Computational Mechanics 19 (1997) 264–270 © Springer-Verlag 1997

On boundary conditions in the element-free Galerkin method

Y. X. Mukherjee, S. Mukherjee

Abstract Accurate imposition of essential boundary conditions in the Element Free Galerkin (EFG) method often presents difficulties because the Moving Least Squares (MLS) interpolants, used in this method, lack the delta function property of the usual finite element or boundary element method shape functions. A simple and logical strategy, for alleviating the above problem, is proposed in this paper. A discrete norm is typically minimized in the EFG method in order to obtain certain variable coefficients. The strategy proposed in this work involves a new definition of this discrete norm. This new strategy works very well in all the numerical examples, for 2-D potential problems, that are presented here. In addition to the discussion of boundary conditions, some recommendations are also made in this paper regarding strategies for refinements in order to improve the accuracy of numerical solutions from the EFG method.

1 Introduction

Significant progress has been made in recent years, (see, for example, Shephard et al. 1995, or Oden, 1995), in creating structured meshes for three-dimensional (3-D) Finite Element Method (FEM) analysis of solids and structures. Shephard et al. (1995), for example, report on a mesh with two million tetrahedra! However, it is generally recognized that successful meshing of 3-D bodies of complex shape can be difficult, time consuming and expensive. For linear analysis, for example, mesh generation is often much more time consuming than the assembly and solution of the FEM equations. Therefore, there is considerable interest in exploring methods of numerical analysis that avoid or greatly simplify this meshing task.

Nayroles et al. (1992) have proposed an interesting method which they call the Diffuse Element Method (DEM). They propose a nodal interpolation scheme in which interpolants are fit to nodal values by a least-squares approximation scheme. These interpolants are called Moving Least Squares (MLS) interpolants. Nayroles et al. (1992) have coupled this interpolation scheme with Galerkin methods to produce the Diffuse Element Method (DEM). The conventional Finite Element Method mesh is not necessary in this approach. They (i.e. Nayroles et al.) present applications of the DEM in 2-D potential theory and in linear elasticity.

In a series of papers, Belytschko and his co-workers have popularized a different implementation of the Diffuse Element Method. They have called their approach the Element Free Galerkin (EFG) method and have applied it to a large variety of (two-dimensional) problems such as potential theory and linear elasticity (Belytschko et al. 1994 a), fracture mechanics with crack growth (Belytschko et al. 1994 b, 1995 a, Lu et al. 1994), dynamic fracture (Belytschko et al. 1995 b, Belytschko and Tabbara, 1996) and plate bending (Krysl and Belytschko, 1995). In their work, Belytschko and his co-workers have coupled the MLS interpolants with a variational (weak) form of the equilibrium equations of the appropriate boundary value problem. They have introduced a background cell structure in order to carry out integration by numerical quadrature.

Very recently, Mukherjee and Mukherjee (1997) have proposed a combination of MLS interpolants with Boundary Integral Equations (BIE). This new method, called the Boundary Node Method (BNM), retains the meshless attribute of the EFG and the dimensionality advantage of the BIE, and only needs nodes (points) to be specified on the bounding surface of a body. Numerical results for 2-D potential theory, obtained by the BNM, are most encouraging (Mukherjee and Mukherjee, 1997).

One of the problems with MLS interpolants is that, in general, they lack the delta function property of the usual BEM or FEM shape functions, in that

$$\Phi_I(\mathbf{x}_J) \neq \delta_{IJ} \tag{1}$$

where Φ_I is the I_{th} shape function evaluated at a nodal point $\mathbf{x}_J$ and δ_{IJ} is the Kronecker delta. This complicates the imposition of essential boundary conditions in the EFG method. Belytschko and his co-workers have employed various strategies such as collocation (Belytschko and Tabbara, 1996), Lagrange multipliers (Belytschko et al., 1994 a) and use of tractions as Lagrange multipliers (Lu et al., 1994) in order to impose essential boundary conditions in the EFG method.

The main contribution of the present paper is the proposal of a simple, logical and effective strategy for the

Communicated by T. Belytschko, 4 September 1996

Y. X. Mukherjee
DeHan Engineering Numerics, 95 Brown Road, Box 1016,
Ithaca, NY 14850, USA

S. Mukherjee
Department of Theoretical and Applied Mechanics,
Cornell University, Ithaca, NY 14853, USA

Correspondence to: S. Mukherjee

imposition of essential boundary conditions in the EFG method. A discrete weighted norm is typically minimized in the EFG method in order to obtain certain variable coefficients. In the present work, a new definition of this norm is employed, together with the use of fluxes as Lagrange multipliers in potential problems. Numerical results for problems in 2-D potential theory, presented in this paper, are most encouraging. Extensions of this idea, to 3-D problems and to linear elasticity, are obvious.

Another issue discussed in this paper is the role of nodes and integration cells in h-refinement of the EFG method. It is pointed out that the role of integration cells in the EFG is different from that of elements in the FEM.

This paper begins with a brief summary of MLS interpolants and the EFG method for problems in 2-D potential theory. The rest of the paper is devoted to issues related to imposition of boundary conditions and h-refinement in the EFG method.

2 MLS interpolation scheme

Following Belytschko et al. (1994 a), one writes

$$u(\mathbf{x}) = \sum_{j=1}^{m} p_j(\mathbf{x}) a_j(\mathbf{x}) = \mathbf{p}^{\mathrm{T}}(\mathbf{x})\mathbf{a}(\mathbf{x}) \tag{2}$$

where, in this work on 2-D problems, the intrinsic polynomial basis $\mathbf{p}(\mathbf{x})$ is quadratic, i.e.

$$\mathbf{p}^{\mathrm{T}}(\mathbf{x}) = [1, x, y, x^2, y^2, xy] \ , \qquad m = 6 \tag{3}$$

The coefficients in equation (2) are obtained by minimizing a weighted, discrete L_2 norm defined as

$$J_{\mathrm{new}} = \sum_{I=1}^{n} w(d_I)\left[\mathbf{p}^{\mathrm{T}}(\mathbf{x}_I)\mathbf{a}(\mathbf{x}) - \hat{u}_I\right]^2 \tag{4a}$$

where n is the number of nodes in the neighborhood of an evaluation point E (usually a Gauss point), with coordinates $\mathbf{x}$, for which the weight functions $w(d_I) \neq 0$, and $\hat{u}_I$ are approximations to the values $u(\mathbf{x}_I)$ at the nodes $\mathbf{x}_I$. This neighborhood of E is called its domain of dependence $\mathscr{D}_E$. Specific weight functions are discussed later in the next section of this paper. A key contribution of the present work is the above new definition of the discrete norm J_{new}. In previous work on the EFG method, the expression

$$J_{\mathrm{old}} = \sum_{I=1}^{n} w(d_I)\left[\mathbf{p}^{\mathrm{T}}(\mathbf{x}_I)\mathbf{a}(\mathbf{x}) - u_I\right]^2 \tag{4b}$$

where u_I are the nodal values, has been used. This replacement of u_I by $\hat{u}_I$ in the definition of J has been carried out in order to accurately satisfy essential boundary conditions at the boundary nodes. This matter is discussed in detail later in this paper.

Using the stationarity of J in equation (4a) one finally obtains

$$u(\mathbf{x}) = \sum_{I=1}^{n} \Phi_I(\mathbf{x})\hat{u}_I \tag{5a}$$

The corresponding result starting from Eq. (4b) is

$$u(\mathbf{x}) = \sum_{I=1}^{n} \Phi_I(\mathbf{x})u_I \tag{5b}$$

In the above, the shape functions $\Phi_I(\mathbf{x})$ are

$$\Phi_I(\mathbf{x}) = \sum_{j=1}^{m} p_j(\mathbf{x}) C_{jI}(\mathbf{x}) \tag{6}$$

with

$$\mathbf{C}(\mathbf{x}) = \mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x}) \tag{7}$$

$$\mathbf{A}(\mathbf{x}) = \sum_{I=1}^{n} w_I(\mathbf{x})\mathbf{p}(\mathbf{x}_I)\mathbf{p}^{\mathrm{T}}(\mathbf{x}_I) \tag{8}$$

$$w_I(\mathbf{x}) = w(d_I) \tag{9}$$

and

$$\mathbf{B}(\mathbf{x}) = [w_1(\mathbf{x})\mathbf{p}(\mathbf{x}_1), w_2(\mathbf{x})\mathbf{p}(\mathbf{x}_2), \cdots, w_n(\mathbf{x})\mathbf{p}(\mathbf{x}_n)] \tag{10}$$

It is important to observe that if J_{old} is used, Eqs. (1) and (5b) imply that, in general, the quantity $u(\mathbf{x}_J)$ is not equal to the nodal value u_J. The distinction between these two quantities is important.

The weight functions must be chosen such that the matrix $\mathbf{A}$ in Eq. (8) is invertible.

The partial derivatives of $\Phi_I(\mathbf{x})$ are obtained as

$$\Phi_{I,i} = \sum_{j=1}^{m} \{p_{j,i} C_{jI} + p_j (C_{,i})_{jI}\} \tag{11}$$

where

$$(C_{,i})_{jI} = \{(\mathbf{A}^{-1})_{,i}\mathbf{B} + \mathbf{A}^{-1}\mathbf{B}_{,i}\}_{jI} \tag{12}$$

$$(\mathbf{A}^{-1})_{,i} = -\mathbf{A}^{-1}\mathbf{A}_{,i}\mathbf{A}^{-1} \tag{13}$$

and $_{,i} = \frac{\partial}{\partial x_i}$.

3 Weight functions

The weight function chosen in this work is the exponential function (Belytschko et al., 1994 a)

$$w(d_I) = \begin{cases} \frac{e^{-(d_I/c)^2} - e^{-(\hat{d}_I/c)^2}}{1 - e^{-(\hat{d}_I/c)^2}} \ , & d_I \leq \hat{d}_I \\ 0 \ , & d_I > \hat{d}_I \end{cases} \tag{14}$$

Here, $d_I = \|\mathbf{x} - \mathbf{x}_I\|$, (the Euclidean distance between $\mathbf{x}$ and $\mathbf{x}_I$), $\hat{d}_I$ is the size of the support for the weight function w_I and c is a constant that controls the relative weights.

The weight function w_I determines the range of influence $\Re_I$ of the node $\mathbf{x}_I$. According to Eq. (14), $\Re_I$ for a node I is a circle of radius $\hat{d}_I$, centered at $\mathbf{x}_I$. The domain of dependence $\mathscr{D}_E$ of an evaluation point E (with coordinates $\mathbf{x}$) is the union of n overlapping circles, each centered at $\mathbf{x}_I$ and of radius $\hat{d}_I$ (Fig. 1). Only in the special case where all the $\hat{d}_I$ are equal to $\hat{d}$ does $\mathscr{D}_E$ become equivalent to a circle of radius $\hat{d}$, centered at E.

4 Coupling of MLS interpolants with variational equations

The partial differential equation to be solved is Laplace's equation in 2-D

$$\nabla^2 u = 0 \qquad \text{in } B \tag{15}$$

with boundary conditions

$$u = \bar{u} \quad \text{on } \partial B_u \ , \qquad q = \bar{q} \quad \text{on } \partial B_q \tag{16}$$

where $\partial B = \partial B_u \cup \partial B_q$ is the boundary of B and $q \equiv \frac{\partial u}{\partial n}$ is the normal derivative u. Here, $\mathbf{n}$ is the usual unit outward normal to ∂B at any point on it.

Starting from

$$0 = \int_B u_{,ii} \delta u \mathrm{d}A + \int_{\partial B_u} \delta q (u - \bar{u}) \mathrm{d}s \tag{17}$$

integration by parts leads to the weak (variational) from used in this work.

$$\begin{aligned} \int_B u_{,i} \delta u_{,i} \mathrm{d}A - \int_{\partial B_u} q \delta u \mathrm{d}s - \int_{\partial B_u} u \delta q \mathrm{d}s \\ = \int_{\partial B_q} \bar{q} \delta u \mathrm{d}s - \int_{\partial B_u} \bar{u} \delta q \mathrm{d}s \end{aligned} \tag{18}$$

where dA and ds are area and length elements in B and on ∂B, respectively, and δu is the usual first variation of u. Equation (17) is equivalent to the one in Zienkiewicz and Taylor (1994-p248, Eq. (9.141)) and the exact potential theory counterpart of the variational equation for elasticity in Lu et al. (1994, Eq. (25)). Note that the flux q is used as the Lagrange multiplier in Eqs. (17–18).

One now derives an interpolation for q on ∂B from Eq. (5a) as

$$q(\mathbf{x}) = \sum_{I=1}^{n} \Psi_I(\mathbf{x}) \hat{u}_I \quad , \qquad \mathbf{x} \in \partial B \tag{19}$$

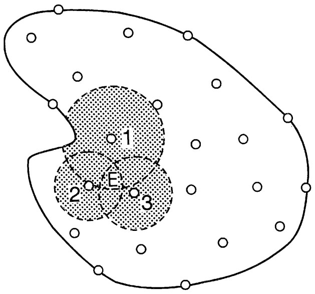


Fig. 1. Domain of dependence of an evaluation point E

where

$$\Psi_I = \frac{\partial \Phi_I}{\partial x} n_x + \frac{\partial \Phi_I}{\partial y} n_y \tag{20}$$

with the spatial derivatives of Φ being obtained from Eq. (11) and n_x, n_y the Cartesian components of the unit normal $\mathbf{n}$.

Substitution of Eqs. (5a) and (19) into the weak form (18) leads to the discretized system of linear equations

$$([\mathbf{K}] - [\mathbf{G}] - [\mathbf{G}^{\mathrm{T}}])\{\hat{\mathbf{u}}\} = \{\mathbf{f}\} - \{\mathbf{g}\} \tag{21a}$$

Please note that if one starts from Eq.(4b) instead of (4a), one gets

$$([\mathbf{K}] - [\mathbf{G}] - [\mathbf{G}^{\mathrm{T}}])\{\mathbf{u}\} = \{\mathbf{f}\} - \{\mathbf{g}\} \tag{21b}$$

which is identical to Eq. (21a) except for the replacement of $\{\hat{\mathbf{u}}\}$ by $\{\mathbf{u}\}$. Equation (21b) is the exact potential theory counterpart of the elasticity Eq. (27a) in Lu et al. (1994).

In the above,

$$K_{IJ} = \int_B (\Phi_{xI} \Phi_{xJ} + \Phi_{yI} \Phi_{yJ}) \mathrm{d}A \tag{22}$$

$$G_{IJ} = \int_{\partial B_u} \Psi_I \Phi_J \mathrm{d}s \tag{23}$$

$$f_I = \int_{\partial B_q} \bar{q} \Phi_I \mathrm{d}s \tag{24}$$

$$g_I = \int_{\partial B_u} \bar{u} \Psi_I \mathrm{d}s \tag{25}$$

where $\Phi_{xI} = \frac{\partial \Phi_I}{\partial x}$, $\Phi_{yI} = \frac{\partial \Phi_I}{\partial y}$ and the vectors $\{\hat{\mathbf{u}}\}$ and $\{\mathbf{u}\}$ contain their nodal values $\hat{\mathbf{u}}_I$ and $\mathbf{u}_I$, respectively, at all the nodes in B.

Assembly is performed at each quadrature point and contributions are gathered at each node. A simple cell structure is used in order to carry out the integrations above by Gaussian quadrature. While the number of cells influences the accuracy of the numerical solution, it has no relationship with the number of degrees of freedom in the EFG method. This issue is discussed further later in this paper.

5 Essential boundary conditions

As mentioned before, the lack of the delta function property of Φ_I (Eq. (1)) causes problems in imposing essential boundary conditions in the EFG method. Three different strategies for imposing essential boundary conditions in elasticity have been proposed by Belytschko and his co-workers. Their potential theory counterparts are described below.

Strategy 1:
Collocation with old definition of J (Eq. (4b))

In this case, Lagrange multipliers are not used and Eq. (21b) reduces to

$$[\mathbf{K}]\{\mathbf{u}\} = \{\mathbf{f}\} \tag{26}$$

Essential boundary conditions take the form

$$u_I = \bar{u}_I = \bar{u}(\mathbf{x}_I)\ , \qquad \mathbf{x}_I \text{ on } \partial B_u \tag{27}$$

Strategy 2:
Fluxes as Lagrange multipliers with old definition of J (Eq. (4b))

This strategy uses fluxes as Lagrange multipliers to enforce (see Eq. (17)).

$$u(\mathbf{x}) = \bar{u}(\mathbf{x})\ , \qquad \mathbf{x} \text{ on } \partial B_u \tag{28}$$

This is enforced in discretized form as (see Eq. (5b))

$$u(\mathbf{x}_J) = \sum_{I=1}^{n} \Phi_I(\mathbf{x}_J) u_I = \bar{u}(\mathbf{x}_J)\ , \qquad \mathbf{x}_J \text{ on } \partial B_u \tag{29}$$

The resulting final equation is (21b) which is solved for $\{\mathbf{u}\}$ which contains u_I at all the nodes in B. The values of u_I at all the nodes are accepted as the numerical solution of the problem.

Strategy 3:
Lagrange multipliers with old definition of J (Eq. (4b))

This strategy is essentially the same as strategy 2, except that general Lagrange multipliers are used here.

New Strategy Proposed in this Paper:
Fluxes as Lagrange multipliers with new definition of J (Eq. (4a))

The new strategy proposed here is very simple, logical and easy to enforce. Equation (28) is enforced in discretized form as (see Eq. (5a))

$$u(\mathbf{x}_J) = \sum_{I=1}^{n} \Phi_I(\mathbf{x}_J) \hat{u}_I = \bar{u}(\mathbf{x}_J), \qquad \mathbf{x}_J \text{ on } \partial B_u \tag{30}$$

Next, one solves the resulting Eq. (21a) for $\{\hat{\mathbf{u}}\}$ which contains $\hat{u}_I$. Finally, one sets (see Eq. (5a))

$$u(\mathbf{x}_J) = \sum_{I=1}^{n} \Phi_I(\mathbf{x}_J) \hat{u}_I \tag{31}$$

at all the nodes $\mathbf{x}_J,\ J = 1, 2, \cdots, N_N$, where N_N is the total number of nodes in B. The values of $u(\mathbf{x}_J)$ at all the nodes are accepted as the numerical solution of the problem.

6 Numerical results

Boundary Conditions Strategy 2 (fluxes as Lagrange multipliers with the old definition of J) and the new strategy (fluxes as Lagrange multipliers with the new definition of J) are compared in the numerical examples below. Henceforth, these strategies are called 'with J_{old}' and 'with J_{new}', respectively. Lu et al. (1994) (see their Table 2) state that Strategy 1 (collocation with old definition of J) can lead to significant errors in linear elasticity problems. Hence, Strategy 1 is not considered further in the present paper.

The discretization (4 × 4 cells, 9 × 9 nodes) and numerical examples considered here are summarized in Fig. 2. In all cases, Laplace's equation is solved in a unit square. The nodal arrangement in each cell used in these examples is the same as that in Lu et al. (1994) Fig. 2a. Also, in all cases, $\hat{d}$ = 0.32, c = 0.48 and a 6 × 6 array of Gauss points is used for numerical integration.

A global error measure is defined as

$$\in = \frac{1}{|u|_{\max}} \sqrt{\frac{1}{N_N} \sum_{i=1}^{N_N} (u_i^{(e)} - u_i^{(n)})^2} \tag{32}$$

where the superscripts (e) and (n) refer to the exact and numerical solutions, respectively, and N_N is the total number of nodes. Also, $\in_1$ is the error with J_{new} and $\in_2$ with J_{old}, respectively.

In all the numerical results shown in Figs. 3–7, a solid line is the exact solution, '×–×–×' is the solution with J_{new} and 'O O O' is the solution with J_{old}.

Example 1 is a patch test for the new formulation – a Dirichlet problem for the solution

$$u = x + y \tag{33}$$

The numerical results for $u = (x, y)$ with J_{new} are shown in Fig. 3. The global error for this example is $\in$ = 0.51%.

It has been proved by Belytschko et al. (1996) that MLS interpolants are consistent and any function in a basis can be reproduced exactly. Here, the intrinsic basis $\mathbf{p}(\mathbf{x})$ is quadratic while the test solution $u(\mathbf{x})$ is linear. Thus, in this case, one has

$$u^{(h)}(\mathbf{x}) = u^{(e)}(\mathbf{x}) \tag{34}$$

where $u^{(h)}(\mathbf{x})$ is the approximation to the exact solution $u^{(e)}(\mathbf{x})$. Unfortunately, however, the MLS interpolants lack the delta function property, so that, if one writes, as in Belytschko et al. (1994)

$$u^{(h)}(\mathbf{x}) = \sum_{I=1}^{n} \Phi_I(\mathbf{x}) u_I \tag{35}$$

one has, in general,

$$u^{(e)}(\mathbf{x}_J) = u^{(h)}(\mathbf{x}_J) = \sum_{I=1}^{n} \Phi_I(\mathbf{x}_J) u_I \neq u_J \tag{36}$$

The last inequality in equation (36) means that one can get errors in the numerical solutions for nodal values u_j even for a linear field. A small error in the nodal values persists even with the new formulation presented in this paper.

Examples 2–5 are all concerned with the exact cubic solution of Laplace's equation

$$u = -x^3 - y^3 + 3xy^2 + 3x^2y \tag{37}$$

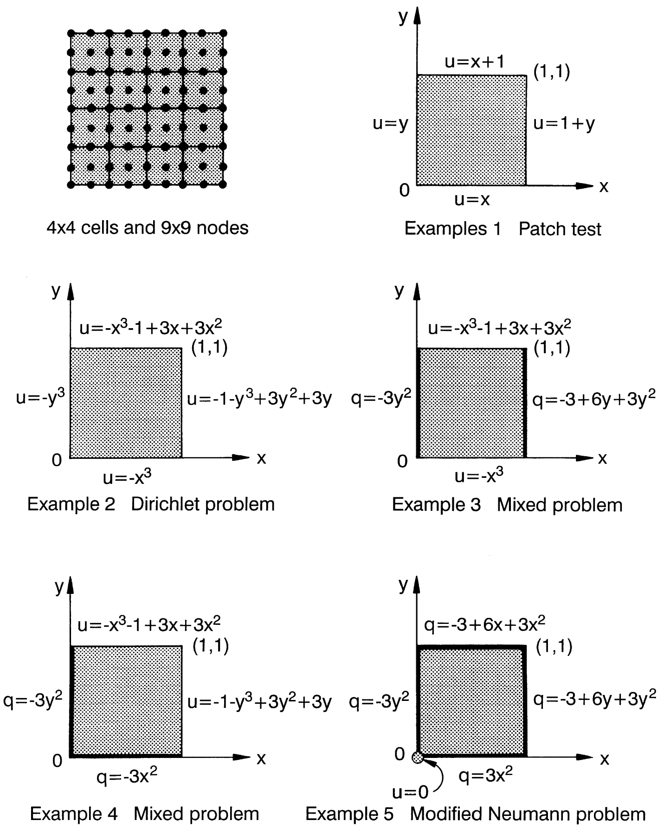


Fig. 2. Geometry and boundary conditions for various examples

Example 2 is a Dirichlet problem, 3 and 4 are mixed and 5 is a modified Neumann problem. The actual boundary conditions imposed on the boundary of a unit square, in each case, are shown in Fig. 2.

Numerical results for these examples are shown in Figs. 4–7 and the global errors are summarized in Table 1. In all cases, substantial improvement in accuracy is achieved with J_{new}. The strategy with J_{old} does quite poorly in the Dirichlet problem in which all the boundary conditions are of the essential type. The errors, from both strategies, are considerably less for example 4 (Fig. 6) compared to those for example 3 (Fig. 5). This is somewhat surprising since both examples 3 and 4 are mixed pro-

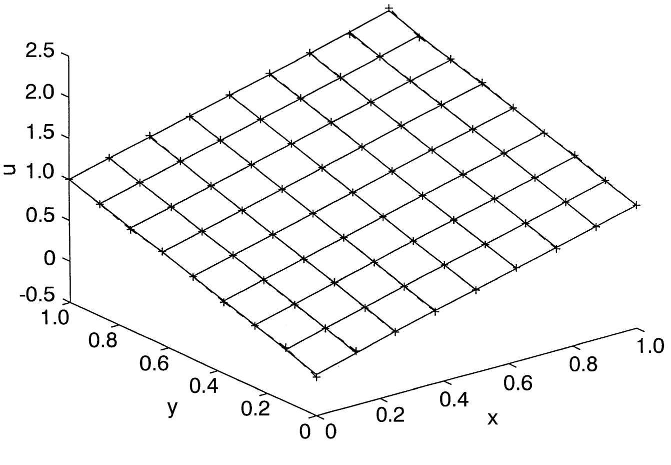


Fig. 3. The solution $u(x,y)$ for the patch test. (Example 1)

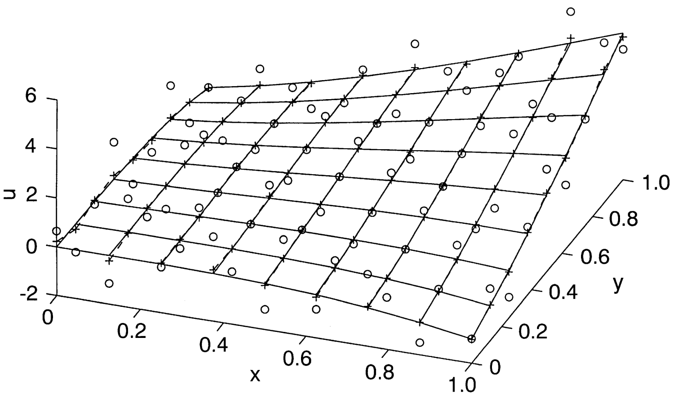


Fig. 4. The solution $u(x,y)$ for a cubic Dirichlet problem. (Example 2)

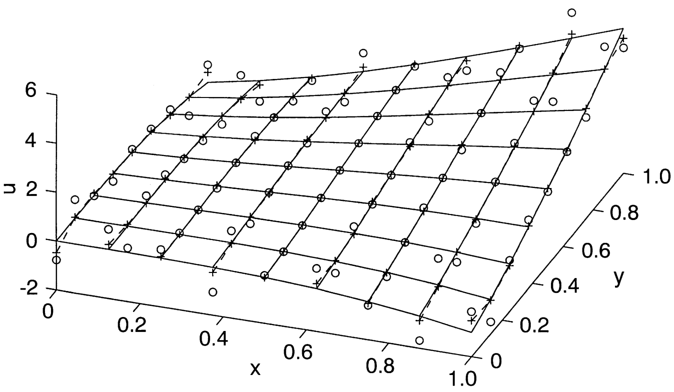


Fig. 5. The solution $u(x,y)$ for a cubic mixed problem. (Example 3)

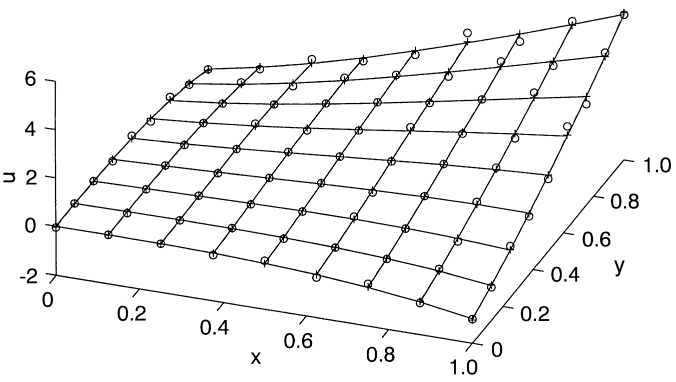


Fig. 6. The solution $u(x,y)$ for a cubic mixed problem. (Example 4)

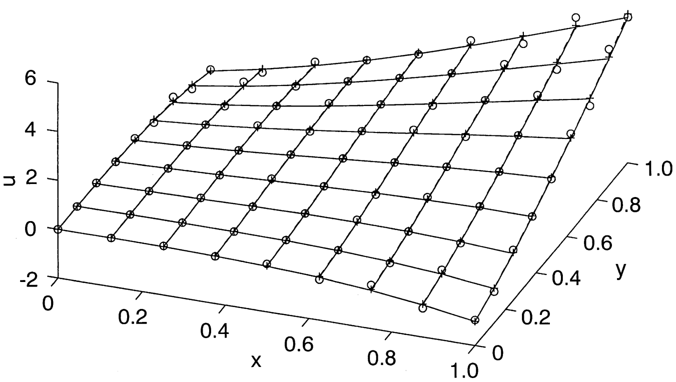
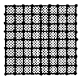
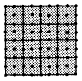
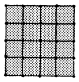
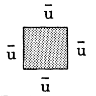
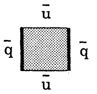
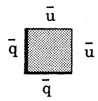
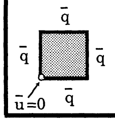


Fig. 7. The solution $u(x,y)$ for a modified Neumann problem. (Example 5).

Table 1. Impact of boundary condition enforcement strategy on global errors 4 × 4 cells, 9 × 9 nodes, $\in_1$: with J_{new}, $\in_2$: with J_{old}

Example	$\in_1$%	$\in_2$%
2	1.85	17.97
3	3.05	11.81
4	0.5	3.06
5	0.9	3.01

Table 2. Dependence of global errors on arrangements of cells and nodes. $\in_1$ with J_{new}, $\in_2$: with J_{old}

<table>
<tr><td rowspan="2">nodes
& cells
boundary
conditions</td><td colspan="2">
8×8 cells, 9×9 nodes
$\hat{d}$=0.32, c=0.48</td><td colspan="2">
4×4 cells, 9×9 nodes
$\hat{d}$=0.32, c=0.48</td><td colspan="2">
4×4 cells, 5×5 nodes
$\hat{d}$=0.64, c=0.96</td></tr>
<tr><td>$\in_1$</td><td>$\in_2$</td><td>$\in_1$</td><td>$\in_2$</td><td>$\in_1$</td><td>$\in_2$</td></tr>
<tr><td> </td><td>0.33%</td><td>1.46%</td><td>1.85%</td><td>17.97%</td><td>1.08%</td><td>7.88%</td></tr>
<tr><td> </td><td>0.59%</td><td>2.17%</td><td>3.05%</td><td>11.81%</td><td>0.64%</td><td>2.68%</td></tr>
<tr><td> </td><td>0.41%</td><td>1.43%</td><td>0.5%</td><td>3.06%</td><td>0.71%</td><td>3.03%</td></tr>
<tr><td>

$\bar{q}$
$\bar{q}$
$\bar{q}$
$\bar{u}$=0
$\bar{q}$

</td><td>0.29%</td><td>0.59%</td><td>0.9%</td><td>3.01%</td><td>0.51%</td><td>1.48%</td></tr>
</table>

blems of the same type in that two edges have essential and two natural boundary conditions. Also, while the global errors in both cases are acceptably small, the new strategy with J_{new} actually does better in the Dirichlet problem than in the mixed problem in example 3 (see Figures 4, 5 and Table 1)! Finally, as expected, both strategies do well in the modified Neumann problem – example 5 (Fig. 7). In this example, $u = 0$ (when using J_{old}) or $\hat{u} = 0$ (when using J_{new}) is imposed at the origin by collocation.

The above numerical results demonstrate the considerable value of the proposed new strategy, with J_{new}, for the satisfaction of boundary conditions in the EFG method.

Nodes and Cells The following remarks pertain to issues related to h refinement in the EFG method.

First, it is very important to realize that the cells in the EFG method are quite different from finite elements in the FEM. Cells are used to carry out numerical integration by Gaussian quadrature and influence the accuracy of a numerical solution. They can be very simple and need not satisfy the usual compatibility requirements of finite elements. Cells can be easily generated inside a code and can be easily refined in a localized region. Finally, the number of degrees of freedom in the EFG depends solely on the number of nodes N_N and has no relationship to the number of cells N_c.

The issue of h refinement is discussed next. One can try to improve a numerical solution either by increasing the number of nodes per cell or by increasing the number of cells while keeping the ratio

$$r_1 = \frac{\#\text{ of nodes}}{\#\text{ of cells}} = \frac{N_N}{N_c} \qquad \text{nearly fixed} \ .$$

The first strategy, while quite appealing, does not seem to work. This can be seen by comparing columns 3 and 4 with 5 and 6 in Table 2 which gives global errors for the same examples 2–5 in Fig. 2, for different discretizations. Note that the errors increase in most cases in columns 3 and 4, compared to those in columns 5 and 6, respectively, even though $\hat{d}$ in columns 3 and 4 is taken to be half of that in columns 5 and 6 in order to keep the number of nodes in $\mathscr{D}_E$ (the domain of dependence of a Gauss point) roughly the same in both these cases. The same trend was observed with the Boundary Node Method (Mukherjee and Mukherjee, 1997, Table 4).

A better approach is the second one where one increases the number of cells while keeping the ratios r_1 and

$$r_2 = \hat{d}/a$$

(where a is the side of a typical square cell) roughly unchanged. This is the situation when one compares columns 1 and 2 with columns 5 and 6, respectively, in Table 2. In both cases, each cell only has corner nodes and $r_2 = 2.56$. The results in column 1 (with J_{new}) are essentially perfect and those in column 2 (with J_{old}) are very good.

In summary, the best strategy for h-refinement in the EFG appears to be to keep a relatively small number of nodes/cell, keep $\hat{d}/a$ fixed and increase the number of integration cells. Depending on the type of problem being solved, this can be done globally or in local regions in the domain B.

7
Discussion and conclusions

A simple and logical strategy is proposed in this paper for the purpose of alleviating problems encountered in imposing essential boundary conditions in the EFG method. This new strategy works very well in all the numerical examples presented here and is recommended for all future EFG applications.

A strategy for h-refinement of the EFG is proposed in which one increases the number of integration cells while keeping the ratio of (*# of nodes*)/(*# of cells*) small and roughly the same.

References

Belytschko, T.; Lu, Y. Y.; Gu, L. 1994 a: Element-free Galerkin methods. Int. J. Numer. Methods. Eng. 37: 229–256

Belytschko, T.; Gu, L.; Lu, Y. Y. 1994 b: Fracture and crack growth by element free Galerkin methods. Modeling Simul. Mater. Sci. Eng. 2: 519–534

Belytschko, T.; Lu, Y. Y.; Gu, L. 1995 a: Crack propagation by element-free Galerkin methods. Engng. Frac. Mech. 51: 295–315

Belytschko, T.; Lu, Y.Y.; Gu, L.; Tabbara, M. 1995 b: Element-free Galerkin methods for static and dynamic fracture. Int. J. Solids Structures. 32: 2547–2570

Belytschko, T.; Tabbara, M. 1996: Dynamic fracture using element-free Galerkin methods. Int. J. Numer. Methods. Eng. 39: 923–938

Belytschko, T.; Krongauz, Y.; Organ, D.; Fleming, M.; Krysl, P. 1996: Meshless methods: An overview and recent developments. Comput. Methods. Appl. Mech. Engng. (In press)

Krysl, P.; Belytschko, T.; 1995: Analysis of thin plates by the element-free Galerkin method. Comput. Mech. 17: 26–35

Lu, Y.Y.; Belytschko, T.; Gu, L. 1994: A new implementation of the element free Galerkin method. Comput. Methods Appl. Mech. Engng. 113: 397–414

Mukherjee, Y. X.; Mukherjee, S. 1997: The boundary node method for potential problems. Int. J. Numer. Methods. Eng. (In press)

Nayroles, B.; Touzot, G.; Villon, P. 1992: Generalizing the finite element method: Diffuse approximation and diffuse elements. Comput. Mech. 10: 307–318

Oden, J. T. 1995: Control of the computational process – modeling, accuracy, stability and efficiency. The third US National Congress on Computational Mechanics. Dallas, J.N. Reddy (ed.) Texas
A & M University, College Station Texas. pp. 2–3

Shephard, M. S.; Flaherty, J. E.; deCougny, H. L.; Ozturan, C.; Bottasso, C. L. 1995: Parallel automated adaptive finite element analysis. The third US National Congress on Computational Mechanics. Dallas, J.N. Reddy (ed.), Texas A & M University, College Station Texas. pp. 5–7

Zienkiewicz, O. C.; Taylor, R. L. 1994: The Finite Element Method, Vol. 1, 4th ed., McGraw Hill, London