)}(x_N), & 0, & 0, & \dots, & 0, & 0 \end{bmatrix}, \dots$$

$$\widehat{\mathcal{I}}_{[p]}^{(0)} = \begin{bmatrix} I_1^{(0)}(x_1), & \dots, & I_N^{(0)}(x_1), & \frac{x_1^{p-1}}{(p-1)!}, & \frac{x_1^{p-2}}{(p-2)!}, & \dots, & x_1, & 1 \\ I_1^{(0)}(x_2), & \dots, & I_N^{(0)}(x_2), & \frac{x_2^{p-1}}{(p-1)!}, & \frac{x_2^{p-2}}{(p-2)!}, & \dots, & x_2, & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ I_1^{(0)}(x_N), & \dots, & I_N^{(0)}(x_N), & \frac{x_N^{p-1}}{(p-1)!}, & \frac{x_N^{p-2}}{(p-2)!}, & \dots, & x_N, & 1 \end{bmatrix};$$

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

and

$$\frac{\widehat{d^k f}}{dx^k} = \left( \frac{d^k f_1}{dx^k}, \frac{d^k f_2}{dx^k}, \dots, \frac{d^k f_N}{dx^k} \right)^T, \widehat{f} = (f_1, f_2, \dots, f_N)^T,$$

in which  $d^k f_j/dx^k = d^k f(x_j)/dx^k$  and  $f_j = f(x_j)$  with  $k = \{1, 2, \dots, p\}$  and  $j = \{1, 2, \dots, N\}$ .

### 4 Galerkin IRBFN technique

For Galerkin finite-element techniques, a weak statement (6), where the continuity requirement for the field variable  $u$  is reduced, is a preferred option. Piecewise polynomials of low order such as linear and quadratic interpolations are generally

used as approximating and weighting functions in numerous small subdomains called elements. In the case that the shape functions  $\varphi_i$  are algebraic polynomials, only the field variable changes continuously throughout the entire domain, and high derivatives (e.g. second derivatives for linear elements) are not defined. Essential boundary conditions are incorporated into the approximate solution prior to the process of discretising the DE, while natural boundary conditions are imposed by means of weighted residual (i.e. the second term in (6)). It should be emphasised that the natural boundary conditions in the weak formulation are approximated rather than identically satisfied. In engineering practice, such a partial satisfaction of the boundary conditions tends to give poor results for surface fluxes or tractions which make the overall results unreliable for many cases [1].

In the present Galerkin 1D-IRBFN technique, we use a Cartesian grid to generate the finite trial and test spaces. One dimensional IRBFNs are employed to represent the field variable and its derivatives on grid lines. The RBF solutions are constructed to satisfy the boundary conditions using the point-collocation approximation and the governing DE using the Galerkin approximation. A distinguishing feature here is that the networks are sought to satisfy a priori the derivative boundary conditions in an exact manner. There is thus no need to use the second term in (5) and (6). As the trial functions are infinitely-differentiable global functions, the present Galerkin 1D-IRBFN technique permits the employment of (5) to solve the differential problem of any order. Moreover, any derivative of the field variable is defined and continuous throughout the entire domain.

From an engineering viewpoint, one would prefer to work in the physical space. The present approximate solution is sought in terms of nodal variable values rather than the usual network weights. The boundary conditions including derivative information are imposed through the conversion process of the network-weight space into the physical space. RBFNs involve two types of data sets, namely centre and collocation points. In the context of point-collocation approximation, RBFNs tend to result in the most accurate approximations when the two sets of points are identical. Here, the collocation points are chosen to be the centres themselves. Unlike conventional differential formulations, the integral RBF formulation has the ability to generate additional coefficients (the constants of integration). This feature thus facilitates the addition of extra equations to the conversion system to represent extra information such as the natural boundary conditions and even the governing equation at the boundary points. The presence of integration constants thus guarantees that all RBFs are used for function approximation. In contrast, for conventional differentiated RBFNs, the enforcement of derivative function values is done at the price of the non-consideration of the function at some RBF centres, which significantly deteriorates the accuracy of the RBF scheme.

Consider a grid line. The conversion system for an 1D-IRBFN scheme of order  $p$  can be described as

$$\begin{pmatrix} \widehat{u} \\ \widehat{e} \end{pmatrix} = \begin{bmatrix} \widehat{\mathcal{I}}^{(0)} \\ \mathcal{I}^1 \\ \mathcal{K} \end{bmatrix} \widehat{\alpha} = \mathcal{C} \widehat{\alpha}, \quad (10)$$



where  $\widehat{e}$ , whose length can be up to  $p$ , is a vector representing extra information (e.g. normal derivative boundary conditions);  $\widehat{e} = \widehat{\mathcal{K}}\widehat{\alpha}$ ;  $\widehat{u}$  and  $\widehat{\mathcal{I}}_{[p]}^{(0)}$  defined as before; and  $\mathcal{C}$  the conversion matrix. It can be seen from (10) that the approximate solution  $u$  is collocated at the whole set of centres. Solving (10) for  $\widehat{\alpha}$  yields

$$\widehat{\alpha} = \mathcal{C}^{-1} \begin{pmatrix} \widehat{u} \\ \widehat{e} \end{pmatrix}, \tag{11}$$

where  $\mathcal{C}^{-1}$  is the inverse or pseudo-inverse of  $\mathcal{C}$ , depending on its dimension. Substitution of (11) into (7)–(9) leads to

$$u(x) = (I_1^{(0)}(x), I_2^{(0)}(x), \dots) \mathcal{C}^{-1} \begin{pmatrix} \widehat{u} \\ \widehat{e} \end{pmatrix}, \tag{12}$$

.....

$$\frac{\partial^p u(x)}{\partial x^p} = (I_1^{(p)}(x), I_2^{(p)}(x), \dots) \mathcal{C}^{-1} \begin{pmatrix} \widehat{u} \\ \widehat{e} \end{pmatrix}. \tag{13}$$

They can be rewritten in the form

$$u(x) = \sum_{i=1}^N \varphi_i(x) u_i + \varphi_{N+1}(x) e_1 + \varphi_{N+2}(x) e_2 + \dots, \tag{14}$$

.....

$$\frac{\partial^p u(x)}{\partial x^p} = \sum_{i=1}^N \frac{d^p \varphi_i(x)}{dx^p} u_i + \frac{d^p \varphi_{N+1}(x)}{dx^p} e_1 + \frac{d^p \varphi_{N+2}(x)}{dx^p} e_2 + \dots. \tag{15}$$

The Galerkin weighting process applied to (1) produces the results

$$\int_{\Omega} \varphi_i L(u) = 0, \tag{16}$$

where the values of  $i$  depend on the problem under consideration as will be discussed later. The system of equations, (16), can then be used to solve for the nodal value of the variable  $u$ .

### 5 Numerical results

Due to space limitation, only numerical results for second- and fourth-order ODEs are presented here. An ODE of order  $p$  is discretised by the 1D-IRBFN- $p$  scheme using multiquadrics (MQ). To generate finite spaces for the trial and test functions, uniformly-distributed MQ centres are employed. The MQ width is simply chosen to be the centre spacing. A 1D-IRBFN-based collocation method is also employed to provide a useful basis for comparison. The accuracy of an approximate solution is measured by means of the discrete relative  $L_2$  norm, denoted by  $N_e$ .

## 5.1 Second-order ODE

Find an approximate solution to the ODE

$$\frac{d^2 \bar{u}}{dx^2} + \bar{u} + x = 0, \quad (17)$$

on the interval  $0 \leq x \leq 1$ . The exact solution of (17) is assumed to be  $\bar{u} = \sin(x)/\sin(1) - x$ , from which one can easily derive the boundary values at  $x = 0$  and  $x = 1$ . We employ 45 sets of uniformly-distributed centres, varying from 3 to 91 with increment of 2. Two types of boundary conditions are considered.

**Dirichlet boundary conditions:** Both  $\mathcal{K}$  and  $\hat{e}$  are set to null. An approximate solution simply takes the form  $u(x) = \sum_{i=1}^N u_i \varphi_i(x)$ . The weighting functions are chosen to be the trial functions that are associated with the unknown nodal values of  $u$ . For this case, they are  $\{\varphi_2(x), \dots, \varphi_{N-1}(x)\}$ . Equation (16) leads to a determinate symmetric system of equations  $A$  for  $(N - 2)$  unknowns (i.e. the values of  $u$  at the interior points). It can be seen from Figure 1 (left) that the Galerkin technique yields much more accurate results and converges faster than the collocation technique.

**Dirichlet ( $x = 0$ ) and Neumann ( $x = 1$ ) boundary conditions:** We employ one extra equation to represent the derivative value at  $x = 1$ . Expression (14) becomes  $u(x) = \sum_{i=1}^N u_i \varphi_i(x) + (du_N/dx)\varphi_{N+1}(x)$  that also contains derivative information. The approximate solution thus satisfy a priori both the Dirichlet and Neumann boundary conditions in an exact manner. The unknown vector consists of the values of  $u$  at  $\{x_j\}_{j=2}^N$ . As a result,  $\{\varphi_2(x), \dots, \varphi_N(x)\}$  are taken as the weighting functions. Again, the Galerkin approach outperforms the collocation approach regarding accuracy (Figure 1 (right)).

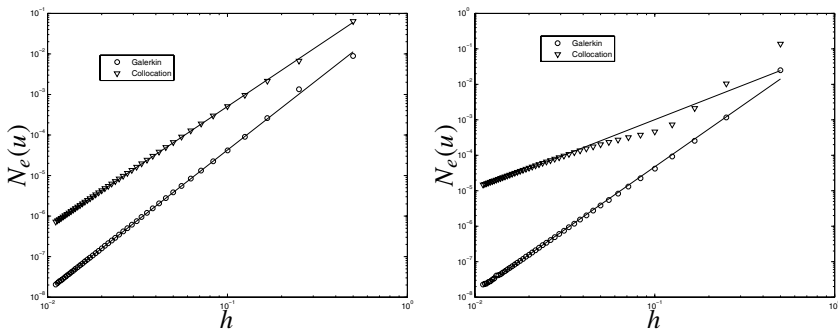


Figure 1: Second-order ODE: Error  $N_e(u)$  versus the centre spacing  $h$  for the Galerkin and collocation solutions. For Dirichlet conditions (left), they converge as  $O(h^{3.46})$  and  $O(h^{2.96})$ , respectively. For Neumann conditions (right), they are  $O(h^{3.54})$  and  $O(h^{1.98})$ .



It has been generally observed that the RBF results for the case of Neumann boundary conditions are generally much less accurate than for the case of Dirichlet boundary conditions. An attractive point here is that the present RBF method yields essentially the same degrees of accuracy for both types of boundary conditions (Figure 1). For both cases, values of the matrix condition number are relatively low, varying from  $O(10^1)$  to  $O(10^4)$ .

**5.2 Fourth-order ODE**

This example is governed by the biharmonic equation

$$\frac{d^4 \bar{u}}{dx^4} + \frac{d^2 \bar{u}}{dx^2} + (4k^3 - 2k) \cos(kx) - x(k^4 - k^2) \sin(kx) = 0, \quad (18)$$

on the interval  $-1/2 \leq x \leq +1/2$ , and Dirichlet boundary conditions. The exact solution to this problem is chosen to be  $\bar{u} = x \sin(kx)$ . We employ  $k = 7\pi/2$  that makes all boundary data nonzero. The present conversion process involves the enforcement of  $u$  at the whole set of centres and  $du/dx$  at the two boundary points, from which the approximate solution will take the form  $u(x) = \sum_{i=1}^N u_i \varphi_i(x) + (du_1/dx) \varphi_{N+1}(x) + (du_N/dx) \varphi_{N+2}(x)$ . The system matrix for solving  $\{u_i\}_{i=2}^{N-1}$  is then generated using the weighting functions  $\{\varphi_2(x), \dots, \varphi_{N-1}(x)\}$ . A number of uniform centre sets, namely 5, 7, 9, ..., 45 points, are employed. Figure 2 shows that the accuracy of the Galerkin method is far superior to that of the collocation method. The present Galerkin matrices have the condition number of  $O(10^1)$  to  $O(10^5)$ .

**5.3 Domain decomposition**

The most time-consuming part of the proposed technique lies in the process of computing volume integrals to form the algebraic system. One strategy to overcome this problem is to use domain decomposition. A substructuring technique

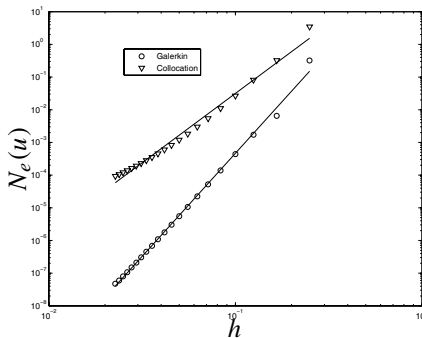


Figure 2: Fourth-order ODE:  $N_e(u)$  versus  $h$  for the Galerkin and collocation solutions. They converge as  $O(h^{6.31})$  and  $O(h^{4.24})$ , respectively.



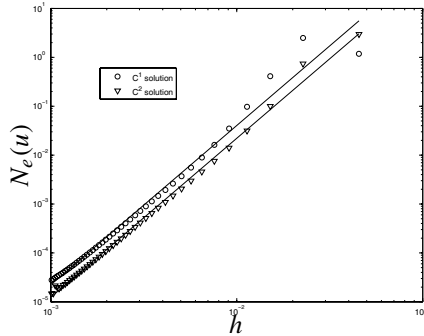


Figure 3: Domain decomposition, 11 subdomains: Errors of the present  $C^1$  and  $C^2$  Galerkin solutions.

will be applied here. The interface system can be constructed by requiring continuity of the field variable and its derivatives of order up to  $(p - 1)$  ( $p$ —the order of the DE) across the subdomain interfaces.

The combination of the substructuring technique and the proposed method that is presented above will lead to an approximate solution  $u$  that is a  $C^1$  function for second-order problems and  $C^3$  for fourth-order problems.

However, as shown in [5], a  $C^p$  solution can be achieved if the DE is enforced to be satisfied at the interface points in the subdomain solutions. This satisfaction can be made through the transformation of the network-weight space into the physical space.

Two versions of the present multidomain Galerkin technique are applied to the following ODE

$$\frac{d^2\bar{u}}{dx^2} + \frac{d\bar{u}}{dx} + \bar{u} = -\exp(-5x)[9979 \sin(100x) + 900 \cos(100x)], \quad 0 \leq x \leq 1, \quad (19)$$

with Dirichlet boundary conditions. The exact solution is chosen to be  $\bar{u}(x) = \sin(100x) \exp(-5x)$ , which is highly oscillatory. The domain is partitioned into 11 subdomains that are then identically represented using grids of  $\{3, 5, \dots, 91\}$  uniform points. Figure 3 clearly shows that the present  $C^2$  solution is, as expected, more stable and accurate than the  $C^1$  solution.

## 6 Concluding remarks

In this paper, a numerical technique, based on 1D-IRBFNs and Galerkin approximation, is developed for solving elliptic differential equations. In contrast to Galerkin finite-element techniques, the Neumann boundary conditions are presently imposed in an exact manner. Unlike conventional RBF techniques, the RBF approximations are presently constructed locally on grid lines through integration and they are expressed in terms of nodal variable values. The proposed

technique produces a system of algebraic equations that is often symmetric and has a relatively-low matrix condition number. To avoid the problem of high cost associated with the evaluation of volume integrals, the use of domain decomposition is discussed, where continuity order can be improved. Numerical results have shown that (a) the present technique achieves a high rate of convergence, (b) its accuracy is much higher than that of the integrated-RBF collocation technique, and (c) the obtained solutions have similar levels of accuracy for both Dirichlet- and Neumann-type boundary-value problems.

## Acknowledgement

This work is supported by the Australian Research Council.

## References

- [1] Brebbia, C.A. & Dominguez, J., *Boundary Elements—An Introductory Course*, Computational Mechanics Publications: Southampton, 1992.
- [2] Fasshauer G.E., *Meshfree Approximation Methods With Matlab*, World Scientific Publishers: Singapore, 2007.
- [3] Mai-Duy, N. & Tanner R.I., A collocation method based on one-dimensional RBF interpolation scheme for solving PDEs. *Int. J. of Numerical Methods for Heat & Fluid Flow*, **17(2)**, pp. 165–186, 2007.
- [4] Mai-Duy N. & Tran-Cong T., A Cartesian-grid collocation method based on radial-basis-function networks for solving PDEs in irregular domains. *Numerical Methods for Partial Differential Equations*, **23(5)**, pp. 1192–1210, 2007.
- [5] Mai-Duy N. & Tran-Cong T., A multidomain integrated radial basis function collocation method for elliptic problems. *Numerical Methods for Partial Differential Equations* (in press).

