PERGAMON

Neural Networks 14 (2001) 185–199

Neural Networks

www.elsevier.com/locate/neunet

Contributed article

Numerical solution of differential equations using multiquadric radial basis function networks

Nam Mai-Duy, Thanh Tran-Cong*

Faculty of Engineering and Surveying, University of Southern Queensland, Toowoomba, QLD 4350, Australia

Received 22 October 1999; revised 13 November 2000; accepted 13 November 2000

Abstract

This paper presents mesh-free procedures for solving linear differential equations (ODEs and elliptic PDEs) based on multiquadric (MQ) radial basis function networks (RBFNs). Based on our study of approximation of function and its derivatives using RBFNs that was reported in an earlier paper (Mai-Duy, N. & Tran-Cong, T. (1999). Approximation of function and its derivatives using radial basis function networks. *Neural networks*, submitted), new RBFN approximation procedures are developed in this paper for solving DEs, which can also be classified into two types: a direct (DRBFN) and an indirect (IRBFN) RBFN procedure. In the present procedures, the width of the RBFs is the only adjustable parameter according to $a^{(i)} = \beta d^{(i)}$, where $d^{(i)}$ is the distance from the ith centre to the nearest centre. The IRBFN method is more accurate than the DRBFN one and experience so far shows that β can be chosen in the range $7 \leq \beta \leq 10$ for the former. Different combinations of RBF centres and collocation points (uniformly and randomly distributed) are tested on both regularly and irregularly shaped domains. The results for a 1D Poisson's equation show that the DRBFN and the IRBFN procedures achieve a norm of error of at least $O(1.0 \times 10^{-4})$ and $O(1.0 \times 10^{-8})$, respectively, with a centre density of 50. Similarly, the results for a 2D Poisson's equation show that the DRBFN and the IRBFN procedures achieve a norm of error of at least $O(1.0 \times 10^{-3})$ and $O(1.0 \times 10^{-6})$ respectively, with a centre density of 12×12.

Keywords: Radial basis function networks; Multiquadric function; Global approximation; Mesh-free method; Solution of differential equation

1. Introduction

Many problems in science and engineering are reduced to a set of differential equations (DEs) through the process of mathematical modelling. Although model equations based on established physical laws may be constructed, analytical tools are frequently inadequate for the purpose of obtaining their closed form solution and usually numerical methods must be resorted to. Principal numerical methods available for solving DEs include the finite difference method (FDM) (Smith, 1978), the finite element method (FEM) (Cook, Malkus & Plesha, 1989; Hughes, 1987; Zienkiewicz & Taylor, 1991), the finite volume method (FVM) (Patankar, 1980) and the boundary element method (BEM) (Brebbia, Telles & Wrobel, 1984). These methods generally require some discretisation of the domain into a number of finite elements (FEs), which is not a straightforward task. In contrast to FE-type approximation, neural networks can be considered as approximation schemes where the input data for the design of a network consist of only a set of unstructured discrete data points. Thus, an application of neural networks for solving DEs can be regarded as a mesh-free numerical method. It has been proved that radial basis function networks (RBFNs) with one hidden layer are capable of universal approximation (Park & Sandberg, 1993). For problems of interpolation and approximation of scattered data, there is a body of evidence to indicate that the multiquadric (MQ) function yields more accurate results in comparison with other radial basis functions (Franke, 1982; Haykin, 1999; Powell, 1988, p. 265). Although it is not proved even experimentally in the present work that MQ function would result in superior accuracy for solving DEs, the MQ function is used to study the solution of DEs in this paper based on the above observation because the main aim of the present work is to demonstrate the procedure for solving DEs rather than the study of the property of the kernels. It is important to note that the accuracy of the RBFN solution is influenced by a parameter which is usually referred to as the width of basis function. The value of this parameter controls the shape of the basis function or the response of the associated neuron. In the case of learning network, the width parameter measures the degree to which excited neurons in the vicinity of the winning

* Corresponding author. Tel. +61-7-463-12539; fax: +61-7-463-12526.
E-mail address: trancong@usq.edu.au (T. Tran-Cong).

0893-6080/01/$ - see front matter
PII: S0893-6080(00)00095-2

Notations

Superscripts	denote elements of a set of neurons or collocation points
Subscripts	denote scalar components of a vector
n	number of collocation points
m	number of neurons
$\boldsymbol{x}$	independent variables
u_e	exact solution
u	approximant of the exact solution u_e
$u_{,ij\ldots l}$	partial derivative of function u with respect to $x_i x_j \ldots x_l$
u_i	approximate function u obtained by integrating $u_{,ii}$
g	radial basis function
h	basis function obtained by differentiating g
$\bar{h}$	basis function obtained by differentiating h
H	basis function obtained by integrating g
$\bar{H}$	basis function obtained by integrating H
w	weight of basis function
a	width of basis function
r	radius from the centre located at the neuron under consideration
c	'spatial position' of the neuron (centre)
$\|\cdot\|$	Euclidean norm
β	scalar factor
d	distance from the centre under consideration to the nearest centre

neuron participate in the learning process (Haykin, 1999). Similarly, in the case of functional approximation, the width parameter of a basis function (centre) controls its influence relative to the influence of its neighbours in the approximation. Large or small values make the neuronal response too flat or too peaked, respectively, and, therefore, both of these two extreme conditions should be avoided. In a study of multiquadric method for scattered data interpolation, Tarwater (1985) has found that by increasing the shape parameter a^2 (equivalent to the RBF's width in this paper), the root-mean-squared error of the goodness of fit drops to a minimum then increases sharply afterwards. In general, as the width increases the system of equations to be solved becomes ill-conditioned (Zerroukat, Power & Chen, 1998). By numerical experimentation, Kansa (1990a) found that the best results achieved by multiquadric approximation scheme occurred when the parameter a^2 is varied according to the following expansion

$$a^{(i)2} = a^2_{\min}(a^2_{\max}/a^2_{\min})^{(i-1)/(m-1)} \tag{1}$$

where $a^2_{\max}$ and $a^2_{\min}$ are input parameters; superscript (i) indexes the ith data point; and m is the number of data points. However, Kansa (1990a) did not report how $a^2_{\max}$ and $a^2_{\min}$ should be chosen until later, when Moridis and Kansa (1994) stated that the ratio $(a^2_{\max}/a^2_{\min})$ should be in the range of 10^1–10^6. Based on the formula (1), Kansa (1990b) and Sharan et al. (1997) have applied the multiquadric approximation scheme successfully for the numerical solution of PDEs by enforcing the equation at an appropriate number of collocation points in order to form a determined system of equations. Similarly, Dubal (1994) reported results of the numerical solution of ODEs. However, recently, in solving some problems of heat transfer, Zerroukat et al. (1998) found that a constant shape parameter ($a^{(i)} = a$) has achieved a better accuracy than a variable $a^{(i)}$ as given by (1) and concluded that improving the accuracy by varying the shape parameter cannot be considered as a general rule. Thus, it is important to note that $a^2_{\min}$ and the ratio $(a^2_{\max}/a^2_{\min})$ are problem-dependent and how to choose the best value of these parameters is still open. Recently, Mai-Duy and Tran-Cong (1999) have developed new methods based on RBFNs for the approximation of both functions and their first and higher derivatives. The so called direct RBFN (DRBFN) and indirect RBFN (IRBFN) methods were studied and it was found that the IRBFN method yields consistently better results for both function and its derivatives (Mai-Duy & Tran-Cong, 1999). The aim of this paper is to report the application of these DRBFN and IRBFN methods in solving DEs. In contrast to the approach taken by other authors as reviewed above, in the present methods the width of the ith neuron (centre) $a^{(i)}$ is determined according to the following simple relation (Moody & Darken, 1989)

$$a^{(i)} = \beta d^{(i)} \tag{2}$$

where β is a factor, $\beta > 0$, and $d^{(i)}$ is the distance from the ith centre to the nearest centre. Relation (2) indicates that it is reasonable to assign a larger width where the centres are widely separated from each other and a smaller width where the centres are closer. The results obtained (Mai-Duy & Tran-Cong, 1999) show that the IRBFN method achieves a better accuracy than the DRBFN method over a wide range of β. Specifically, for all problems attempted it was found that the best value for β is generally in the range of 7–10 for the IRBFN method. Although this empirical result might allow some tolerance in the choice of the β value, a more automatic algorithm such as the one proposed by Van Hulle (1998) is highly desirable. Van Hulle (1998) showed that the kernel-based maximum entropy learning rule (kMER) can achieve equiprobabilistic topographic map formation of regular, fixed-topology lattices for data-dependent nonparametric regression problems. However, the application of the latter method to the numerical solution of DEs, which is the case here, will need further investigation. Both direct and indirect methods are studied in this work for comparative purpose. The scope of this work is limited to linear ODEs and linear elliptic PDEs, with an indication of possible extension to other types of DEs in future work. The paper is organised as follows. A brief review of DRBFN and IRBFN methods for approximation

of function and its derivatives is given in Section 2. In Section 3 the DRBFN and IRBFN procedures are developed for solving DEs over regular domains which are defined in Section 3.1. Both procedures are illustrated with the aid of some numerical examples in Section 4. Two examples of 1D second order equations are discussed in Section 4.1. The solutions of some 2D elliptic PDEs with Dirichlet and Neumann boundary conditions in regular domains are demonstrated in Section 4.2. A procedure for solving PDEs on a domain with curved boundaries (irregular domain) is described with an illustrative example in Section 4.3. The effect of randomness of collocation points is investigated in Section 4.4. A discussion of the present methods with regard to other methods and types of DEs is given in Section 5. Section 6 concludes the paper.

2. Review of methods for approximation of function and its derivatives

Mai-Duy and Tran-Cong (1999) have reported the so called direct and indirect RBFN methods for an approximation of function and its derivatives given a set of discrete unstructured function values, $\{u(x^{(i)}) = y^{(i)}\}_{i=1}^{n}$, and demonstrated that the IRBFN method based on MQ RBF yields superior accuracy in comparison with the DRBFN. For the benefit of the present discussion the essence of the methods is summarised in this section. The function u to be approximated is defined by $u : R^p \rightarrow R^1$ and decomposed into basis functions as

$$u(\boldsymbol{x}) = \sum_{i=1}^{m} w^{(i)} g^{(i)}(\boldsymbol{x}) \tag{3}$$

where the set of radial basis functions $\{g^{(i)}\}_{i=1}^{m}$ with $m \leq n$ is chosen in advance and the set of weights $\{w^{(i)}\}_{i=1}^{m}$ is to be found. In the context of functional approximation, a discussion of the reason for $m \leq n$can be found, for example, in Haykin (1999, p. 278). In the present context of solving DEs, $m = n$ is normally chosen. Here, and in subsequent discussion, superscripts are used to index elements of a set of neurons or collocation points while subscripts denote scalar components of a p-dimensional vector. Given (3), the derivatives of the function, $u_{,j\ldots l}$, are calculated by

$$u_{,j\ldots l}(\boldsymbol{x}) = \frac{\partial^k u}{\partial x_j \ldots \partial x_l} = \sum_{i=1}^{m} w^{(i)} \frac{\partial^k g^{(i)}}{dx_j \ldots \partial x_l}. \tag{4}$$

In this work the chosen RBF is the MQ given by

$$g^{(i)}(\left\|\boldsymbol{x} - \boldsymbol{c}^{(i)}\right\|) = g^{(i)}(r) = \sqrt{r^2 + a^{(i)2}} \tag{5}$$

for some $a^{(i)} > 0$. With the model u decomposed into m fixed basis functions in a given family (3), the unknown weights $\{w^{(i)}\}_{i=1}^{m}$ are found, with the help of the general linear least squares principle, by minimising the sum squared error

$$SSE = \sum_{i=1}^{n} [y^{(i)} - u(\boldsymbol{x}^{(i)})]^2 \tag{6}$$

with respect to the weights of u.

2.1. Direct method

In the direct method (DRBFN) the closed form RBFN approximating function (3) is first obtained from a set of training points and the derivative functions are then calculated directly by differentiating such closed form RBFN. Thus, the decomposition of the function can be written as

$$u(\boldsymbol{x}) = \sum_{i=1}^{m} w^{(i)} g^{(i)}(\boldsymbol{x}) = \sum_{i=1}^{m} w^{(i)} \sqrt{r^2 + a^{(i)2}}. \tag{7}$$

Once the weights in (7) are found, the derivatives (e.g. up to second order with respect to x_j) are calculated by

$$u_{,j}(\boldsymbol{x}) = \sum_{i=1}^{m} w^{(i)} h^{(i)}(\boldsymbol{x}), \tag{8}$$

$$u_{,jj}(\boldsymbol{x}) = \sum_{i=1}^{m} w^{(i)} \bar{h}^{(i)}(\boldsymbol{x}), \tag{9}$$

where

$$h^{(i)}(\boldsymbol{x}) = \frac{\partial g^{(i)}}{\partial x_j} = \frac{x_j - c_j^{(i)}}{(r^2 + a^{(i)2})^{0.5}}, \tag{10}$$

$$\bar{h}^{(i)}(\boldsymbol{x}) = \frac{\partial h^{(i)}}{\partial x_j} = \frac{\partial^2 g^{(i)}}{\partial x_j \partial x_j} = \frac{r^2 + a^{(i)2} - (x_j - c_j^{(i)})^2}{(r^2 + a^{(i)2})^{1.5}}. \tag{11}$$

2.2. Indirect method

In the indirect method (IRBFN) the formulation of the problem starts with the decomposition of the derivative of the function into RBFs. The derivative expression is then integrated to yield an expression for the original function, which is then solved via the general linear least squares principle given an appropriate set of discrete data points. For example, in the case of multivariate functions with up to second derivatives, the relevant expressions are

$$u_{,jj}(\boldsymbol{x}) = \sum_{i=1}^{m} w^{(i)} g^{(i)}(\boldsymbol{x}) = \sum_{i=1}^{m} w^{(i)} \sqrt{r^2 + a^{(i)2}}, \tag{12}$$

$$u_{,j}(\boldsymbol{x}) = \sum_{i=1}^{m} w^{(i)} H^{(i)}(\boldsymbol{x}) + C_1, \tag{13}$$

$$u(\boldsymbol{x}) = \sum_{i=1}^{m} w^{(i)} \bar{H}^{(i)}(\boldsymbol{x}) + C_1 x_j + C_2, \tag{14}$$

where C_1 and C_2 are functions of independent variables

other than x_j and

$$H^{(i)}(\boldsymbol{x}) = \int g^{(i)}(\boldsymbol{x})\mathrm{d}x_j = \frac{(x_j - c_j^{(i)})\sqrt{r^2 + a^{(i)2}}}{2} + \frac{r^2 - (x_j - c_j^{(i)})^2 + a^{(i)2}}{2}\ln((x_j - c_j^{(i)}) + \sqrt{r^2 + a^{(i)2}}) \quad (15)$$

$$\bar{H}^{(i)}(\boldsymbol{x}) = \int H^{(i)}(\boldsymbol{x})\mathrm{d}x_j = \frac{(r^2 + a^{(i)2})^{1.5}}{6} + \frac{r^2 - (x_j - c_j^{(i)})^2 + a^{(i)2}}{2}(x_j - c_j^{(i)})\mathrm{In}((x_j - c_j^{(i)}) + \sqrt{r^2 + a^{(i)2}}) - \frac{r^2 - (x_j - c_j^{(i)})^2 + a^{(i)2}}{2}\sqrt{r^2 + a^{(i)2}}. \quad (16)$$

The detailed implementation and accuracy of the above methods were reported previously (Mai-Duy & Tran-Cong, 1999). The next section discusses the application of these methods in a solution procedure for DEs.

3. DRBFN and IRBFN procedures for solving DEs

For simplicity, let us consider the 2D Poisson's equation over the domain Ω

$$\nabla^2 u = p(\boldsymbol{x}), \qquad x \in \Omega \quad (17)$$

where ∇^2 is the Laplacian operator, $\boldsymbol{x}$ is the spatial position, p is a known function of $\boldsymbol{x}$ and u is the unknown function of $\boldsymbol{x}$ to be found. Eq. (17) is subject to Dirichlet and/or Neumann boundary conditions over the boundary Γ

$$u = p_1(\boldsymbol{x}), \qquad \boldsymbol{x} \in \Gamma_1, \quad (18)$$

$$\boldsymbol{n}\cdot\nabla u = p_2(\boldsymbol{x}) \qquad \boldsymbol{x} \in \Gamma_2 \quad (19)$$

where $\boldsymbol{n}$ is the outward unit normal; ∇ is the gradient operator; Γ_1 and Γ_2 are the boundaries of the domain such as $\Gamma_1 \cup \Gamma_2 = \Gamma$ and $\Gamma_1 \cap \Gamma_2 = \varnothing$; p_1 and p_2 are known functions of $\boldsymbol{x}$.

Numerical solution of DEs such as (17)–(19) is intimately connected with approximating function and its derivatives. It is proposed here that the solution u and its derivatives can be approximated in terms of basis functions (7)–(9) in a direct procedure or (12)–(14) in an indirect procedure. The design of networks is based on the information provided by the given DE and its boundary conditions. Note that in the present context of solving DEs, the 'data' points are more general collocation points instead of just actual given numerical values of the function to be approximated or interpolated. Thus, at a data (collocation) point either the DEs (in the case of internal points) or the DEs and the boundary conditions (in the case of boundary points) are forced to satisfy.

3.1. Definition of regular and irregular domains

In the present work, a regularly shaped domain is defined as a rectangular region in 2D or a parallelepiped region in 3D. For example, a 2D regular domain is defined by

$$a \le x_1 \le b; \qquad c \le x_2 \le d,$$

where a, b, c, d are constant. A domain that cannot be defined as above in any Cartesian coordinate system is called irregular. For example, any 2D domain with curved boundaries or 3D domain with curved surfaces are classified as irregular. If the domain is irregularly shaped it can be converted into a regularly shaped one as discussed in Section 4.3.

3.2. Collocation points versus RBF centres

In general, the location of the collocation points can be different from the location of the RBC centres. For the regular domain, the centres can be conveniently arranged on a regular grid and the collocation points can either be randomly distributed or they can be the same as the RBF centres (Fig. 1). Furthermore, the number of collocation points can be different from the number of RBF centres and the centres can be a subset of the set of collocation points. If the collocation points are the same as the RBF centres of the network then $m = n$.

3.3. DRBFN procedure

In the direct approach the sum squared error associated with (17)– (19) is given by

$$SSE = \sum_{x^{(i)}\in\Omega} [(u_{,11})\boldsymbol{x}^{(i)}) + u_{,22}(\boldsymbol{x}^{(i)})) - p(\boldsymbol{x}^{(i)})]^2 + \sum_{x^{(i)}\in\Gamma_1} [u(\boldsymbol{x}^{(i)}) - p_1(\boldsymbol{x}^{(i)})]^2 + \sum_{x^{(i)}\in\Gamma_2} [(n_1 u_{,1}(\boldsymbol{x}^{(i)}) + n_2 u_{,2}(\boldsymbol{x}^{(i)})) - p_2(\boldsymbol{x}^{(i)})]^2. \quad (20)$$

Upon substitution of the expressions for u and its derivatives, i.e. (7)–(9), into the above expression for *SSE* followed by the application of the linear least squares principle, a system of linear algebraic equations is obtained in terms of the unknown weights in the output layer of the network as follows

$$(\boldsymbol{G}^T\boldsymbol{G})\boldsymbol{w} = \boldsymbol{G}^T\hat{\boldsymbol{p}} \quad (21)$$

where $\boldsymbol{G}$ is the design matrix whose rows contain basis functions corresponding to the terms $(u_{,11}(\boldsymbol{x}^{(i)}) + u_{,22}(\boldsymbol{x}^{(i)}))$, $u(\boldsymbol{x}^{(i)})$ and $(n_1 u_{,1}(\boldsymbol{x}^{(i)}) + n_2 u_{,2}(\boldsymbol{x}^{(i)}))$ and therefore the number of rows is greater than the number of columns (number of neurons); $\boldsymbol{w}$ is the vector of weights and $\hat{p}$ is the vector whose elements correspond to the terms $p(\boldsymbol{x}^{(i)})$, $p_1(\boldsymbol{x}^{(i)})$ and $p_2(\boldsymbol{x}^{(i)})$.

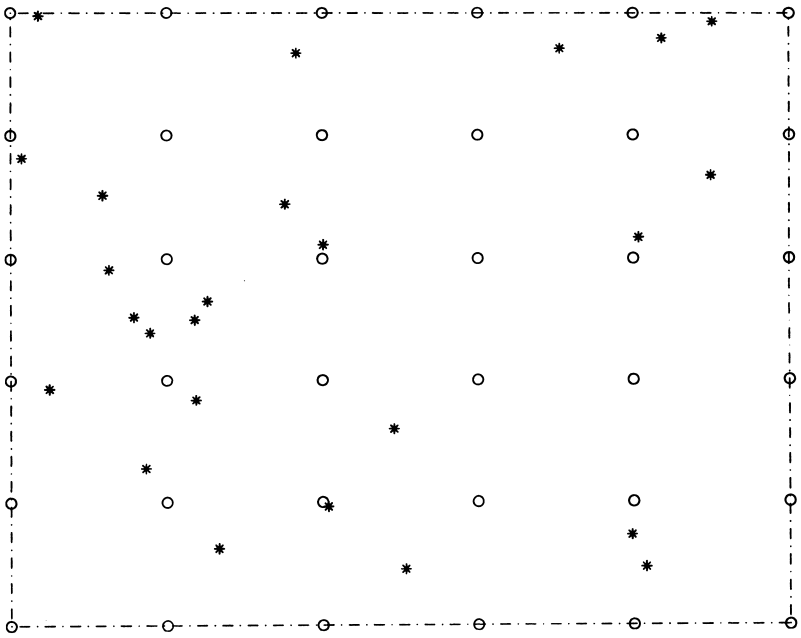

Fig. 1. RBF centres and collocation points. Legends ○: RBF centre and ∗: collocation point. RBF centres are regularly distributed for best results while collocation points can be random.

Alternatively, the solution u in the least squares sense (20) can be obtained by using the method of orthogonal-triangular decomposition with pivoting (or the QR method) (Dongarra, Bunch, Molwer & Stewart, 1979) for an over-determined system of equations, which in this case is

$$\boldsymbol{G}\boldsymbol{w} = \hat{\boldsymbol{p}}. \tag{22}$$

In practice, the QR method is able to produce the solution at larger values of β than the normal equations method arising from the linear least squares procedure (Eq. (21)) and hence the QR method is used in this work.

3.4. IRBFN procedure

Note that in the expressions (12)–(14) associated with the indirect method, the function u is obtained via a particular $u_{,jj}$ which is generally only one of a number of possible starting points. For the method to be correct, all starting points must lead to the same value for function u. Thus, in the indirect approach all possible starting points are taken into account and the sum squared error is given by

$$\begin{aligned} SSE = & \sum_{x^{(i)}\in\Omega} [(u_{,11}(\boldsymbol{x}^{(i)}) + u_{,22}(\boldsymbol{x}^{(i)})) - p(\boldsymbol{x}^{(i)})] \\ & + \sum_{x^{(i)}\in\Omega} [u_1(\boldsymbol{x}^{(i)}) - u_2(\boldsymbol{x}^{(i)})]^2 \\ & + \sum_{x^{(i)}\in\Gamma_1} [u_1(\boldsymbol{x}^{(i)}) - p_1(\boldsymbol{x}^{(i)})]^2 + \sum_{x^{(i)}\in\Gamma_2} [(n_1 u_{,1}(\boldsymbol{x}^{(i)}) \\ & + n_2 u_{,2}(\boldsymbol{x}^{(i)})) - p_2(\boldsymbol{x}^{(i)})]^2 \end{aligned} \tag{23}$$

where the term $u_1(\boldsymbol{x}^{(i)})$ is obtained via $u_{,11}$ and $u_2(\boldsymbol{x}^{(i)})$ is obtained via $u_{,22}$. Furthermore, the unknown in the direct procedure also contains the set of weights introduced by the interpolation of the constants of integration (e.g. C_1 and C_2 in (14)) in the remaining independent coordinate directions. For example, if C_1 is a constant of integration resulting from the integration of a basis function along the x_j direction then it is a function of all independent variables except x_j and hence it is interpolated along all directions except x_j. As in the case of direct procedure using the linear least squares principle, a system of equations (normal equations) can be obtained with appropriate substitution of the expression for the function u and its derivatives (12)– (14) into (23). However, it was found that the singular value decomposition (SVD) method (Press, Flannery, Teukolsky & Vetterling, 1988) provides superior results over a wide range of RBF's width (Mai-Duy & Tran-Cong, 1999) and hence SVD is used in this work.

4. Numerical examples

In the following examples, the value of β is varied over a wide range with an increment of 0.2 to investigate its effect on the accuracy of the solution. It appears that there is an upper limit for β above which the system of Eqs. (22) is ill-conditioned (see discussion in Section 5). In the present work, the value of β is considered to reach an upper limit when the system of Eqs. (22) becomes rank deficient.

A measure of the relative error of the solution or the norm

of the error of the solution, N_e, is defined as

$$N_e = \sqrt{\frac{\sum_{i=1}^{q} (u_e(\boldsymbol{x}^{(i)}) - u(\boldsymbol{x}^{(i)}))^2}{\sum_{i=1}^{q} u_e(\boldsymbol{x}^{(i)})^2}} \tag{24}$$

where $u(\boldsymbol{x}^{(i)})$ and $u_e(\boldsymbol{x}^{(i)})$ are the calculated and exact solution at the point i, respectively, and q is the number of collocation points. (Note that in the case of regularly shaped domains $q = n$ and in the case of irregularly shaped domains $q < n$ is the number of collocation points contained within the original domain only.)

4.1. 1D second order equations

4.1.1. Example 1

Consider the following 1D second order equation

$$\nabla^2 u = -16\pi^2 \sin(4\pi x) \tag{25}$$

on $0 \le x \le 1$ with $u = 2$ at $x = 0$ and $x = 1$. The exact solution can be verified to be

$$u_e(x) = 2 + \sin(4\pi x). \tag{26}$$

This problem was solved by Dubal (1994) using the multiquadric approximation scheme in conjunction with a domain decomposition technique. Subdomains are built from a 1D MQ approximation 'template' which is an $M \times M$ matrix constructed from multiquadric functions and their derivatives placed at M regularly spaced data points lying on the unit line. The best result with the 'average percentage relative error' of $-1.17 \times 10^{-2}\%$ was found in the case of 64 subdomains each of which contains 5 equally spaced points ($M = 5$) resulting in a total of 320 data points. In contrast, a total of 50 equally spaced points are used for the design of both DRBFN and IRBFN procedures in the present work. The accuracy of the solution is more dependent on the value of β in the case of DRBFN than in the case of IRBFN procedure as shown in Figs. 2 and 3. Fig. 3 shows that the IRBFN procedure achieves a better accuracy than the DRBFN procedure over a wide range of β. However, at some large values of β the DRBFN results are significantly improved. For example, at $\beta = 8.0$, the maximum errors are $2.08 \times 10^{-2}\%$ and $5.93 \times 10^{-6}\%$ for DRBFN and IRBFN, respectively.

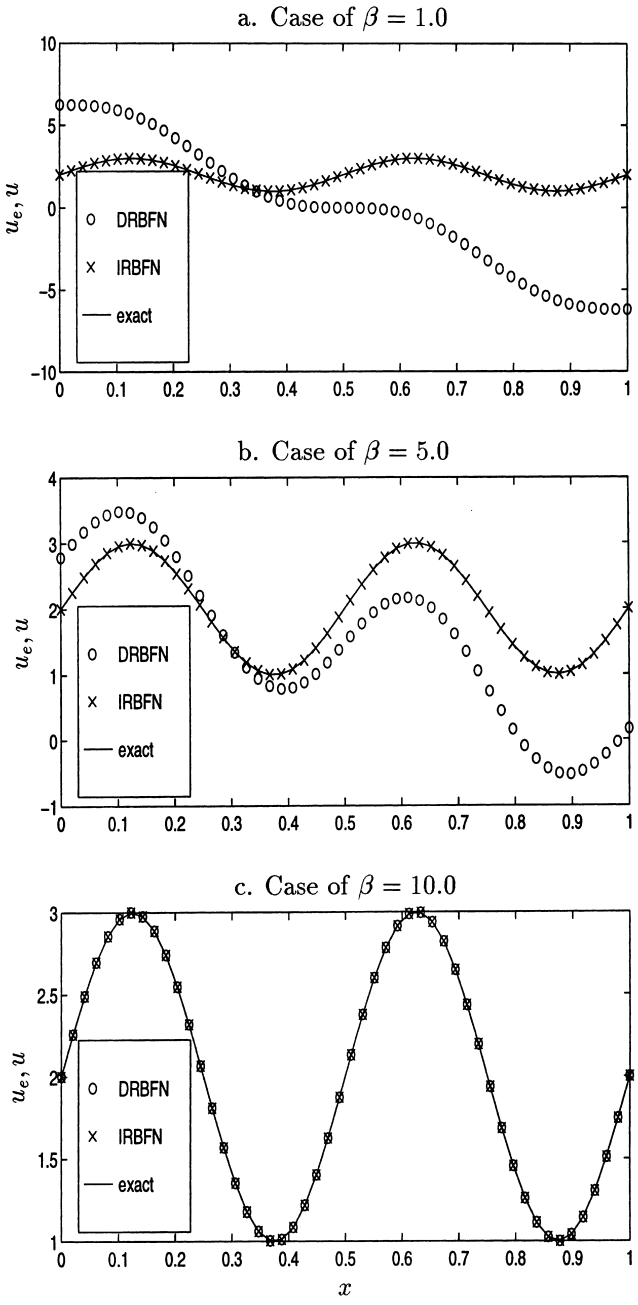

Fig. 2. Solution of $\nabla^2 u = -16\pi^2 \sin(\pi x)$: plots of the exact solution and the approximate solutions obtained from DRBFN and IRBFN procedures at some values of β. The centre density is 50 and uniform. The results show that the DRBFN method does not achieve an accuracy comparable with the IRBFN method until β is larger.

4.1.2. Example 2

As a second example, consider the following equation

$$\nabla^2 u - 3\nabla u + 2u = \exp(4x) \tag{27}$$

on $-1 \le x \le 1$, which has the exact solution

$$u_e(x) = B_1 \exp(x) + B_2 \exp(2x) + \frac{1}{6}\exp(4x) \tag{28}$$

and the boundary conditions are chosen so that $B_1 = 2$ and $B_2 = 1$. This problem was also studied by Dubal (1994). Using the same method as mentioned in Section 4.1.1, Dubal reported that the minimum value of the 'average percentage relative error' was $9.23 \times 10^{-3}\%$ in the case of 32 subdomains with $M = 5$ (i.e. 160 data points). In the present methods, a total of 50 equally spaced points are used for the design of DRBFN and IRBFN. The observations on the DRBFN and IRBFN results (Figs. 4 and 5) are similar to those indicated in the above example of Section 4.1.1. At

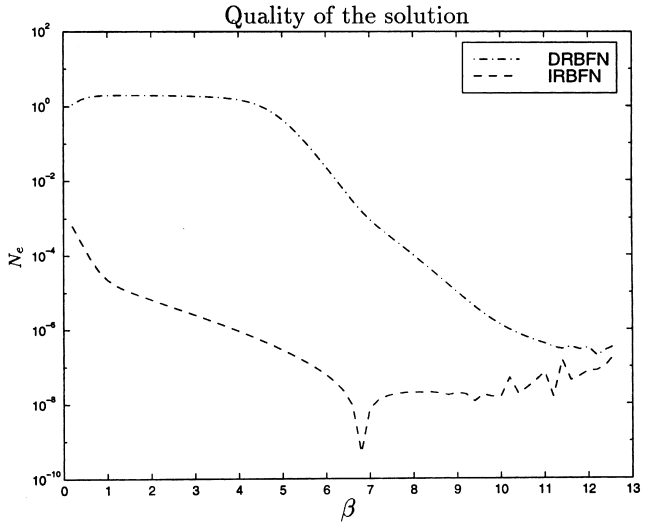

Fig. 3. Solution of $\nabla^2 u = -16\pi^2 \sin(\pi x)$: comparison of N_e between DRBFN and IRBFN procedures. The centre density is 50 and uniform. The IRBFN method performs better than the DRBFN method but it exhibits some peculiar behaviour at certain value of β. Both methods encounter singular system matrix at large β. Further explanation of these behaviours is given in later examples.

$\beta = 8.0$, the maximum errors are 1.98% for DRBFN and 1.51×10^{-5}% for IRBFN.

4.2. Elliptic PDE in regularly shaped domain

In this section two examples in 2D are studied.

4.2.1. Example 1

The problem here is to determine a function $u(x_1, x_2)$ satisfying the following PDE

$$\nabla^2 u = \sin(\pi x_1)\sin(\pi x_2) \tag{29}$$

on the rectangle $0 \le x_1 \le 1$, $0 \le x_2 \le 1$ subject to the Dirichlet condition $u = 0$ along the whole boundary of the domain. The exact solution is given by

$$u_e(x_1, x_2) = -\frac{1}{2\pi^2}\sin(\pi x_1)\sin(\pi x_2). \tag{30}$$

This problem was solved using feed forward neural networks (FFNN) by Dissanayake and Phan-Thien (1994). The authors used the following configurations 2–3–3–1 (i.e. two input nodes followed by two hidden layers of three nodes each and one output node), 2–5–5–1 and 2–10–10–1 to solve the problem using the following data point densities 5×5, 10×10 and 20×20. The best results obtained correspond to the last network architecture with the highest density. It is observed in the paper that the difference in the accuracy of the solutions are not significant between the last two configurations (i.e. 2–5–5–1 and 2–10–10–1) and also between the two denser data distributions (i.e. 10×10 and 20×20). The best FFNN results of Dissanayake and Phan-Thien (1994) are used here to compare

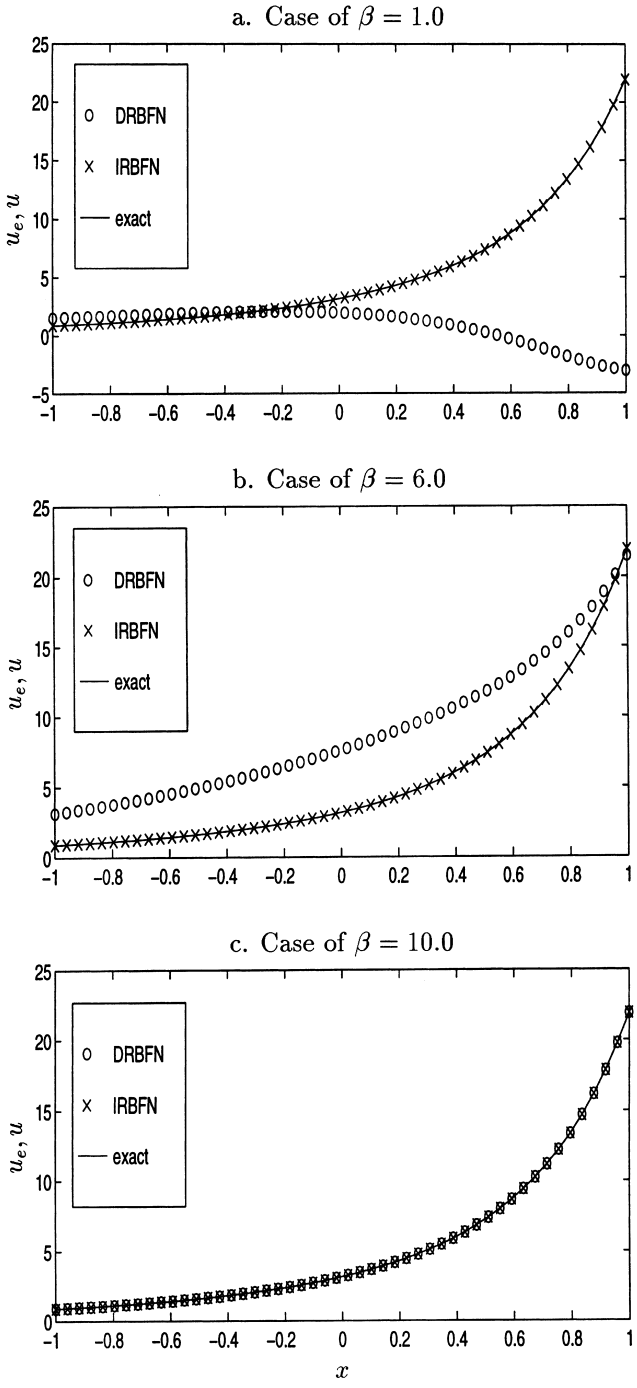

Fig. 4. Solution of $\nabla^2 u - 3\nabla u + 2u = \exp(4x)$: plots of the exact solution and the approximate solutions obtained from DRBFN and IRBFN procedures at some values of β. The centre density is 50 and uniform. The results show that the DRBFN method does not achieve an accuracy comparable with the IRBFN method until β is larger.

with the present DRBFN and IRBFN solutions. The comparison of N_e between FFNN, DRBFN and IRBFN solution is shown in Fig. 6 and it can be seen that the IRBFN solutions are the most accurate ones in all cases of data densities under consideration (i.e. 5×5, 10×10 and 20×20). The influence of data density on the N_e of the

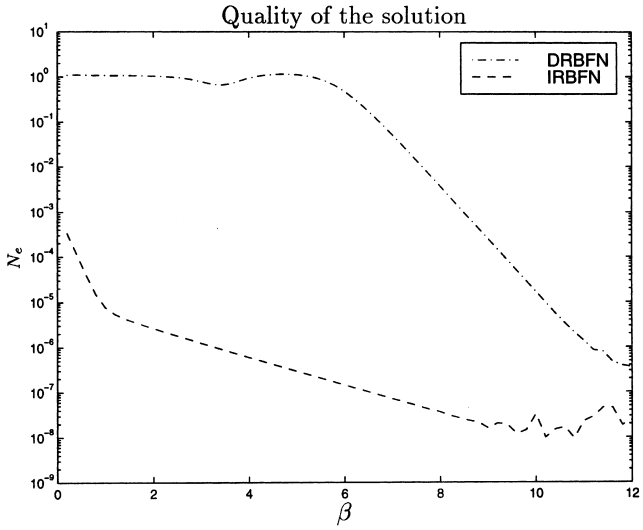

Fig. 5. Solution of $\nabla^2 u - 3\nabla u + 2u = \exp(4x)$: comparison of N_e between DRBFN and IRBFN procedures. The centre density is 50 and uniform. The IRBFN method performs better than the DRBFN method. Both methods encounter singular system matrix at large β. Further explanation of this behaviour is given in later examples.

DRBFN and IRBFN solutions is depicted in Fig. 7 which shows that increasing the data density makes the N_e–β curve more stable and results in an improvement of solution. However, the improvement only occurs at large values of β for the DRBFN procedure in contrast with a wider range of values of β for the IRBFN procedure. Furthermore, with β in the range $5 < \beta < 8.6$ the IRBFN solution has a norm of error of less than 10^{-6} in the case of 20×20 data density. Table 1 gives some indication of the rate of convergence of the solutions as the data density is increased for both the FFNN and IRBFN procedures. The table also shows that the IRBFN procedure appears to have a higher rate of convergence.

4.2.2. Example 2

In this example, the boundary conditions of the problem include both Dirichlet and Neumann type. Consider the following PDE

$$\nabla^2 u = (\lambda^2 + \mu^2)\exp(\lambda x_1 + \mu x_2) \tag{31}$$

on the rectangle $0 \le x_1 \le 1$, $0 \le x_2 \le 1$ with the following boundary conditions

$$u = \exp(\lambda x_1 + \mu x_2), \quad x_2 = 0 \text{ and } x_2 = 1,$$

$$u_{,1} = \lambda\exp(\lambda x_1 + \mu x_2), \quad x_1 = 0 \text{ and } x_1 = 1.$$

The exact solution is

$$u_e(x_1, x_2) = \exp(\lambda x_1 + \mu x_2). \tag{32}$$

Here, λ and μ are chosen to be 2 and 3, respectively. This problem was solved using the multiquadric approximation scheme by Kansa (1990b). The author used a total of 30 nodal points, including 12 scattered points in the interior

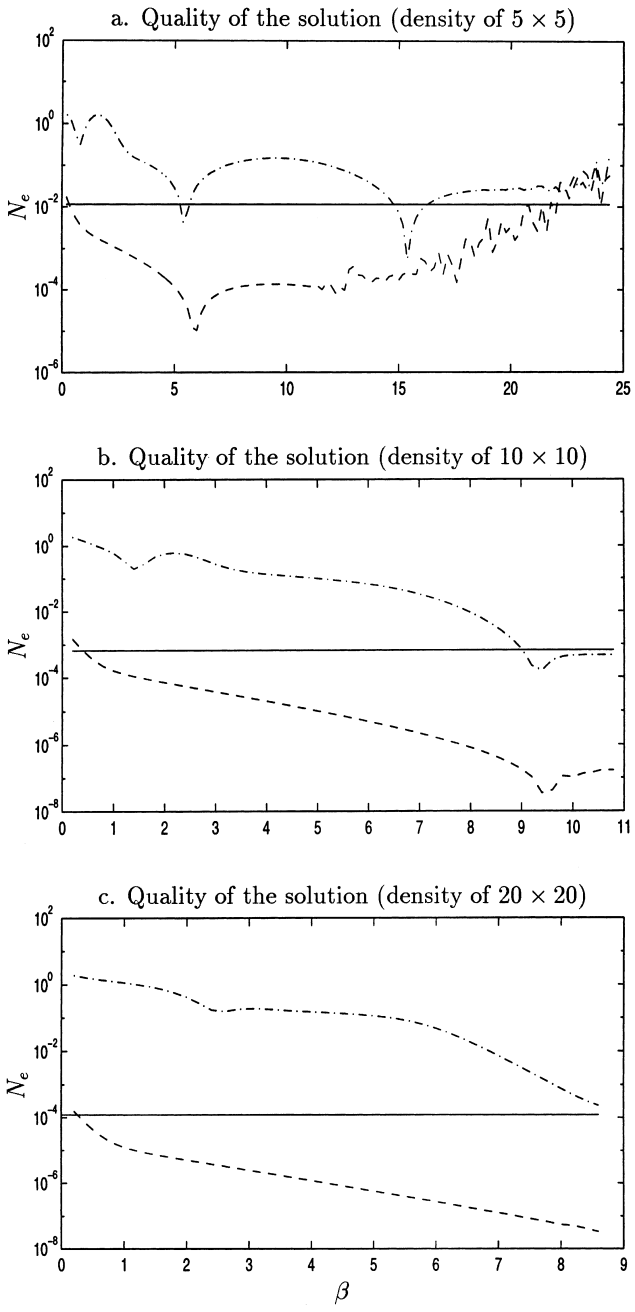

Fig. 6. Solution of $\nabla^2 u = \sin(\pi x_1)\sin(\pi x_2)$: comparison of N_e between FFNN, DRBFN and IRBFN procedures with some centre densities: 5×5, 10×10 and 20×20. Solid line: FFNN, dashdot line: DRBFN and dashed line: IRBFN. The FFNN results are taken from Dissanayake and Phan-Thien (1994). When the centre density is lower, the network ability to approximate is more limited and the peculiar sudden drops in the norm of error appear to be fortuitous. General improvement in accuracy is achieved with increasing centre density as shown. However, the onset of singular condition occurs sooner as the centre density increases.

and 18 along the boundary. The reported results showed that the norm of error is 1.66×10^{-2} (this error figure is calculated by the present authors using Kansa's reported results). Using the present methods, the results obtained here are similar to those in the previous example. The

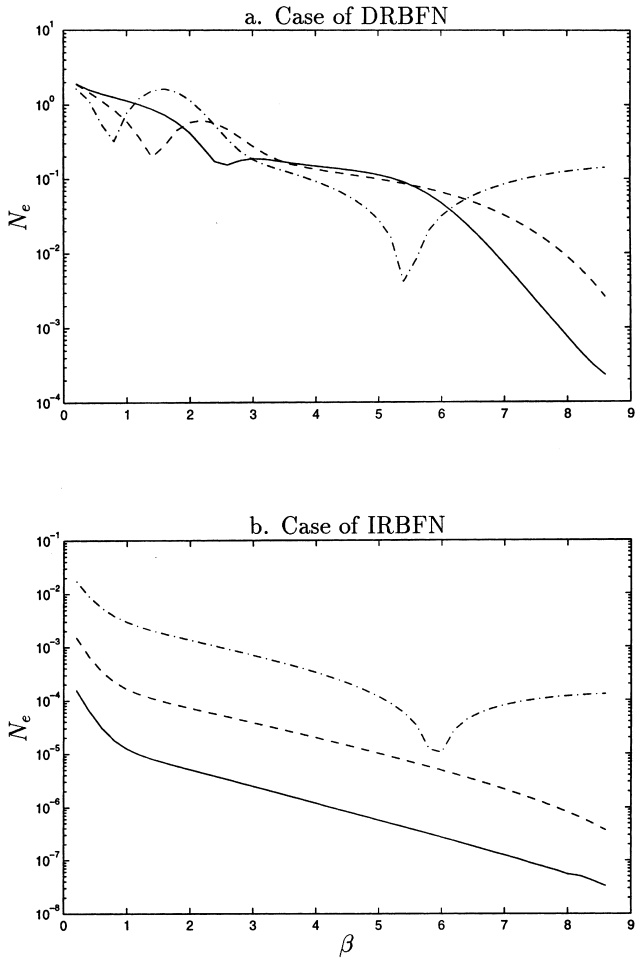

Fig. 7. Solution of $\nabla^2 u = \sin(\pi x_1)\sin(\pi x_2)$: effect of centre density on N_e of DRBFN and IRBFN results. Dashdot line: density of 5×5, dashed line: density of 10×10 and solid line: density of 20×20. In this plot the behaviours of the DRBFN and the IRBFN methods are better contrasted. The peculiar sudden isolated improvements in the norm of error generally disappear with increasing centre density in both methods. However, the performance of the IRBFN method generally improves over the whole range of β whereas the DRBFN method does not significantly improve until $\beta > 6$.

Table 1
N_e of FFNN and IRBFN solutions. The 'improve factor' is defined as the ratio of N_e between two data densities. The FFNN results are taken from Dissanayake and Phan-Thien (1994)

FFNN	IRBFN ($\beta = 5.0$)	
Density of 10×10	1.40×10^{-4}	1.01×10^{-5}
Density of 20×20	1.21×10^{-4}	5.72×10^{-7}
Improve factor	1.15	17.65

IRBFN solution achieves greater accuracy in all cases (data density of 6×6, 9×9 and 12×12) as shown in Figs. 8 and 9. For example, with the data density of 6×6 and $\beta = 8.0$, the norms of error N_es are 4.52×10^{-2} for DRBFN and 5.75×10^{-5} for IRBFN procedure.

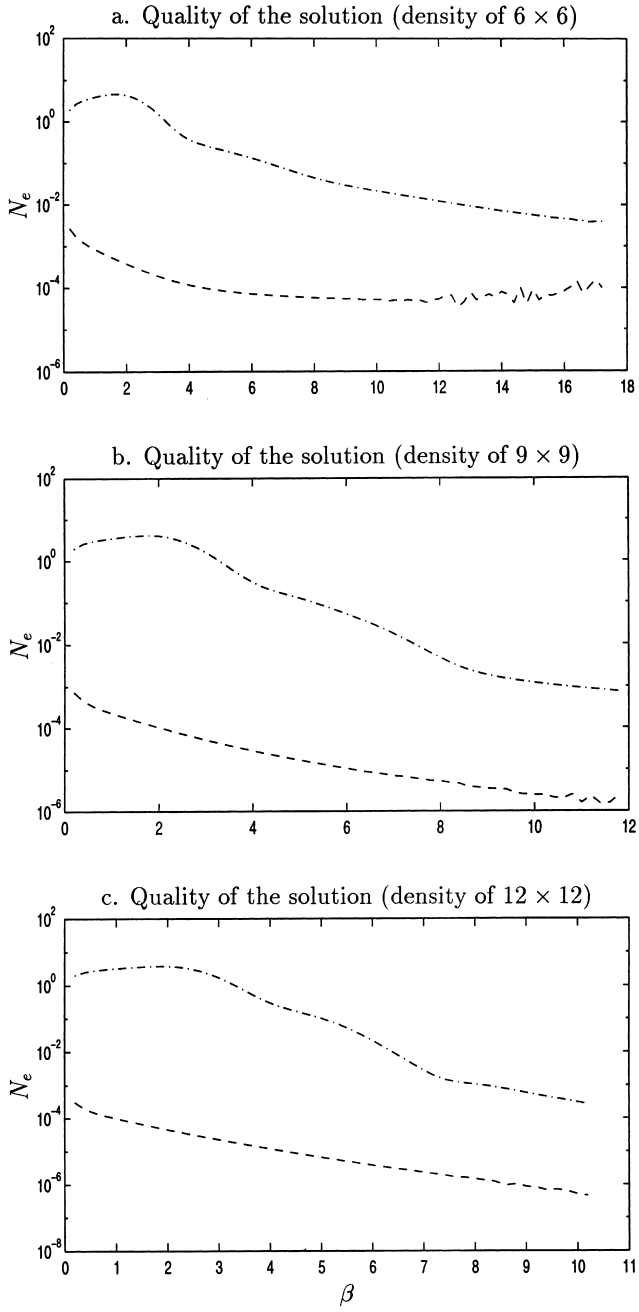

Fig. 8. Solution of $\nabla^2 u = (\lambda^2 + \mu^2)\exp(\lambda x_1 + \mu x_2)$: comparison of N_e between DRBFN and IRBFN procedures with data densities: 6×6, 9×9, 12×12. Dashdot line: DRBFN and dashed line: IRBFN. General improvement in accuracy is achieved with increasing centre density as shown. However, the onset of singular condition occurs sooner as the centre density increases.

4.3. Elliptic PDE in irregularly shaped domain

The 2D elliptic Eq. (17) subject to the boundary conditions (18) and (19) is reconsidered here. However, the domain Ω in this case is irregularly shaped. Suppose that

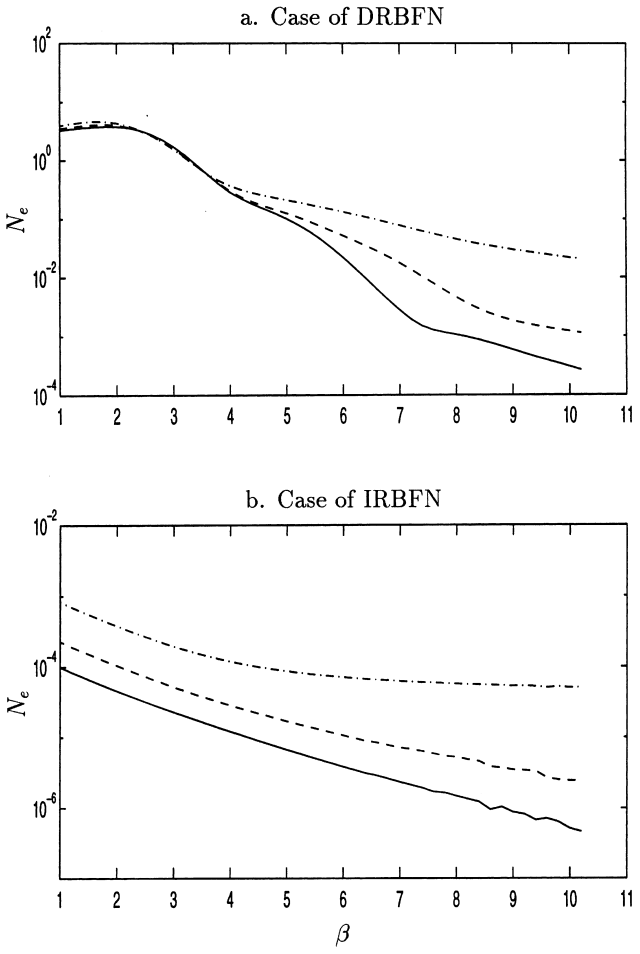

Fig. 9. Solution of $\nabla^2 u = (\lambda^2 + \mu^2)\exp(\lambda x_1 + \mu x_2)$: effect of data density on Ne of DRBFN and IRBFN results. Dashdot line: density of 6×6, dashed line: density of 9×9 and solid line: density of 12×12. In this plot the behaviours of the DRBFN and the IRBFN methods are better contrasted. The performance of the IRBFN method generally improves over the whole range of β whereas the DRBFN method does not significantly improve until $\beta > 6$.

(17) is rewritten as

$$\nabla^2 u = p(x), \qquad \boldsymbol{x} \in \Omega^* \supset \Omega \tag{33}$$

which is subject to the same boundary conditions (18) and (19) as before. It can be seen that a solution $u(\boldsymbol{x})$ of (33), (18) and (19) is also a solution of the original system (17)– (19). Based on this observation, a more convenient 'superdomain' Ω^* can be chosen to replace the original domain Ω subject to the condition $\Omega^* \supset \Omega$.. The boundary of the original domain therefore becomes an internal surface within Ω^*. For example, a convenient superdomain for the present 2D problem is a set of appropriate rectangles that is sufficiently large to completely cover the original domain with curved boundary (Fig. 10). A set of centres can then be regularly defined on this regular superdomain Ω^*. The collocation points then consist of all the centres and points used to define the original boundary. For example, a set of uniformly distributed collocation points can be easily defined on Ω^* and the boundary collocation points can be defined as the intersection of the grid lines with the original boundary curve. Then, DRBFN and IRBFN are designed using the regularly shaped domain with only regularly distributed collocation points as RBF centres. The boundary conditions are decomposed into basis functions of the new domain of the network and imposed at the boundary points thus created.

For the purpose of illustration, let us consider the following Poisson's equation

$$\nabla^2 u = -2 \tag{34}$$

on an elliptical domain with semi-major axis of length $a = 10$ and semi-minor axis of length $b = 8$. The homogeneous condition $u = 0$ is imposed along the whole boundary. The exact solution is

$$u_e(x_1, x_2) = -\left(\frac{a^2 b^2}{a^2 + b^2}\right)\left(\frac{x_1^2}{a^2} + \frac{x_2^2}{b^2} - 1\right). \tag{35}$$

The problem was solved using a multiquadric approximation scheme by Sharan, Kansa and Gupta (1997). Due to the symmetry of this problem, the authors considered only the first quadrant of the ellipse using 28 data points. The maximum error was found to be $3.57 \times 10^{-4}\%$ and the accuracy was up to five significant decimal places. In the present methods, the rectangular superdomain used is 10×8 and the density of centres is 11×9, which completely covers the elliptical domain (the number of collocation points corresponding to the first quadrant is the same as that of Sharan et al. (1997). The comparison of N_e between DRBFN and IRBFN solutions is depicted in Fig. 11 showing again that the IRBFN solution is more accurate than the DRBFN solution. The solutions corresponding to larger values of β are very accurate. For example, at $\beta = 8.0$, the accuracies are up to four significant decimal places for DRBFN results and eight significant decimal places for IRBFN results with maximum errors being $3.56 \times 10^{-3}\%$ and $2.27 \times 10^{-6}\%$, respectively. To further demonstrate the accuracy of the methods the norm of error is calculated at 29 random points, which is 1.3561×10^{-5} and 5.411×10^{-7} for the DRBFN and IRBFN procedures, respectively. The maximum error is $1.58 \times 10^{-2}\%$ and $3.80 \times 10^{-4}\%$ for the DRBFN and IRBFN procedures, respectively.

4.4. Random collocation points

In the examples discussed so far the centres are also the collocation points in the case of regularly shaped domains. In the case of irregularly shaped domains, extra collocation points are generated on the curved boundary in order to accurately describe the boundary of the domain. In this section the effects of randomness of internal collocation points are investigated. Fig. 1 illustrates the distribution of regular RBF centres and random collocation points. The

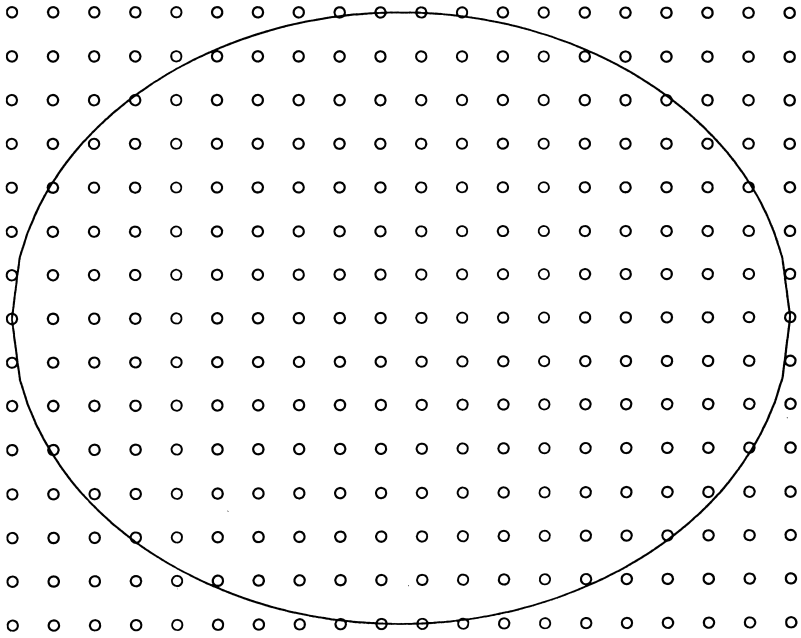

Fig. 10. Regularly extended domain: in the present work, a regularly shaped domain is defined as a rectangular region in 2D or a parallelepiped region in 3D. For example, a 2D regular domain is defined by $a \leq x_1 \leq b$; $c \leq x_2 \leq d$, where a, b, c, d are constant. A domain that cannot be defined as above in any Cartesian coordinate system is called irregular. The figure illustrates the method of dealing with irregular domains by containing it in a convenient set of regular domains.

RBF centres are arranged on a regular grid and can be different from the collocation points. However, the set of collocation points can include the centres as a subset. The 2D example of Section 4.2.2. is reconsidered here with random collocation points. In order to compare the present results with those of Kansa (1990b) the same number of internal random collocation points (12 points) and the same boundary collocation points (18 points uniformly distributed) are used as shown in Fig. 12. The centres (not shown) are uniformly distributed with a density of 6×6. Fig. 13 shows that the present methods produce results which are more accurate than those reported by Kansa (1990b) for the same problem. Further improvements can be achieved by including the centres in the set of collocation points as shown in Fig. 14. Note that this inclusion increases the number of collocation points without changing the number of unknowns. Fig. 14 also shows that more accurate results are obtained by increasing the centre density as well as including the centres in the set of collocation points.

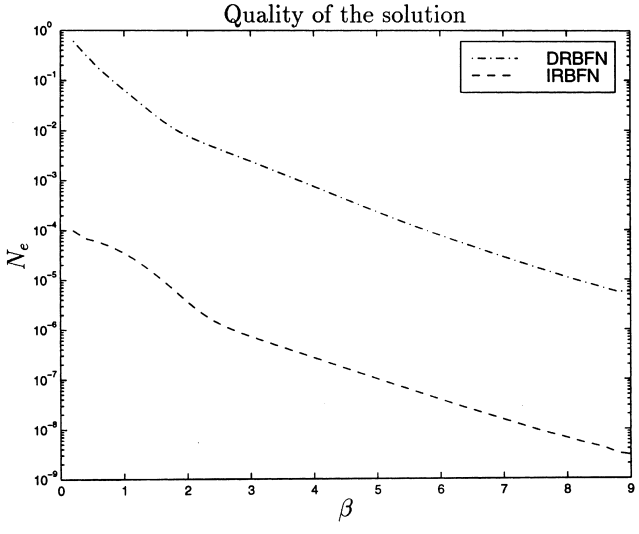

Fig. 11. Poisson's equation in an elliptical domain: comparison of N_e between DRBFN and IRBFN procedures.

5. Discussion

5.1. Types of DEs and accuracy of the IRBFN method

The scope of the present work is limited to the solution of linear ODEs and linear elliptic PDEs. The results presented in earlier sections show that the present DRBFN method yields similar accuracy to other methods found in the literature. On the other hand, the present IRBFN method produces results which are several orders of magnitude more accurate than those associated with the DRBFN method (accuracy is measured in terms of norm of error). The above conclusion corresponds to the experimentally found 'best' values of β in the range 7–10. Unfortunately, a theoretical determination of an optimal value for β is, to our knowledge, not available. The better norm of error achieved in the IRBFN method is also underlined by a smooth error distribution over the entire domain as shown in Figs. 15–17 where further comments can be found in the captions. Although the error distribution for the IRBFN is seen to be smooth in the present data-independent procedure

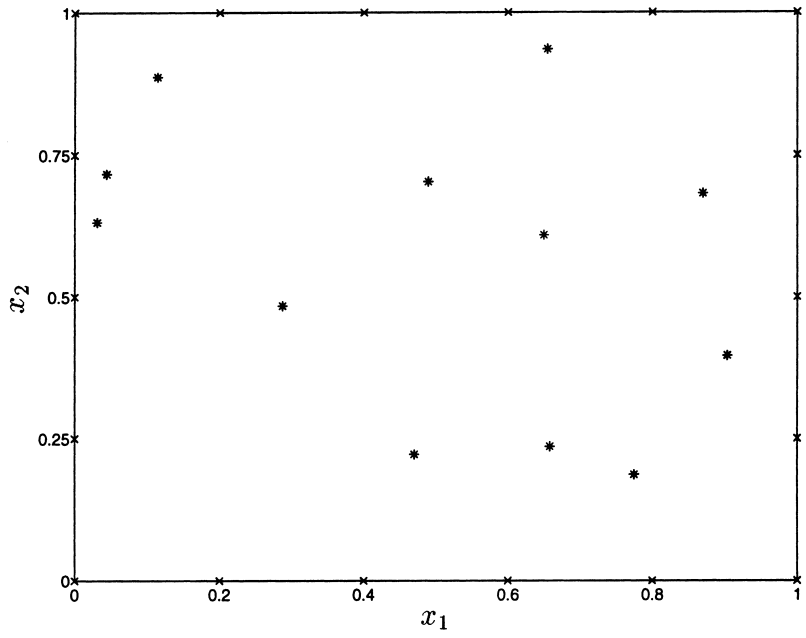

Fig. 12. Random internal collocation points for the solution of $\nabla^2 u = (\lambda^2 + \mu^2)\exp(\lambda x_1 + \mu x_2)$. Legends $*$: internal random point and $\times$: boundary point. For the purpose of comparison with the results of Kansa (1990b), the same number of collocation points are used, however the randomness cannot be made the same.

(data-independent in the sense that the centres are fixed and hence the kernel width is also fixed according to Eq. (2) for a given β), Van Hulle (1998) showed that the kernel-based maximum entropy learning rule (kMER) can achieve equiprobabilistic topographic map formation for data-dependent nonparametric regression problems. The application of the latter method to the numerical solution of DEs will need further investigation in future work.

5.2. Why is the IRBFN method more accurate?

A formal theoretical proof of the superior accuracy of the present IRBFN method cannot be offered at this stage, at least by the present authors. However, a heuristic argument can be presented as follows. In the direct methods, the starting point is the decomposition of the unknown functions into some finite basis and all derivatives are obtained as a

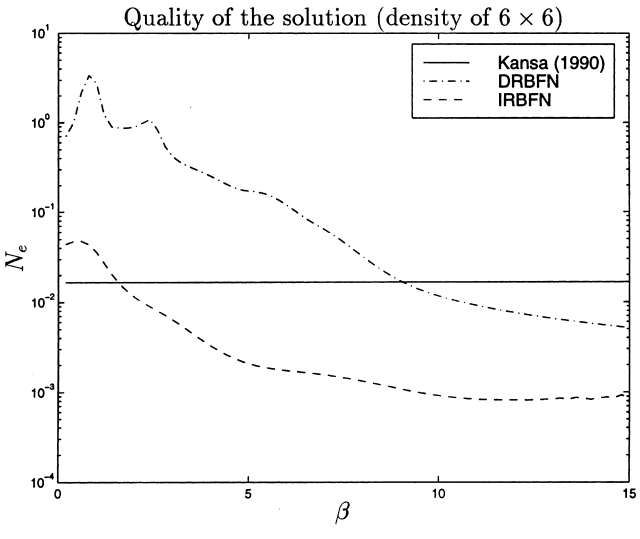

Fig. 13. Solution of $\nabla^2 u = (\lambda^2 + \mu^2)\exp(\lambda x_1 + \mu x_2)$: comparison of N_e between Kansa's, DRBFN and IRBFN results (random collocation points). Although Kansa's result (Kansa, 1990b) is not a function of β, it is shown at all values of β for comparison with the present results. The IRBFN method is much more accurate and stable over a wider range of β values.

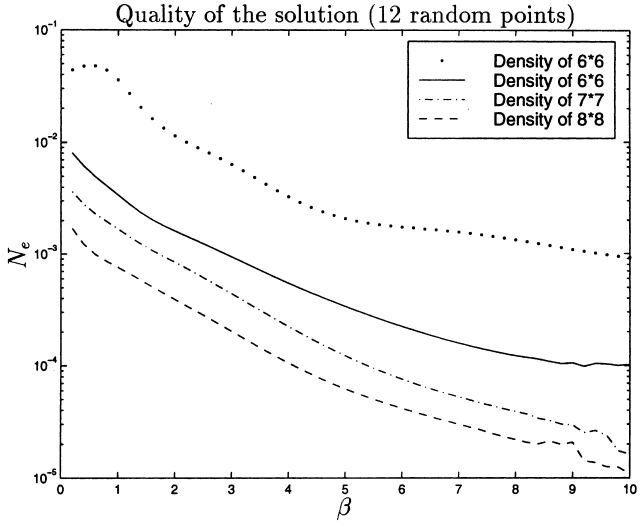

Fig. 14. Solution of $\nabla^2 u = (\lambda^2 + \mu^2)\exp(\lambda x_1 + \mu x_2)$: effect of centre density on N_e of IRBFN solution. The top curve corresponds to the case where the collocation points are random and disjoint from the set of centres. For the other cases, the legend describes the density of centres which together with the 12 random points constitute the set of collocation points. Thus, the set of centres is a subset of the collocation points.

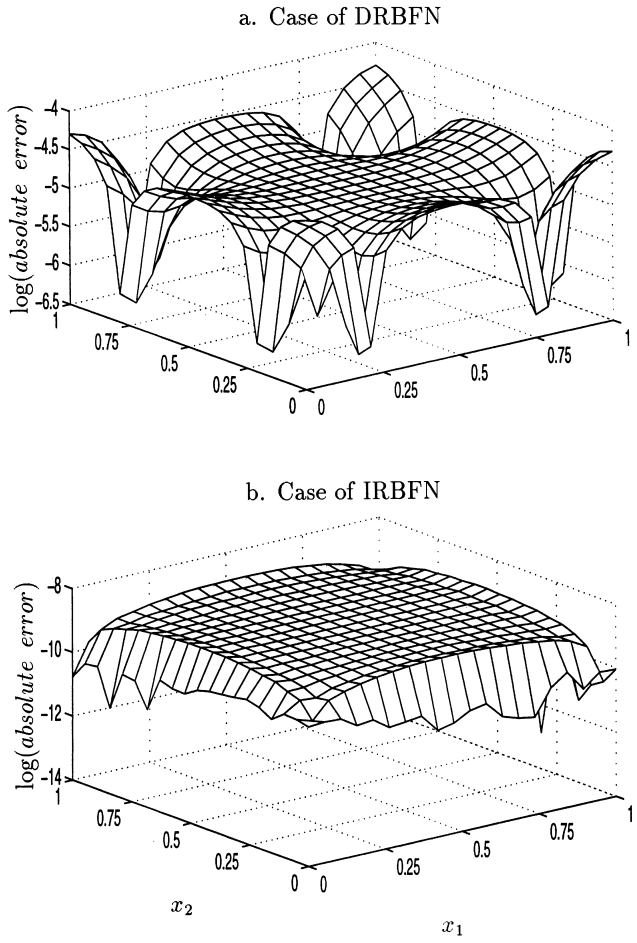

Fig. 15. Error in the solution of $\nabla^2 u = \sin(\pi x_1)\sin(\pi x_2)$: the numerical solutions are obtained with $\beta = 8$ in both DRBFN and IRBFN methods. Absolute error is used instead of relative error because the exact solution vanishes in certain regions of the domain. For the IRBFN, the error distribution is seen to be smoother and particularly the error is smaller along the boundary in comparison with the case of DRBFN.

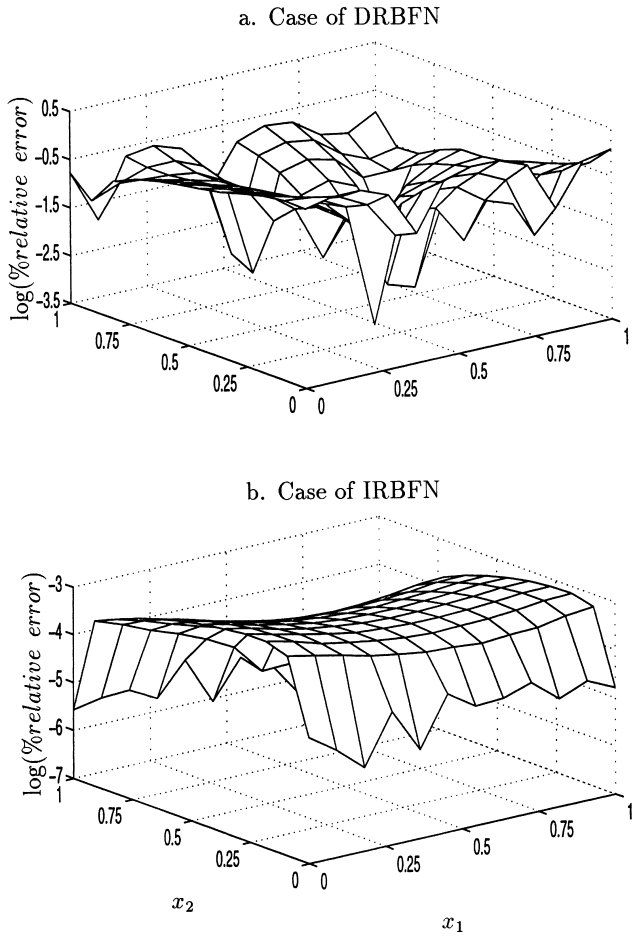

Fig. 16. Error in the solution of $\nabla^2 u = (\lambda^2 + \mu^2)\exp(\lambda x_1 + \mu x_2)$: the numerical solutions are obtained with $\beta = 8$ in both DRBFN and IRBFN methods. Percentage relative error is used in the plot. For the IRBFN, the error distribution is seen to be smoother and particularly the error is smaller along the boundary in comparison with the case of DRBFN.

consequence. Any inaccuracy in the assumed decomposition is usually magnified in the process of differentiation. In contrast, in the indirect approach the starting point is the decomposition of the highest derivatives present in the relevant DEs into some finite basis. Lower derivatives and finally the function itself are obtained by integration which has the property of damping out or at least containing any inherent accuracy in the assumed shape of the derivatives.

5.3. RBFN and FFNN algorithms

The architecture of the present RBFNs consists of just one hidden layer in contrast with the usually multi-layered FFNNs (of the back-propagation multi-layer perceptron (MLP) type). Furthermore, as pointed out by Haykin (1999, p. 293), the RBF network and the MLP have totally different arguments for their activation functions and the MLPs construct global approximations to nonlinear input–output mapping in contrast to local approximations in the case of the RBF networks. Consequently, in terms of 'training', the present RBFN methods result in a direct system of linear algebraic equations in the unknown weights whereas the FFNN method relies on an iterative algorithm (e.g. back-propagation) for determining the weights from some initial guess. Although the back-propagation algorithm for the FFNN method is computationally efficient, it suffers from the problem of local minima (Haykin, 1999, pp. 229–232) and therefore convergence to the true solution is a very difficult issue in the case of FFNNs (for example, the results of Dissanayake and Phan-Thien (1994) show that their FFNN method for the problem considered in Section 4.2.1. required 1000 iterations for convergence). The results presented show that the system of algebraic equations becomes ill-conditioned at large β, which is generally observed by other authors (Zerroukat et al., 1998). Too small β values correspond to very localised approximation and when β is too large the kernel becomes too flat, losing its ability to approximate; and therefore heuristically these

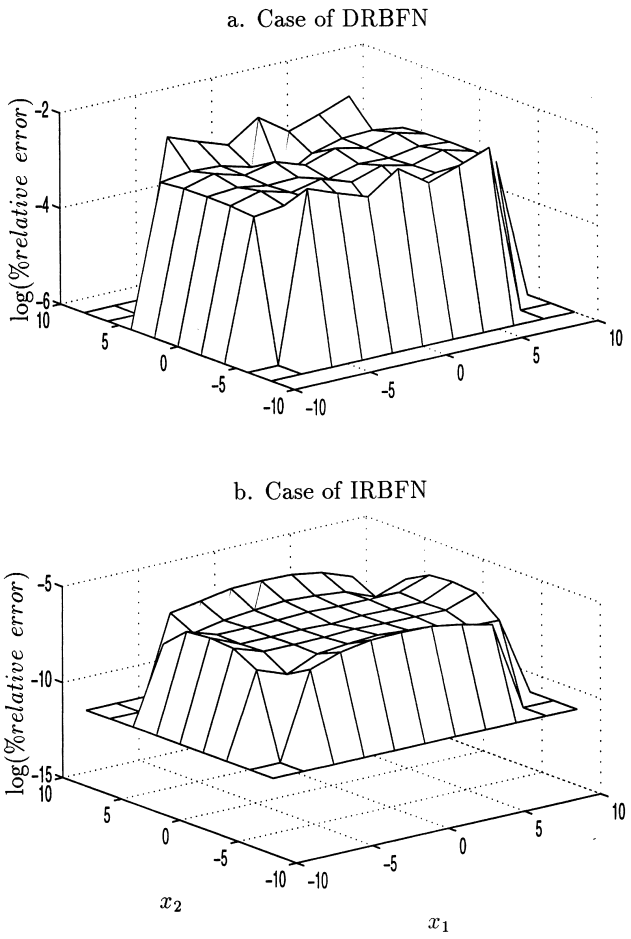

Fig. 17. Error in the solution of $\nabla^2 u = -2$: the numerical solutions are obtained with $\beta = 8$ in both DRBFN and IRBFN methods. Percentage relative error is used in the plot. Since the relevant domain here is elliptical, the error outside this domain is of no interest and arbitrarily set so that the relevant data can be seen in the plot. For the IRBFN, the error distribution is seen to be smoother. Although not shown, the error is smaller along the boundary in comparison with the case of DRBFN.

two extremes would result in sub-optimal network performance as observed experimentally in this work. Even when the value of β is in the 'optimal' range, some peculiarities in the N_e–β curve are observed. Despite this peculiar behaviour, the IRBFN results are better than the corresponding DRBFN results. The reason for this peculiar behaviour is not clear. However, the results shown here indicate that the sudden drop in the N_e–β curve (which represents a large improvement in accuracy) is due to a fortuitous combined effect of the centre density and the value of β. As the centre density increases, this sudden drop disappears and the accuracy is better over the whole of the 'optimal' range of β for the IRBFN method (Fig. 7b). The second peculiar behaviour is the oscillation observed at the tail of the N_e–β curves, which is also due to the combined effect of the centre density and the β value. However, the reason here is probably due to the numerical ill-conditioning of the system matrix. This explanation is supported by the observation that the system matrix size for the IRBFN method is slightly more than twice the size of the DRBFN system matrix, leading to an earlier occurrence of the oscillatory behaviour as β increases (Fig. 6a). These peculiar behaviours can be eliminated with increasing centre density, however, at the expense of the freedom to choose β in the sense that the range of optimal β value is narrowed. Thus the present RBFN algorithms are stable for a range of β value, provided that the centre density is sufficient, up to some critical value of β above which the system matrix becomes nearly singular. A theoretical relationship for the balance between centre density and the value for β requires formal investigation beyond the scope of the present work.

5.4. Other types of DEs

In principle, the present RBFN methods can be applied to nonlinear DEs. However, the nonlinearity necessitates an iterative solution procedure. For example, the coefficients in the DEs can be estimated using the data obtained from a previous iteration and become constant in the current iteration. For time-dependent problems, the method can be extended by treating the time derivatives with finite difference techniques and the resultant method is still mesh-free. The investigation associated with the above discussion is beyond the scope of the present paper and will be carried out in the near future.

6. Concluding remarks

New robust and highly accurate element-free procedures based on MQ RBFNs for solving DEs are discussed in this paper. The ease of preparation of input data (i.e. only discrete RBF centres and collocation points, which could be randomly or regularly distributed, are required), robustness of the methods (stability over a wide range of β) and high accuracy of the solution (norm of error of $O(1.0 \times 10^{-6})$ at least for the IRBFN method with optimal β value) make the method very attractive in comparison with conventional methods such as the FDM, FEM, FVM and BEM. Although the DRBFN method yields somewhat inferior norms of error of $O(1.0 \times 10^{-3})$ (with optimal β value), the mesh-free nature still makes the method attractive. Unlike the FFNN method, the present procedures are not iterative and hence more efficient. In all the tests carried out, with the same centre density, the indirect RBFN procedure achieves better accuracy than the direct RBFN procedure over a wide range of β and hence the choice of RBF width is less critical in the case of the IRBFN procedure. Furthermore, the IRBFN solution is improved significantly with increasing centre density, which indicates that the IRBFN procedure is well-behaved with respect to 'mesh refinement'. Both regularly shaped and irregularly shaped domains can be handled with ease. Further developments of the present procedures and their applications in solving large scale problems in several science and engineering

fields are under way and the results will be reported in the near future.

Acknowledgements

This work is supported by a Special USQ Research Grant (Grant No. 179-310) to Thanh Tran-Cong. Nam Mai-Duy is supported by a USQ scholarship. This support is gratefully acknowledged. The authors would like to thank the referees sincerely for their helpful suggestions to improve the original manuscript.

References

Brebbia, C. A., Telles, J. C. F., & Wrobel, L. C. (1984). *Boundary element techniques: theory and applications in engineering*, Berlin, Heidelberg: Springer-Verlag.

Cook, R. D., Malkus, D. S., & Plesha, M. E. (1989). *Concepts and applications of finite element analysis*, Toronto: John Wiley & Sons.

Dissanayake, M. W. M. G., & Phan-Thien, N. (1994). Neural-network-based approximations for solving partial differential equations. *Communications in Numerical Methods in Engineering*, *10*, 195–201.

Dongarra, J. J., Bunch, J. R., Moler, C. B., & Stewart, G. W. (1979). *LINPACK user's guide*, Philadelphia: SIAM.

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

Franke, R. (1982). Scattered data interpolation: tests of some methods. *Mathematics of Computation*, *38* (157), 181–200.

Haykin, S. (1999). *Neural networks: a comprehensive foundation*, New Jersey: Prentice-Hall.

Hughes, T. J. R. (1987). *The finite element method*, New Jersey: Prentice-Hall.

Kansa, E. J. (1990a). Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics—I. Surface approximations and partial derivative estimates. *Computers and Mathematics with Applications*, *19* (8/9), 127–145.

Kansa, E. J. (1990b). 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), 147–161.

Mai-Duy, N. & Tran-Cong, T. (1999). Approximation of function and its derivatives using radial basis function networks. Submitted to *Neural Networks*.

Moody, J., & Darken, C. J. (1989). Fast learning in networks of locally-tuned processing units. *Neural Computations*, *1*, 281–294.

Moridis, G. J., & Kansa, E. J. (1994). The Laplace transform multiquadrics method: a highly accurate scheme for the numerical solution of linear partial differential equations. *Journal of Applied Science and Computation*, *1* (2), 375–407.

Park, J., & Sandberg, I. W. (1993). Approximation and radial basis function networks. *Neural Computation*, *5*, 305–316.

Patankar, S. V. (1980). *Numerical heat transfer and fluid flow*, New York: McGraw-Hill.

Powell, M. J. D. (1988). Radial basis function approximations to polynomial. In D. F. Griffiths & G. A. Watson, *Numerical Analysis 1987 Proceedings* (pp. 223–241).

Press, W. H., Flannery, B. P., Teukolsky, S. A., & Vetterling, W. T. (1988). *Numerical recipes in C: the art of scientific computing*, Cambridge: Cambridge University Press.

Sharan, M., Kansa, E. J., & Gupta, S. (1997). Application of the multiquadric method for numerical solution of elliptic partial differential equation. *Journal of Applied Science and Computation*, *84*, 275–302.

Smith, G. D. (1978). *Numerical solution of partial differential equations: finite difference methods*, Oxford: Clarendon Press.

Tarwater, A. E. (1985). A parameter study of Hardy's multiquadric method for scattered data interpolation. Technical Report UCRL-563670. Lawrence Livemore National Laboratory.

Van Hulle, M. M. (1998). Kernel-based equiprobabilistic topographic map formation. *Neural Computation*, *10*, 1847–1871.

Zerroukat, M., Power, H., & Chen, C. S. (1998). A numerical method for heat transfer problems using collocation and radial basis functions. *International Journal for Numerical Methods in Engineeering*, *42*, 1263–1278.

Zienkiewicz, O. C., & Taylor, R. L. (1991). *The finite element method*. (4th ed.). London: McGraw-Hill.