A domain-type boundary-integral-equation method for two-dimensional biharmonic Dirichlet problem

N. Mai-Duy[*,a], T. Tran-Cong[a] and R.I. Tanner[b]

[a] Faculty of Engineering and Surveying,

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

[b] School of Aerospace, Mechanical and Mechatronic Engineering,

The University of Sydney, Sydney, NSW 2006, Australia

Submitted to *EABE*, 28-Mar-2006; Revised, 5-Jun-2006

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

Abstract This paper reports a new boundary-integral-equation method (BIEM) for numerically solving biharmonic problems with Dirichlet boundary conditions. For the solution of these problems in convex polygons, it was found that the accuracy of the conventional BIEM is significantly reduced, and spurious oscillatory behaviour is often observed in the boundary solutions especially for areas near corners (Mai-Duy N, Tanner RI. An effective high order interpolation scheme in BIEM for biharmonic boundary value problems. Eng Anal Bound Elem 2005; 29:210–23). In this study, a new treatment for these difficulties is proposed. The unknown functions in boundary integrals are approximated using a domain-type interpolation scheme rather than traditional boundary-type interpolation schemes. Two test problems are considered to validate the formulation and to demonstrate the attractiveness of the proposed method.

KEY WORDS: biharmonic Dirichlet problems, boundary integral equations, radial-basis-function networks, double boundary conditions

1 INTRODUCTION

Many engineering problems such as bending of thin plate and flow of viscous fluid can be formulated in terms of biharmonic equation/two Poisson equations. Owing to the existence of free-space fundamental solutions, one can obtain exact integral representations of these differential equations. The boundary-element method (BEM)/boundary-integral-equation method (BIEM) is a powerful numerical tool to solve the integral equations. For biharmonic problems, the conventional BIEM requires a pair of BIEs in order to deal with two unknown variables on the boundary. Since the governing differential equation is satisfied exactly, very accurate solutions are typically obtained. The interested reader is referred to, for example, References

[1-6] for the solution of thin-plate problems, and References [7,8] for the solution of fluid-flow problems.

The present work is concerned with biharmonic Dirichlet problems in non-smooth geometries. The field variable (normal deflection of the plate/streamfunction of the flow) and its first-order normal derivative are prescribed along the boundary of the domain. Such problems have been investigated using BIEMs since the late 1960s [9]. A non-iterative coupled approach was usually used to solve a pair of BIEs. For most problems reported, the variations of the boundary data were simple (e.g homogeneous boundary conditions). Recently, the accuracy of the BIEM was examined in detail for the case where the boundary data have complicated shapes/high-order variations [10]. In that work, the governing equation was chosen in the form of two Poisson equations. Linear elements, quadratic elements and radial-basis-function networks (RBFNs) were employed to represent the variations of the variables along the boundary. Numerical results showed that BIEMs work well in a smooth geometry, where one does not need to derive a computational boundary condition for the intermediate variable from the normal derivative boundary conditions. This feature can be seen as an advantage of BIEMs over other discretization methods such as finite-difference methods. However, in a non-smooth geometry, BIEMs produced fluctuations in the computed boundary solutions especially for areas near corners. In solving inverse biharmonic boundary-value problems, Zeb et al [11] also reported such phenomena. It was found that the use of an iterative decoupled approach can prevent spurious oscillatory behaviour, however it causes a very slow convergence with mesh refinement [10].

The objective of this paper is to present a new treatment for the above difficulties. The unknown functions in boundary integrals are approximated by utilizing a domain-type interpolation scheme rather than traditional boundary-type interpolation schemes. In this study, global high-order radial-basis-function networks

(RBFNs) are considered. The problem of implementing the multiple boundary conditions is resolved here by using integration to construct the RBF approximations (indirect/integrated RBFNs (IRBFNs)) [12,13]. Numerical results show that this approach produces not only smooth solutions but also rapid convergence. The proposed method is truly meshless for the case of homogeneous biharmonic equations.

The present formulation also appears to be suitable for solving nonlinear problems. From the literature, two well-known iterative techniques, namely a Picard-type and a Newton-type iteration schemes, are often used to handle the nonlinearity of the system matrix. The latter exhibits quadratic convergence, while the former is often slow. In the context of BIEMs, to use a Newton-type procedure, the interior equations must be brought into the system matrix [14,15]. In the present numerical procedure, the system matrix is composed of the BIE for interior points, and one can use a Newton-type iteration scheme for the solution of the resultant equations.

The remainder of the paper is organized as follows. The BIE analog of the biharmonic equation is given in Section 2. Section 3 presents a brief review of integrated RBFNs. The proposed method is described in Section 4, followed by several numerical examples in Section 5 to demonstrate the validity and attractiveness of the present implementation. Section 6 gives some concluding remarks.

2 BOUNDARY INTEGRAL EQUATION

Consider the fourth-order biharmonic equation

$$\nabla^4 v = b, \tag{1}$$

where b is a driving function, subject to Dirichlet boundary conditions v and $\partial v/\partial n$ (n – the outward normal to the boundary). This equation is one of the simplest forms of high-order differential equations, which is important in the modelling of many engineering applications.

The BIE analog of (1) [9,16,17] can be written as

$$C(\mathbf{y})v(\mathbf{y}) + \int_\Gamma \frac{\partial G^H(\mathbf{y},\mathbf{x})}{\partial n} v(\mathbf{x}) d\Gamma = \int_\Gamma G^H(\mathbf{y},\mathbf{x}) \frac{\partial v(\mathbf{x})}{\partial n} d\Gamma$$
$$- \int_\Gamma \left(\frac{\partial G^B(\mathbf{y},\mathbf{x})}{\partial n} u(\mathbf{x}) - G^B(\mathbf{y},\mathbf{x}) \frac{\partial u(\mathbf{x})}{\partial n} \right) d\Gamma - \int_\Omega G^B(\mathbf{y},\mathbf{x}) b(\mathbf{x}) d\Omega, \qquad (2)$$

where $\mathbf{y}$ is the source point, $\mathbf{x}$ the field point, Γ the piecewise smooth boundary of a domain Ω in $\mathbf{R}^2$, $C(\mathbf{y})$ the free term coefficient which is 1 if $\mathbf{y}$ is an interior point, 1/2 if $\mathbf{y}$ is a point on the smooth boundary and $\theta/(2\pi)$ if $\mathbf{y}$ is a corner (θ the internal angle of the corner in radians), u the new variable defined as $u = \nabla^2 v$, and G^H and G^B the harmonic and biharmonic fundamental solutions whose forms respectively are

$$G^H = \frac{1}{2\pi} \ln\left(\frac{1}{r}\right), \quad \frac{\partial G^H}{\partial n} = -\frac{1}{2\pi r}\frac{\partial r}{\partial n}, \qquad (3)$$
$$G^B = \frac{1}{8\pi} r^2 \left[\ln\left(\frac{1}{r}\right) + 1\right], \quad \frac{\partial G^B}{\partial n} = \frac{1}{8\pi} r \frac{\partial r}{\partial n} \left[1 + 2\ln\left(\frac{1}{r}\right)\right], \qquad (4)$$

in which $r = \|\mathbf{y} - \mathbf{x}\|$.

For viscous fluid flows, the variables v and u represent the streamfunction and vorticity, respectively while for thin plate bending problems, they are the deflection and "bending moment", respectively. The variable u has no physical meaning inside the plate domain, however it may be equal to the bending moment on the boundary.

3 INTEGRATED RADIAL-BASIS-FUNCTION NETWORKS

A function v, to be approximated, can be represented by an RBFN as follows

$$v(\mathbf{x}) = \sum_{i=1}^{m} w^{(i)} g^{(i)}(\mathbf{x}), \tag{5}$$

where $\mathbf{x}$ is the input vector, m the number of RBFs, $\{w^{(i)}\}_{i=1}^{m}$ the set of network weights to be found and $\{g^{(i)}(\mathbf{x})\}_{i=1}^{m}$ the set of RBFs. It has been proved that RBFNs have the property of universal approximation [18]. This study is concerned with multiquadrics (MQ) whose form is

$$g^{(i)}(\mathbf{x}) = \sqrt{(\mathbf{x} - \mathbf{c}^{(i)})^T(\mathbf{x} - \mathbf{c}^{(i)}) + a^{(i)2}}, \tag{6}$$

where $\mathbf{c}^{(i)}$ and $a^{(i)}$ are the centre and width of the ith RBF, respectively, and superscript T denotes the transpose of a vector. For simplicity, the set of centres is chosen to be the same as the set of collocation points, i.e $\{\mathbf{c}^{(i)}\}_{i=1}^{m} \equiv \{\mathbf{x}^{(i)}\}_{i=1}^{n}$ with $m = n$, while the widths are chosen according to

$$a^{(i)} = \beta d^{(i)}, \tag{7}$$

where β is a positive scalar and $d^{(i)}$ is the minimum of distances from the ith center to its neighbours.

The pth-order IRBFN, which is denoted by IRBFN-p, is defined as an RBFN approximation scheme in which RBFNs are employed to represent the pth-order derivatives and then integrated p times to obtain expressions for lower-order derivatives and the original function itself [19].

For the present two-dimensional case, the IRBFN-p scheme can be expressed as

$$\frac{\partial^p v(\mathbf{x})}{\partial x_j^p} = \sum_{i=1}^{m} w_{[x_j]}^{(i)} g^{(i)}(\mathbf{x}) = \sum_{i=1}^{m} w_{[x_j]}^{(i)} H_{[x_j]}^{[p](i)}(\mathbf{x}), \tag{8}$$

$$\frac{\partial^{p-1} v(\mathbf{x})}{\partial x_j^{p-1}} = \sum_{i=1}^{m} w_{[x_j]}^{(i)} H_{[x_j]}^{[p-1](i)}(\mathbf{x}) + C1_{[x_j]}(x_k), \tag{9}$$

$$\frac{\partial^{p-2} v(\mathbf{x})}{\partial x_j^{p-2}} = \sum_{i=1}^{m} w_{[x_j]}^{(i)} H_{[x_j]}^{[p-2](i)}(\mathbf{x}) + \frac{x_j}{1!} C1_{[x_j]}(x_k) + C2_{[x_j]}(x_k), \tag{10}$$

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

$$v_{[x_j]}(\mathbf{x}) = \sum_{i=1}^{m} w_{[x_j]}^{(i)} H_{[x_j]}^{[0](i)}(\mathbf{x}) + \frac{x_j^{p-1}}{(p-1)!} C1_{[x_j]}(x_k) + \cdots + Cp_{[x_j]}(x_k), \tag{11}$$

where subscript $[x_j]$ is used to denote the quantities that are associated with the process of integration with respect to the x_j direction; $C1_{[x_j]}(x_k), C2_{[x_j]}(x_k), \cdots$ and $Cp_{[x_j]}(x_k)$ $(k \neq j)$ are "integration constants"; and

$$\left\{ H_{[x_j]}^{[p-1](i)}(\mathbf{x}) \right\}_{i=1}^{m} = \left\{ \int H_{[x_j]}^{[p](i)}(\mathbf{x}) dx_j \right\}_{i=1}^{m}, \ \cdots, \ \left\{ H_{[x_j]}^{[0](i)}(\mathbf{x}) \right\}_{i=1}^{m} = \left\{ \int H_{[x_j]}^{[1](i)}(\mathbf{x}) dx_j \right\}_{i=1}^{m}$$

are new basis functions for the approximation of lower-order derivatives and the original function v, respectively.

It can be seen from (8)-(11) that, apart from RBFNs, there are new "coefficients" and new basis functions arising from the integration process. Each "coefficients" $Cl_{[x_j]}$ $(l = 1, \cdots, p)$ is a function of the variable x_k $(k \neq j)$, and they can also be approximated using IRBFNs. Here, IRBFN-2s are chosen to represent these functions

$$\begin{aligned}
\frac{d^2Cl_{[x_j]}(x_k)}{dx_k^2} &= \sum_{i=1}^{\overline{m}_j-2}\overline{w}^{(i)}\overline{g}^{(i)}(x_k) = \sum_{i=1}^{\overline{m}_j-2}\overline{w}^{(i)}\overline{H}^{[2](i)}(x_k), & (12)\\
\frac{dCl_{[x_j]}(x_k)}{dx_k} &= \sum_{i=1}^{\overline{m}_j-2}\overline{w}^{(i)}\overline{H}^{[1](i)}(x_k) + \overline{C}1 = \sum_{i=1}^{\overline{m}_j-1}\overline{w}^{(i)}\overline{H}^{[1](i)}(x_k), & (13)\\
Cl_{[x_j]}(x_k) &= \sum_{i=1}^{\overline{m}_j-2}\overline{w}^{(i)}\overline{H}^{[0](i)}(x_k) + \overline{C}1x_k + \overline{C}2 = \sum_{i=1}^{\overline{m}_j}\overline{w}^{(i)}\overline{H}^{[0](i)}(x_k), & (14)
\end{aligned}$$

where

$$\left\{\overline{H}^{[1](i)}(x_k)\right\}_{i=1}^{\overline{m}_j-2} = \left\{\int \overline{H}^{[2](i)}(x_k)dx_k\right\}_{i=1}^{\overline{m}_j-2},\ \left\{\overline{H}^{[0](i)}(x_k)\right\}_{i=1}^{\overline{m}_j-2} = \left\{\int \overline{H}^{[1](i)}(x_k)dx_k\right\}_{i=1}^{\overline{m}_j-2},$$

$\overline{H}^{[1](\overline{m}_j-1)}(x_k) = 1,\ \overline{H}^{[1](\overline{m}_j)}(x_k) = 0,\ \overline{H}^{[0](\overline{m}_j-1)}(x_k) = x_k,\ \overline{H}^{[0](\overline{m}_j)}(x_k) = 1,$

$\overline{w}^{(\overline{m}_j-1)} = \overline{C_1}$ and $\overline{w}^{(\overline{m}_j)} = \overline{C_2}.$

Making use of (12)-(14), expressions (8)-(11) can be rewritten in a compact form

$$\begin{aligned}
\frac{\partial^p v(\mathbf{x})}{\partial x_j^p} &= \sum_{i=1}^{m} w_{[x_j]}^{(i)} H_{[x_j]}^{[p](i)}(\mathbf{x}), & (15)\\
\frac{\partial^{p-1} v(\mathbf{x})}{\partial x_j^{p-1}} &= \sum_{i=1}^{m+\overline{m}_j} w_{[x_j]}^{(i)} H_{[x_j]}^{[p-1](i)}(\mathbf{x}), & (16)\\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
v_{[x_j]}(\mathbf{x}) &= \sum_{i=1}^{m+p\overline{m}_j} w_{[x_j]}^{(i)} H_{[x_j]}^{[0](i)}(\mathbf{x}), & (17)
\end{aligned}$$

where $(\overline{m}_j - 2)$ is the number of new RBF centres used for the approximation of integration constant functions.

4 THE PROPOSED BIEM

For traditional BIEMs, two BIEs are required for the solution procedure and they are often solved in a coupled manner. The variables in boundary integrals are approximated independently. Lagrange polynomials such as constant, linear or quadratic interpolating functions are usually employed to approximate the variations of u and $\partial u/\partial n$ along the boundary. Once all boundary unknown quantities are available, solutions at the interior points can be obtained by direct evaluation.

For the proposed BIEM, only one BIE, namely (2), is required. A domain-type interpolation scheme is employed to represent the variable v, from which approximations to the unknown variables u and $\partial u/\partial n$ are derived. The unknowns here are the values of the variable v at the interior points. Once the unknown vector is found, other solutions can be produced by means of IRBFNs. It should be noted that fourth-order problems require two boundary conditions at each boundary point. Hence, attention needs to be paid to the following issues

- To write the unknown functions u and $\partial u/\partial n$ in terms of nodal values of the variable v, and
- To implement the prescribed normal derivative boundary conditions $\partial v/\partial n$.

It can be seen that the unknown functions u and $\partial u/\partial n$ can be replaced by

$$u = \frac{\partial^2 v}{\partial x_1^2} + \frac{\partial^2 v}{\partial x_2^2}, \tag{18}$$

$$\frac{\partial u}{\partial n} = n_1 \frac{\partial u}{\partial x_1} + n_2 \frac{\partial u}{\partial x_2}, \tag{19}$$

$$= n_1 \left(\frac{\partial^3 v}{\partial x_1^3} + \frac{\partial^3 v}{\partial x_1 \partial x_2^2} \right) + n_2 \left(\frac{\partial^3 v}{\partial x_2 \partial x_1^2} + \frac{\partial^3 v}{\partial x_2^3} \right). \tag{20}$$

Making use of (18)-(20), the present solution procedure involves the approximation of the second- and third-order derivatives of the variable v.

From the prescribed boundary conditions v and $\partial v/\partial n$, the values of $\partial v/\partial x_1$ and $\partial v/\partial x_2$ at the boundary points can be easily obtained. Let n_{ip} be the number of interior points, and n_{bp1} and n_{bp2} be the numbers of boundary points having the boundary conditions $\partial v/\partial x_1$ and $\partial v/\partial x_2$, respectively. For example, consider a rectangular domain. If the domain is discretized using a rectangular grid formed by parallel n_{x_1} and n_{x_2} lines, one can have $n = n_{x_1} n_{x_2}$, $n_{ip} = (n_{x_1} - 2)(n_{x_2} - 2)$, $n_{bp1} = 2n_{x_2}$ and $n_{bp2} = 2n_{x_1}$.

In (15)-(17), the IRBFN expressions of a function and its derivatives are written in terms of network weights. In solving engineering problems, one would prefer to implement numerical methods in the physical space. To do so, expansion coefficients/network weights need to be converted into nodal space values of the variable. Owing to the presence of integration constants, the prescribed normal derivative boundary conditions can be imposed through the conversion process. This can be seen as an advantage of the integration-based formulation over the usual differentiation-based formulation. Consider the integration process with respect to the x_j direction. The conversion system can be formed by collocating the function (17) at the nodal points $\left\{\mathbf{x}^{(i)}\right\}_{i=1}^{n}$ and the derivative $\partial v/\partial x_j$ at the relevant boundary points $\left\{\mathbf{x}^{(i)}\right\}_{i=1}^{(n_{bpk},k\neq j)}$

$$
\begin{aligned}
v(\mathbf{x}^{(1)}) &= \sum_{i=1}^{m+p\overline{m}_j} w_{[x_j]}^{(i)} H_{[x_j]}^{[0](i)}(\mathbf{x}^{(1)}), \quad (\mathbf{x}\text{: nodal point}) \\
\cdots\cdots&\cdots\cdots\cdots\cdots\cdots\cdots\cdots \qquad (21)\\
v(\mathbf{x}^{(n)}) &= \sum_{i=1}^{m+p\overline{m}_j} w_{[x_j]}^{(i)} H_{[x_j]}^{[0](i)}(\mathbf{x}^{(n)}), \\
\frac{\partial v(\mathbf{x}^{(1)})}{\partial x_j} &= \sum_{i=1}^{m+(p-1)\overline{m}_j} w_{[x_j]}^{(i)} H_{[x_j]}^{[1](i)}(\mathbf{x}^{(1)}), \quad (\mathbf{x}\text{: boundary point}) \\
\cdots\cdots&\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
\frac{\partial v(\mathbf{x}^{(n_{bpk},k\neq j)})}{\partial x_j} &= \sum_{i=1}^{m+(p-1)\overline{m}_j} w_{[x_j]}^{(i)} H_{[x_j]}^{[1](i)}(\mathbf{x}^{(n_{bpk},k\neq j)}),
\end{aligned}
$$

or in a matrix-vector form

$$
\begin{pmatrix} \widehat{v} \\ \widehat{\frac{\partial v}{\partial x_j}} \end{pmatrix} = \mathcal{C}_{[x_j]} \widehat{w}_{[x_j]}, \tag{22}
$$

where $\mathcal{C}_{[x_j]}$ is the matrix of dimension $(n + n_{bpk}) \times (m + p\overline{m}_j)$ (here, $n = m$). In constructing the conversion matrix $\mathcal{C}_{[x_j]}$, the values of $H_{[x_j]}^{[1](i)}(\mathbf{x})$ are set to zeros for $i > m + (p-1)\overline{m}_j$.

Solving (22), one obtains

$$
\widehat{w}_{[x_j]} = \mathcal{C}_{[x_j]}^{-1} \begin{pmatrix} \widehat{v} \\ \widehat{\frac{\partial v}{\partial x_j}} \end{pmatrix} \tag{23}
$$

where $\mathcal{C}_{[x_j]}^{-1}$ is the Moore-Penrose pseudoinverse. It can be seen that (22) and (23) describe the relations between the physical space and the network-weight space.

Making use of (23), the variable v and its derivatives at the arbitrary point $\mathbf{x}$ can be computed by

$$\frac{\partial^l v(\mathbf{x})}{\partial x_j^l} = \left[H_{[x_j]}^{[l](1)}(\mathbf{x}), H_{[x_j]}^{[l](2)}(\mathbf{x}), \cdots, 0\right] \mathcal{C}_{[x_j]}^{-1} \left(\begin{array}{c} \widehat{v} \\ \widehat{\frac{\partial v}{\partial x_j}} \end{array} \right), \quad l = (p, p-1, \cdots, 1), \quad (24)$$

$$v_{[x_j]}(\mathbf{x}) = \left[H_{[x_j]}^{[0](1)}(\mathbf{x}), H_{[x_j]}^{[0](2)}(\mathbf{x}), \cdots, H_{[x_j]}^{[0](m+p\overline{m}_j)}(\mathbf{x})\right] \mathcal{C}_{[x_j]}^{-1} \left(\begin{array}{c} \widehat{v} \\ \widehat{\frac{\partial v}{\partial x_j}} \end{array} \right). \quad (25)$$

Expressions (24) and (25) can be reduced to

$$\frac{\partial^l v(\mathbf{x})}{\partial x_j^l} = \left[E_{[x_j]}^{[l](1)}(\mathbf{x}), E_{[x_j]}^{[l](2)}(\mathbf{x}), \cdots, E_{[x_j]}^{[l](n)}(\mathbf{x})\right] \widehat{v} + K_{[x_j]}^{[l]}(\mathbf{x}), \quad l = (p, p-1, \cdots, 1), \quad (26)$$

$$v_{[x_j]}(\mathbf{x}) = \left[E_{[x_j]}^{[0](1)}(\mathbf{x}), E_{[x_j]}^{[0](2)}(\mathbf{x}), \cdots, E_{[x_j]}^{[0](n)}(\mathbf{x})\right] \widehat{v} + K_{[x_j]}^{[0]}(\mathbf{x}), \quad (27)$$

where $E_{[x_j]}^{[.](.)}(\mathbf{x})$ and $K_{[x_j]}^{[.]}(\mathbf{x})$ are known functions. At the collocation points, the values of the functions $v_{[x_1]}$ and $v_{[x_2]}$ are forced to be exactly the same (they are unknowns to be found); at an arbitrary point, the value of the variable v can be computed as

$$v(\mathbf{x}) = \frac{1}{2}\left(v_{[x_1]}(\mathbf{x}) + v_{[x_2]}(\mathbf{x})\right). \quad (28)$$

Partial derivatives of third-order in the case of using IRBFN-2 and mixed partial derivatives of third-order in the case of using IRBFN-4 can be computed according to the following relations

$$\frac{\partial^3 v}{\partial x_j^3} = \frac{\partial}{\partial x_j}\left(\frac{\partial^2 v}{\partial x_j^2}\right), \quad (29)$$

$$\frac{\partial^3 v}{\partial x_j x_k^2} = \frac{\partial}{\partial x_j}\left(\frac{\partial^2 v}{\partial x_k^2}\right), \quad (30)$$

where the functions to be differentiated on the RHS of (29) and (30) are the derivative functions rather than the original function. Using (26) with $l = (1, 2)$ and regarding v ($l = 1$) as a generic function, one can also express the above third-order partial derivatives in terms of nodal variable values.

Expressions for the unknown functions u and $\partial u/\partial n$ can now be written in terms of nodal values of the variable v over the whole domain. Since the variable v is prescribed along the boundary, one only needs to find the values of the variable v at the interior points. The objective here is to generate a number of algebraic equations equal to the number of unknowns. This can be achieved by applying the BIE (2) at the interior points. It should be emphasized that the present equation system consists of the interior equations only, thus completely avoiding all difficulties in numerical computation caused by the singularity of boundary integrals. All boundary integrals involved are regular, and numerical integration can be conducted using standard Gaussian quadrature. Once the linear algebraic system is set up, the solution can be determined using Gauss elimination.

In the context of BIEMs, the use of a Newton-type iteration algorithm was reported to be imperative for the solution of complex nonlinear problems [15]. It can be seen that the present formulation is based on the BIE for interior points. Consequently, one can use a Newton-type procedure to handle the nonlinearity of the system.

5 NUMERICAL RESULTS

In the following test cases, the width of the ith RBF is simply chosen to be the minimum distance from the ith centre to neighbouring centres ($a^{(i)} = \beta d^{(i)} = d^{(i)}$). The accuracy of a numerical solution produced by an approximation scheme is measured by means of the discrete relative L_2 norm defined as

$$N_e = \sqrt{\frac{\sum_{i=1}^{n_t} \left[f_e(\mathbf{x}^{(i)}) - f(\mathbf{x}^{(i)})\right]^2}{\sum_{i=1}^{n_t} f_e(\mathbf{x}^{(i)})^2}}, \qquad (31)$$

where n_t is the number of test points, $\mathbf{x}^{(i)}$ is the ith test point, f and f_e are the calculated and exact solutions, respectively. Another important measure is the convergence rate of the solution with respect to the refinement of spatial discretization

$$N_e(h) \approx \gamma h^{\alpha} = O(h^{\alpha}) \tag{32}$$

in which h is the spacing (mesh size), and α and γ are the exponential model's parameters. Given a set of observations, these parameters can be found by the general linear least squares technique.

5.1 Linear problem

A test problem chosen here is taken from Reference [11]. Consider the homogeneous biharmonic equation in the square domain $-2 \leq x_1, x_2 \leq 2$, subject to Dirichlet boundary conditions v and $\partial v/\partial n$. The exact solutions of this problem are given by

$$v = \frac{1}{2}x_1 \left(\sin x_1 \cosh x_2 - \cos x_1 \sinh x_2\right), \tag{33}$$

$$\frac{\partial v}{\partial x_1} = \frac{1}{2}\left(\sin x_1 \cosh x_2 - \cos x_1 \sinh x_2\right) + \frac{1}{2}x_1 \left(\cos x_1 \cosh x_2 + \sin x_1 \sinh x_2\right), \tag{34}$$

$$\frac{\partial v}{\partial x_2} = \frac{1}{2}x_1 \left(\sin x_1 \sinh x_2 - \cos x_1 \cosh x_2\right), \tag{35}$$

$$u = \cos x_1 \cosh x_2 + \sin x_1 \sinh x_2, \tag{36}$$

$$\frac{\partial u}{\partial x_1} = \cos x_1 \sinh x_2 - \sin x_1 \cosh x_2, \tag{37}$$

$$\frac{\partial u}{\partial x_2} = \sin x_1 \cosh x_2 + \cos x_1 \sinh x_2. \tag{38}$$

The geometry is non-smooth and the boundary data v and $\partial v/\partial n$ have complicated shapes (high-order variations). Two versions of the proposed method, namely IRBFN-2 and IRBFN-4, are employed. Their convergence behaviours are investigated through a number of uniform densities, $5 \times 5, 7 \times 7, \cdots, 31 \times 31$. Results concerning the discrete relative L_2 norm are shown in Table 1, together with those obtained by a linear-BIEM. For the linear BIEM, discontinuous elements (double nodes) are used to deal with the problem of multi-values of the normal derivative at the corner. Figures 1 and 2 display the computed solutions obtained by the IRBFN-4 and linear BIEMs.

For the conventional linear BIEM, large fluctuations occur on the boundary especially for areas close to corners (Figure 1). In contrast to the boundary solutions, good accuracy is obtained for the interior solutions v and u, probably due to the fact that spurious oscillations may cancel out each other during the numerical evaluation of boundary integrals. The method gives very low convergence rates for all solutions (Table 1).

For the proposed method, the two IRBFN versions perform well, especially for the higher-order one. The computed solutions are all smooth and they are in good agreement with the exact solutions (Figures 2). Accurate results and high convergence rates are obtained. For example, using IRBFN-4, the relative L_2 error at the highest density of 31×31 for the solution v is 2.60×10^{-5}, and the convergence obtained is of $O(h^{3.00})$ (Table 1). The order of the accuracy of the method cannot be given theoretically at this stage; further studies are needed.

Results obtained show that the spurious oscillation behaviour on the boundary is overcome with the use of a domain-type interpolation scheme. When compared with the iterative decoupled approach that also has the capability to prevent the noise [10], the proposed method gives a much faster convergence with respect to the refinement of spatial discretization.

5.2 Nonlinear problem

As mentioned earlier, the present system of algebraic equations is comprised of the interior equations rather than the boundary equations. As a result, the nonlinear term is brought into the system matrix and a Newton-type iteration algorithm can be used for its solution. This feature is demonstrated here by considering the solution to the following nonlinear driving function

$$b = v^2 \left[\left(x_1^2 + x_2^2 \right)^2 + 8x_1x_2 + 4 \right] \exp(-x_1x_2). \tag{39}$$

The exact solution can be verified to be

$$v_e = \exp(x_1x_2). \tag{40}$$

The driving function is computed in the form of (39) and hence, it contains a nonlinear term, namely v^2. Consider the unit square domain $0 \leq x_1, x_2 \leq 1$. One can easily use (40) to obtain the values of the boundary conditions v and $\partial v/\partial n$. The volume integral is generated by the nonlinear driving function b. Here, this integral is computed using the cell integration approach. The domain of interest is represented by a set of triangular elements. The values of the driving function at the integration points are obtained by means of IRBFNs. Regular integrals are evaluated using two-dimensional Gaussian quadrature, while weakly-singular integrals are calculated using the transformation to polar coordinates [20]. The trust-region method, see for example [21], is applied here to handle the nonlinearity of the resultant equations.

To study the convergence behaviour, a number of uniform meshes, namely $3 \times 3, 5 \times 5, \cdots$ and 15×15, are employed. It takes only a few iterations (two or three) to get a convergent solution. Table 2 presents the results obtained by the IRBFN-4 scheme, indicating a rapid improvement in accuracy with mesh refinement. The convergence rate is of $O(h^{3.12})$ and the discrete relative L_2 norm at the finest mesh of 15×15 is 5.11×10^{-7}.

6 CONCLUDING REMARKS

A domain-type interpolation scheme based on integrated radial-basis-function networks is introduced into BIEs to represent the field variable for numerically solving biharmonic Dirichlet problems. Only one BIE is required for the solution procedure, and there is no need to apply this equation at the boundary points, thereby avoiding difficulties in numerical computation caused by the singularity of the boundary integrals. Derivative boundary conditions are implemented through the process of constructing the RBF approximations. Nonlinear systems of algebraic equations obtained can be solved effectively with trust-region methods. Numerical results show that spurious oscillatory behaviour in the computed boundary solutions due to the effect of corners is overcome. The proposed method attains a significant improvement in accuracy and convergence rate over conventional BIEMs. The application of the method to complex viscous flows will be reported in future work.

ACKNOWLEDGEMENTS

This research was supported by the Australian Research Council. Dr. N. Mai-Duy is the recipient of an Australian Research Council Post-Doctoral Fellowship. The authors would like to thank the referees for their helpful comments.

REFERENCES

1. Jaswon MA, Maiti M. An integral equation formulation of plate bending problems. Journal of Engineering Mathematics 1968;2:83–93.

2. Bezine GP. Boundary integral formulation for plate flexure with an arbitrary boundary conditions. Mechanics Research Communications 1978;5:197–206.

3. Stern M. A general boundary integral formulation for the numerical solution of plates bending problems. International Journal of Solids and Structures 1979;15:769–82.

4. Karami G, Zarrinchang J, Foroughi B. Analytic treatment of boundary integrals in direct boundary element analysis of plate bending problems. International Journal for Numerical Methods in Engineering 1994;37:2409–27.

5. Wearing JL, Bettahar O. The analysis of plate bending problems using the regular direct boundary element method. Engineering Analysis with Boundary Elements 1995;16(3):261–71.

6. Paris F, de Leon S. Thin plates by the boundary element method by means of two Poisson equations. Engineering Analysis with Boundary Elements 1996;17(2):111–22.

7. Ingham DB, Kelmanson MA. In: Brebbia CA, Orszag SA, editors. Lecture Notes in Engineering (Vol. 7): Bounday Integral Equation Analyses of Singular, Potential, and Biharmonic Problems. Berlin: Springer-Verlag, 1984.

8. Camp CV, Gipson GS. A boundary element method for viscous flows at low Reynolds numbers. Engineering Analysis with Boundary Elements 1989;6(3):144–51.

9. Brebbia CA, Telles JCF, Wrobel LC. Boundary Element Techniques Theory and Applications in Engineering. Berlin: Springer-Velag; 1984.

10. Mai-Duy N, Tanner RI. An effective high order interpolation scheme in BIEM for biharmonic boundary value problems. Engineering Analysis with Boundary Elements 2005;29:210–23.

11. Zeb A, Elliott L, Ingham DB, Lesnic D. A comparison of different methods to solve inverse biharmonic boundary value problems. International Journal for Numerical Methods in Engineering 1999;45:1791–1806.

12. Mai-Duy N, Tanner RI. Solving high order partial differential equations with radial basis function networks. International Journal for Numerical Methods in Engineering 2005;63:1636–54.

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

14. Tosaka N, Onishi K. Boundary integral equation formulations for steady Navier-Stokes equations using the Stokes fundamental solutions. Engineering Analysis 1985;2(3):128–32.

15. Dargush GF, Banerjee PK. A boundary element method for steady incompressible thermoviscous flow. International Journal for Numerical Methods in Engineering 1991;31:1605–26.

16. Nowak AJ, Neves AC. The Multiple Reciprocity Boundary Element Method. Southampton: Computational Mechanics Publications; 1994.

17. Pozrikidis C. A Practical Guide to Boundary Element Methods with the Software Library BEMLIB. Boca Raton: CRC Press; 2002.

18. Park J, Sandberg IW. Universal approximation using radial basis function networks. Neural Computation 1991;3:246–57.

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

20. Ramachandran PA. Boundary Element Methods in Transport Phenomena. Glasgow: Computational Mechanics Publications; 1994.

21. Nocedal J, Wright SJ. Numerical Optimization (Springer Series in Operations Research). New York: Springer Verlag; 1999.

Table 1: Linear problem: accuracy and convergence rate. Notice that $a(-b)$ means $a \times 10^{-b}$ and h is the spacing (mesh size).

	Linear			IRBFN-2			IRBFN-4		
	$Ne(v)$	$Ne(u)$		$Ne(v)$	$Ne(u)$		$Ne(v)$	$Ne(u)$	
$n_{x_1} = n_{x_2}$		Boundary	Interior		Boundary	Interior		Boundary	Interior
5	1.1720(-2)	4.6349(0)	1.7829(-1)	1.6088(-2)	1.2453(-1)	2.4408(-1)	1.7484(-2)	1.0325(-1)	2.6475(-1)
7	1.2453(-2)	4.3553(0)	1.2632(-1)	3.5504(-3)	3.6317(-2)	3.9176(-2)	2.6457(-3)	2.1247(-2)	3.1643(-2)
9	1.0236(-2)	4.0599(0)	9.0515(-2)	2.2881(-3)	1.9263(-2)	1.6400(-2)	1.0782(-3)	9.8104(-3)	1.0121(-2)
11	8.4913(-3)	3.7946(0)	7.0285(-2)	1.5054(-3)	1.2177(-2)	8.4594(-3)	7.5027(-4)	5.6267(-3)	4.6423(-3)
13	7.2201(-3)	3.5625(0)	5.7887(-2)	1.0191(-3)	8.5855(-3)	5.0264(-3)	5.0061(-4)	3.5808(-3)	2.5094(-3)
15	6.2889(-3)	3.3602(0)	4.9755(-2)	7.1189(-4)	6.4935(-3)	3.3126(-3)	3.3372(-4)	2.4541(-3)	1.5060(-3)
17	5.6002(-3)	3.1830(0)	4.4192(-2)	5.1374(-4)	5.1478(-3)	2.3524(-3)	2.2568(-4)	1.7799(-3)	9.7711(-4)
19	5.0898(-3)	3.0262(0)	4.0332(-2)	3.8248(-4)	4.2196(-3)	1.7626(-3)	1.5571(-4)	1.3494(-3)	6.7476(-4)
21	4.7163(-3)	2.8860(0)	3.7703(-2)	2.9293(-4)	3.5462(-3)	1.3734(-3)	1.0983(-4)	1.0595(-3)	4.9015(-4)
23	4.4533(-3)	2.7592(0)	3.6049(-2)	2.3002(-4)	3.0384(-3)	1.1022(-3)	7.9198(-5)	8.5548(-4)	3.7082(-4)
25	4.2849(-3)	2.6431(0)	3.5238(-2)	1.8457(-4)	2.6436(-3)	9.0522(-4)	5.8317(-5)	7.0667(-4)	2.8978(-4)
27	4.2031(-3)	2.5356(0)	3.5226(-2)	1.5089(-4)	2.3291(-3)	7.5729(-4)	4.3779(-5)	5.9459(-4)	2.3265(-4)
29	4.2069(-3)	2.4348(0)	3.6038(-2)	1.2535(-4)	2.0735(-3)	6.4322(-4)	3.3459(-5)	5.0821(-4)	1.9050(-4)
31	4.3029(-3)	2.3393(0)	3.7778(-2)	1.0559(-4)	1.8621(-3)	5.5333(-4)	2.6012(-5)	4.4080(-4)	1.5844(-4)
	$O(h^{0.64})$	$O(h^{0.35})$	$O(h^{0.84})$	$O(h^{2.37})$	$O(h^{1.98})$	$O(h^{2.87})$	$O(h^{3.00})$	$O(h^{2.58})$	$O(h^{3.52})$

Table 2: Non-linear problem: accuracy and convergence rate. Notice that $a(-b)$ means $a \times 10^{-b}$ and h is the spacing (mesh size).

$n_{x_1} = n_{x_2}$	$Ne(v)$
3	2.2711(-4)
5	2.8471(-5)
7	7.8375(-6)
9	3.4314(-6)
11	1.6681(-6)
13	8.7782(-7)
15	5.1051(-7)
	$O(h^{3.12})$

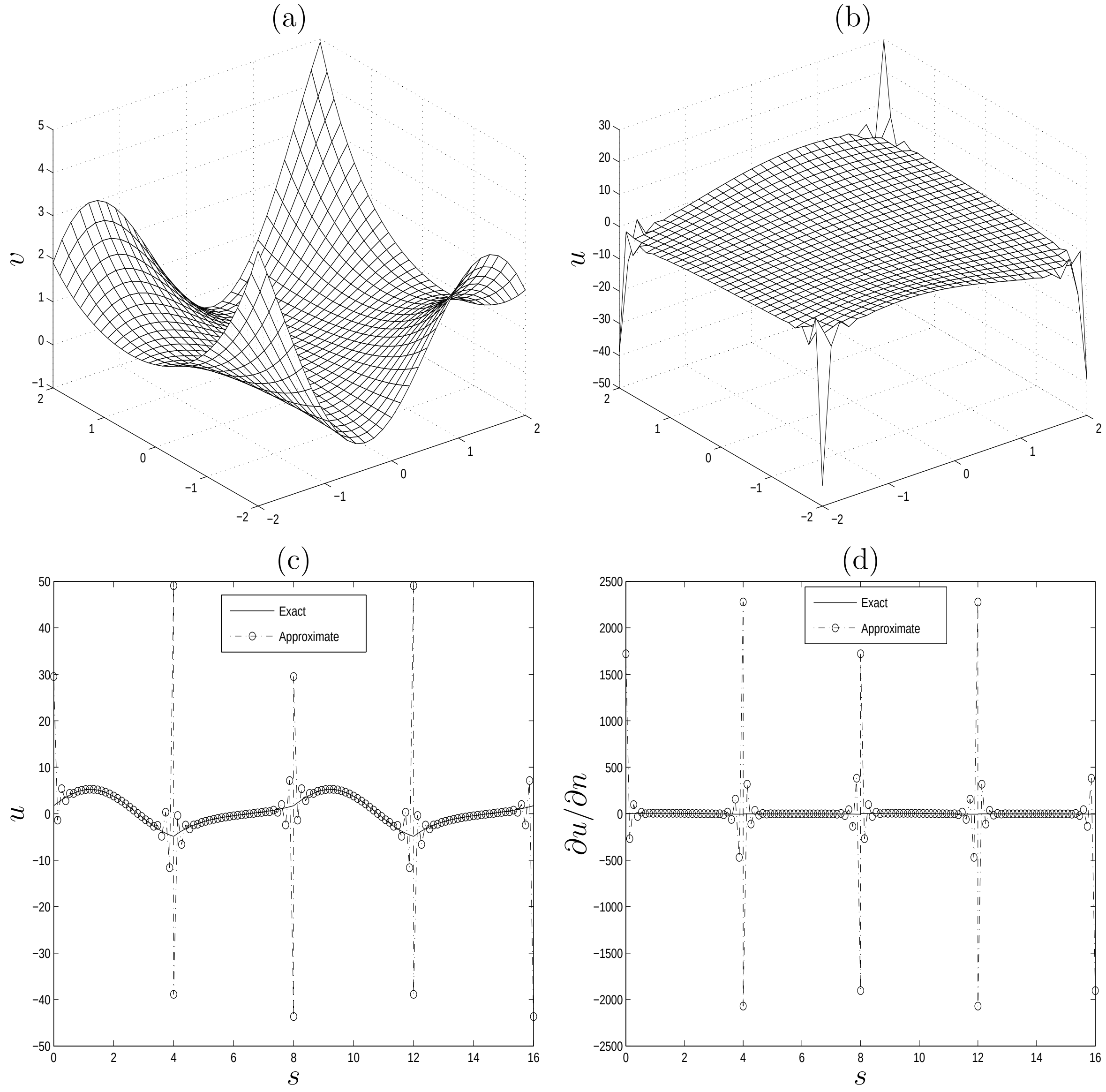

Figure 1: Linear problem, 31×31: approximate solutions obtained by linear-BIEM. Figures (a) and (b) present the variations of v and u over the whole domain, while figures (c) and (d) show the variations of u and its gradient along the boundary s. Large fluctuations appear in the computed boundary solutions u and $\partial u/\partial n$.

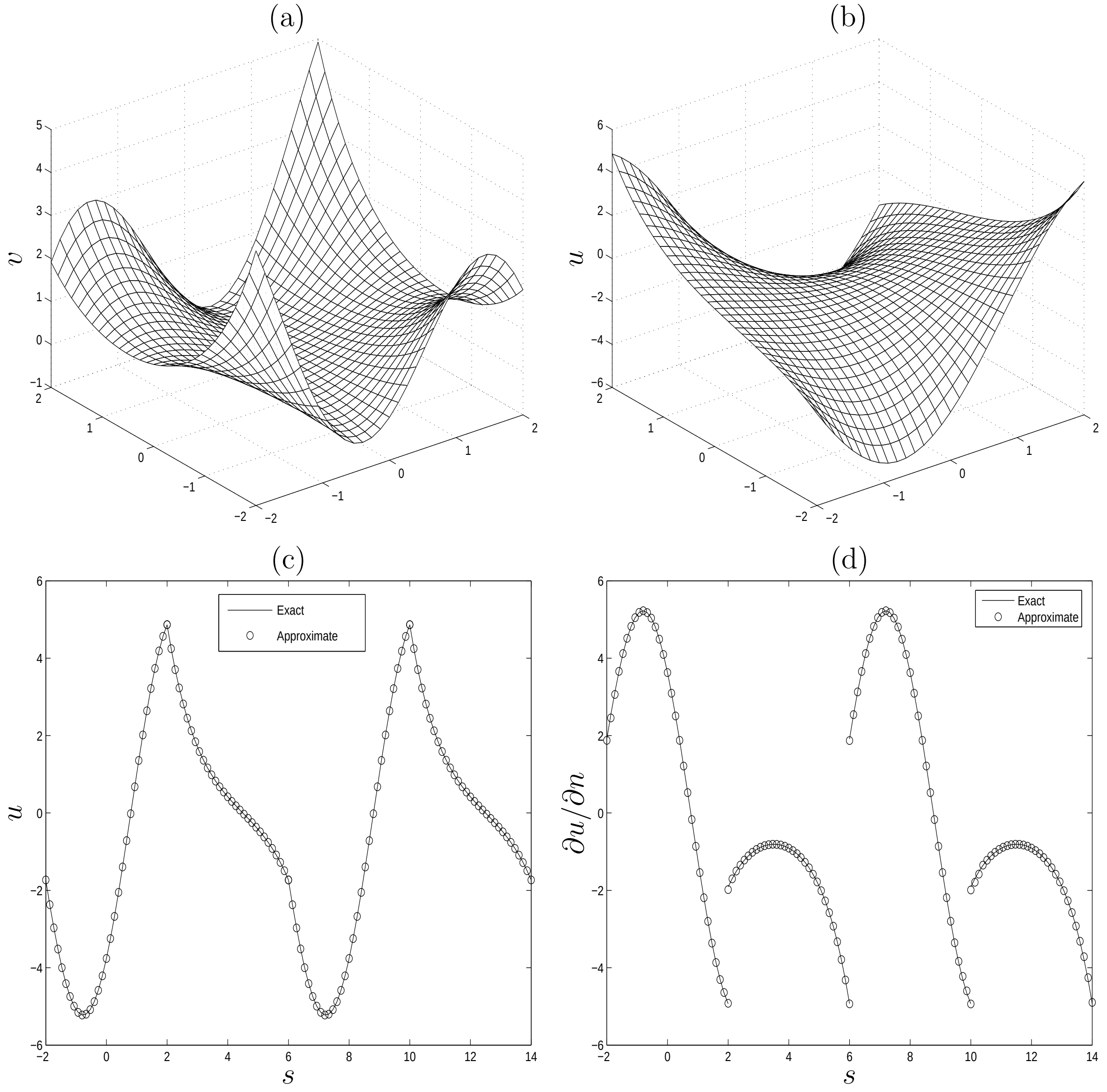

Figure 2: Linear problem, IRBFN-4, 31 × 31: approximate solutions obtained by the proposed method. Figures (a) and (b) present the variations of v and u over the whole domain, while figures (c) and (d) show the variations of u and its gradient along the boundary s. All computed solutions are smooth and in good agreement with the exact solutions.