AN APPROXIMATION OF THE ERROR IN ELEMENT FREE
GALERKIN METHOD

L. Gavete*, S. Falcón†, and J.C. Bellido††

* Escuela Técnica Superior de Ingenieros de Minas
  Universidad Politécnica
c/ Rios Rosas 21, 28003 Madrid, Spain
† Union Fenosa Distribución
c/ Capitán Haya 53, 28020, Madrid, Spain
e-mail: sfalcon@uef.es
†† CEU-Arquitectura
c/Camino de las Moreras s/n, 28040 Madrid, Spain
e-mail: jcbell@ceu.es

Key words: Meshless Methods, Moving Least Squares, Error Indicator, Error Estimator.

Abstract. The diffuse element method developed by Nayroles et al.1 is a new way for solving partial differential equations. In this method, only a mesh of nodes and a boundary description is needed to develop the Galerkin equations. The approximating functions are polynomials fitted to the nodal values of each local domain by a weighted least squares approximation. Belytschko et al.2,3 developed an alternative implementation using moving least squares approximation as were defined by Lancaster and Salkauskas4. They called their approach the Element Free Galerkin (EFG) method. In their work, Belytschko and his co-workers have introduced a background cell structure in order to carry out integration by numerical quadrature and Lagrange multipliers to enforce essential boundary conditions. Liu et al.5 has recently proposed a different kind of "griddles" multiple scale methods based on reproducing kernel and wavelet analysis (RPKM method).

Oñate et al.6 focused on the application to fluid flow problems with a standard point collocation technique. All these methods can be considered as Finite Point or Meshless Methods. In this paper we present a new approximation of the error for Element Free Galerkin (EFG) method, whose evaluation is computationally so simple that it can be readily implemented in existing EFG codes. The estimator allows the global energy norm error to be well approximated and also gives a good evaluation of local errors. It can thus be combined with a full adaptive process of refinement or, more simply, provide guidance for grid redesign which allows the user to obtain a desired accuracy.
1. INTRODUCTION

Several meshfree methods have been recently developed for the construction of Galerkin approximations to partial differential equations. The diffuse element method developed by Nayroles et al.\textsuperscript{1} is a new way for solving partial differential equations. In this method, only a mesh of nodes and a boundary description is needed to develop the Galerkin equations. The approximating functions are polynomials fitted to the nodal values of each local domain by a weighted least squares approximation.

Belytschko et al.\textsuperscript{2,3} developed an alternative implementation using moving least squares approximation as were defined by Lancaster and Salkauskas\textsuperscript{4}. They called their approach the Element Free Galerkin (EFG) method. In their work, Belytschko and his co-workers have introduced a background cell structure in order to carry out integration by numerical quadrature and Lagrange multipliers to enforce essential boundary conditions. Liu et al.\textsuperscript{5} has recently proposed a different kind of "gridless" multiple scale methods based on reproducing kernel and wavelet analysis (RPKM method).

Other path in the evolution of meshless methods has been the development of generalized finite difference (GFD) method, also called meshless finite difference method. One of the early contributors to the former was Perrone and Kao\textsuperscript{6}, but the more robust of these methods were developed by Liszka and Orkisz\textsuperscript{7,8}, using moving least squares(MLS) interpolation\textsuperscript{4}.

Oñate et al.\textsuperscript{9} focused on the application to fluid flow problems with a standard point collocation technique. All these methods can be considered as Finite Point or Meshless Methods.

On the one hand, Duarte and Oden\textsuperscript{10} and on the other, Babuska and Melenk\textsuperscript{11} have shown how meshless methods can be based on the partition of unity. In this line, the first authors have developed a new method that they denominate h-p clouds. Its main idea is in fact the construction of families of functions using the partition of unity, that is to say, they multiply a partition of unity by polynomials or another class of functions. The resulting functions conserve the good properties of the partition of unity and their linear combinations can represent polynomials of the degree that is required, being able to appreciate their interest therefore for the formation of p-adaptive hierarchical families.

A group of functions is called partition of unity if it has the following properties

\begin{align}
1) \quad \Phi^i_i &\in C^\infty \quad 1 \leq i \leq n \\
2) \quad \sum_{i=1}^{n} \Phi^i_i(x) &= 1 \quad \forall x \subset \Omega
\end{align}

being k the highest order of the polynomial, which is completely included in the basis.

There is not an only way of building the polynomial. They had to complete the condition of constituting a partition of unity, being able to base their election in if the problem to solve is or not linear, the complexity of the geometry of the domain, required regularity (C\textsuperscript{0}, C\textsuperscript{1}, ..., etc), etc.
The EFG method has been observed, using numerical studies, to be advantageous with respect to its rate of convergence and efficiency of modelling, when compared with finite element method (FEM). Other previously noted advantages include the elimination of problems associated with the meshing of complex domains, and high resolution of gradients with no post-processing of data. A drawback of the EFG methods lies in that they do not allow a direct imposition of essential boundary conditions.

The subject of error estimates for meshless methods and a consequent adaptive analysis, in which the approximation is successively refined to reach predetermined standards of accuracy, is central to the effective use of meshless codes for practical engineering analysis.

This paper begins with a brief summary of EFG method including a simple and effective strategy for the imposition of essential boundary conditions. The main contribution of the present paper is the proposal of a simple error indicator for EFG method. The error indicator is described in detail and some of the key characteristics are discussed. Finally, the performance of the error indicator is assessed by solving some test cases.

2. ELEMENT FREE GALERKIN/PENALISATION METHOD

In the Element Free Galerkin Method the idea is to replace the piecewise interpolation typical of the finite element method (FEM), for a local least squares fitting. The resulting function is more regular than the function of the FEM, since the discontinuous coefficients are replaced, by continuous functions of weight, that which gives a continuity $C^r$ ($r \geq 1$). It is possible to disconnect the number of nodes from the number of approximation parameters, because the least squares fitting replaces the standard FEM interpolation. The approximate function becomes smooth by using continuous weighting functions. To preserve the local character of the approximation is necessary to chose weighting functions that vanish at a certain distance from the point.

Around a point $\mathbf{x}$, the function $u^h(\mathbf{x})$ is locally approximated by:

$$u^h(\mathbf{x}) = \sum_{i=1}^{m} p_i(\mathbf{x}) a_i(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \mathbf{a}(\mathbf{x})$$  \hspace{1cm} (3)

where $m$ is the number of terms in the basis, the monomial $p_i(\mathbf{x})$ are basis functions, and $a_i(\mathbf{x})$ are their coefficients, which as indicated, are functions of the spatial co-ordinates $\mathbf{x}$.

The coefficients $a_i(\mathbf{x})$ are obtained by performing a weighted least square fit for the local approximation, which is obtained by minimising the difference between the local approximation and the function. This yields the quadratic form

$$J = \sum_{i=1}^{n} w(d_i)(\mathbf{p}^T(\mathbf{x}_i) \mathbf{a}(\mathbf{x}) - u^i)^2$$  \hspace{1cm} (4)

where $w(d_i) = w(\mathbf{x} - \mathbf{x}_i)$ is a weighting function with compact support.

Equation (4) can be rewritten in the form

$$J = (\mathbf{P} \mathbf{a} - \mathbf{u})^T \mathbf{W}(\mathbf{x}) (\mathbf{P} \mathbf{a} - \mathbf{u})$$  \hspace{1cm} (5)
where

\[ u^T = (u_1, u_2, \ldots, u_n) \]  

\[ P = \begin{bmatrix} \{P(x_1)\}^T \\ \vdots \\ \{P(x_n)\}^T \end{bmatrix} \]

\[ \{P(x_i)\}^T = \{P_1(x_i), \ldots, P_m(x_i)\} \]

\[ W = \text{diag}[w_1(x - x_1), \ldots, w_n(x - x_n)] \]

To find the coefficients \( a \), we obtain the extremum of \( J \) by

\[ \frac{\partial J}{\partial a} = A(x) a(x) - H(x) u = 0 \]

where

\[ A = P^T W(x) P \]

\[ H = P^T W(x) \]

and therefore

\[ a(x) = A^{-1}(x) H(x) u \]

The dependent variable \( u^h \) can then be expressed as

\[ u^h(x) = \sum_{j=1}^{n(x)} \Phi_j(x) u_j \]

where

\[ \Phi_i(x) = p^T(x) A^{-1}(x) H_i(x) \]

with \( H_i \) being the column \( i \) of \( H \).

The partial derivatives of the MLS shape functions are obtained as

\[ \Phi_{ij}(x) = p^T J A^{-1} H_j + p^T [A^{-1} H_{ij} - A_j A^{-1} H_j] \]

One of the biggest problems in the implementation of meshless methods resides in that the used approach is not an interpolation. MLS approximation, in general lack the delta function property of the usual FEM shape function, in that

\[ \Phi_i(x_j) = \delta_{ij} \]

where \( \Phi_i \) is the \( i \)th shape function evaluated at a nodal point \( x_j \) and \( \delta_{ij} \) is the Kronecker delta. This implies a difficulty when imposing the essential boundary conditions that it has led to the appearance of different solutions like they are, among other, Lagrange multipliers (Belytschko et al.) or modified
variational principles (Lu et al.).

According to Krongauz and Belytschko, the most satisfactory solution is the use of a joining with finite elements. Other important method to treat essential boundary conditions is given by Mukherjee and Mukherjee taking account that the method of moving least squares employed is an approximation instead of interpolation.

Another solution consists on to force that the weighting functions are singular in the boundary where the Dirichlet conditions are imposed (Duarte and Oden). Interpolating shape functions can be obtained using singular weights.

There are other techniques for meshless methods: in RPKM method, different procedures to generate admissible approximations for treatment of essential boundary conditions has been proposed by Günther and Liu and Gosz and Liu.

While these methods provide a means of overcoming the inherent difficulties of meshless methods, also have limitations and drawbacks. For example, Lagrange multipliers pose difficulties in that the resulting stiffness matrix is not longer positive definite or banded, and the size of the problem is increased. While modified variational principles enable the stiffness matrix to remain positive definite and banded, they are reported to be less accurate and are rather inconvenient. Coupling with finite elements wastes some of the advantages of meshless approximates and can result in discontinuities in the derivatives of the approximates.

In this paper we consider moving least squares method with appropriated weighting functions, and areas of influence for each one of them. Then, we obtain a local approximation that is close of interpolation. As we employ a local approximation, we need to satisfy the essential boundary conditions only approximately. A way to do it, is to use a constrained variational principle with a penalty function.

Consider the problem of making a functional \( \Pi \) stationary, subject to the unknown \( u \) obeying some set of additional relationships or constraints, which can be introduced at some points or over boundaries of the domain \( \Omega \).

For instance, if we require that \( u \) obey

\[
P(u) = 0 \quad \text{on} \quad \partial \Omega
\]

we would add to the original functional \( \Pi \) the term

\[
\alpha \int_{\partial \Omega} P^T(u)P(u)d(\partial \Omega)
\]

then we obtain

\[
\Theta = \Pi + \alpha \int_{\partial \Omega} P^T(u)P(u)d(\partial \Omega)
\]

in which \( \alpha \) is a penalty number and then require the stationarity of the functional \( \Theta \) will satisfy the constraints only approximately.

Alternatively, if the constraint \( P \) is applicable only at one or more points of the boundary, then the simple addition of \( P^T(u) P(u) \) at these points to the general functional \( \Pi \) will introduce a discrete
number of constraints.

In practical application with finite elements the method of penalty functions has proved quite effective. In EFG method to be able to employ this method with accuracy it is necessary that the approximation involved was very close to an interpolation. In order to do it is sufficient to consider areas of influence for each weighting function very small overlapped but with sufficient number of nodes for the involved approximation, (see Gavete et al.).

Furthermore taking account MLS we shall calculate as best values for the function and the gradients, these obtained by the approximations

\[ u_h(x) = \sum_{i=1}^{n(x)} \Phi_i(x) u_1 \]

\[ \frac{\partial u_h}{\partial x}(x) = \sum_{i=1}^{n(x)} \frac{\partial \Phi_i}{\partial x}(x) u_1 \]

and similarly for the other gradient.

The values of \( u_h, \frac{\partial u_h}{\partial x}, \frac{\partial u_h}{\partial y} \) at all the nodes are accepted as the numerical solution of the problem.

3. ERROR ESTIMATION IN MESHLESS METHODS

The error estimation should be a main tool in every adaptivity process. This is the reason of the actual great importance of the estimation. It allows us to know the quality of the solution, and hence if it is or not acceptable. Moreover, it provides some information about the changes that are necessary to make in the used mathematical model to reach, in an economic way, the desired solution.

In meshless methods the “a posteriori” error has been calculated in order to redistribute the nodes. In this paper we would try to make an overview of some strategies used to achieve it. Some error indicators and error estimators have been developed for meshless methods using different techniques. In the last years some research has been devoted to developing reliable error estimation procedures in meshless methods: Orkisz, Duarte and Oden, Gavete et al., Laouar and Villon.

Orkisz presented an adaptive multigrid meshless finite difference method. A solution convergence rate

\[ \beta_i^K = \frac{\|u_i^K - u_i^{K-1}\|}{\|u_i^K\|} \leq c_3 \quad \forall \ i=1, ..., n^o \text{ common nodes} \]  

is examined at those nodes \( P_i, i=1,2,... \) which preserve the same location in the subsequent meshes \( ...,k-2,k-1,k,... \); \( c_3 \) is an imposed threshold value resulting from the required solution precision. Using as error indicators the residuals of the solution and \( \beta_i^K \), a series of more and more dense meshes can be generated, to establish an adaptive solution process.

Laouar y Villon, presented a technique of resolution using the diffuse element method with adaptive set of nodes. Nodes are generated by a quadtree type decomposition of the area and the
adjustment is made with the help of a posteriori knowledge of error estimate. They estimate the error by

\[ \|\varepsilon\|_h = \left( \int_{\Omega} \left( (\{\hat{\sigma}\} - \{\bar{\sigma}\})^r (\{\hat{\sigma}\} - \{\bar{\sigma}\}) \right) \, d\Omega \right)^{1/2} \]  

(24)

where, \( \{\hat{\sigma}\} \) is a diffuse element continuous stress field obtained by auto-equilibrium, \( \{\bar{\sigma}\} \) is a continuous stress field obtained by auto-equilibrium.

Duarte y Oden\textsuperscript{10,14}, use the partition of unity concept in a very general manner by constructing it from an moving least square shape function. The major advantage of the Duarte-Oden formulation is that it enables the extrinsic basis to vary from node to node, thus facilitating hp-adativity. Also Duarte and Oden have shown that the rate of convergence for the hp-clouds approximations is given by

\[ |u(x) - u_h(x)|_{m,\Omega} \leq c \, h^{k+1-m} |u|_{k+1,\Omega} \]  

(25)

where h is the window size, k is the polynomial degree of the basis, m is the order of the governing equation. Liu, Li and Belytschko\textsuperscript{21} have obtained similar results for the more restrictive case of a kernel (or MLS) approximation. In all cases the results have been restricted to convex domains with Lipschitz continuous boundaries.

4. A SIMPLE ERROR INDICATOR

In this paper we are going to use an easy postprocess error indicator. At the beginning, we use the solution provided by the EFG/penalisation method, that is the function value and its gradients in every node of the domain.

Moreover, we could calculate the gradient values in the Gauss integration point. For example in 1-D and with one Gauss point in every integration domain, we would obtain results as those shown in figure 1. This figure shows a continuous approximation curve of the gradient using MLS (it has been supposed an interpolation) and the bars chart represents the gradients in the integration points.

From this data, we could obtain an indication of the “a posteriori” error. Let us suppose that for each node in the domain we make a tessellation of the gradients, and we get two different values in every integration point. One of them is the calculated in the nearest node to each one of the integration points, and the other is calculated using MLS in the integration points. Using the integral norms, this procedure is an error indicator that provides a very valid and useful quality information. We use as better value the solution, the one coming from the integration points (see fig. 1).

The local error measurement, defined as the difference between the approximate solution to the exact one, is often very difficult to calculate and could provide a confused information. This is the situation in areas with punctual loads. If we want to be able to make global and local approximations, we should use integral norms. Furthermore, some norms have physical meanings that make understand better the problem we are studying.
We will use the classic energy norm and in each integration domain we will evaluate it in the same way that it is done in FEM with the approximation error. In both cases we will use as “a posteriori” error indicator the difference between the two approaches previously defined. One is calculated through EFG-penalisation in the integration points and the other by tessellation of the gradients.

\[
\|\varepsilon\|_{\Omega_i}^2 = \left[ \int_{\Omega_i} \left( \{\varepsilon\} - \{\varepsilon^e\} \right)^T \left( \{\sigma\} - \{\sigma^e\} \right) d\Omega \right] = \\
\left[ \int_{\Omega_i} \left( \{\sigma\} - \{\sigma^e\} \right)^T [D]^{-1} \left( \{\sigma\} - \{\sigma^e\} \right) d\Omega \right] = \\
\left[ \int_{\Omega_i} \{\varepsilon\}^T [D] \{\varepsilon\} d\Omega \right] \\
\|w\|_{\Omega_i}^2 = \left[ \int_{\Omega_i} \{\sigma^e\}^T [D]^{-1} \{\sigma^e\} d\Omega \right]^2
\]

Although this norm is defined on the whole domain, we note that the square of each can be obtained by summing contributions of each one of the integration cells Ω. Thus,

\[
\|\varepsilon\|_{\Omega}^2 = \sum_i \|\varepsilon\|_{\Omega_i}^2
\]
\[
\|w\|_{\Omega}^2 = \sum_i \|w\|_{\Omega_i}^2
\]
The relative error, \( \eta \), and the effectivity index, \( \Theta \), are defined by:

\[
\eta = \frac{\|e\|}{\| \text{Energy} \|}, \tag{30}
\]

\[
\Theta = \frac{\|e\|}{\|E\|}, \tag{31}
\]

where \( |E| \) is the exact error. As the exact energy norm is not known in advance, is approximated by \( | \text{Energy} |^2 = \|w\|^2 + \| e \|^2 \).

The relative error and the effectivity index can be determined for the whole domain or for each one of the integration cells.

5. NUMERICAL RESULTS

To demonstrate the performance of the error indicator, we study two different cases with known analytical solutions. To verify the validity of the described error indicator we present two numerical examples: Laplace Equation and Elasticity Equation.

The following two weight functions were tested:

a) Polynomial weight function (Quartic Spline):

\[
w(d) = 1 - 6 \left( \frac{d}{dm} \right)^2 + 8 \left( \frac{d}{dm} \right)^3 - 3 \left( \frac{d}{dm} \right)^4 \tag{32}
\]

when \( d \leq dm \), and \( w_i = 0 \) when \( d > dm \); and where \( d = \sqrt{(x - x_i)^2 + (y - y_i)^2} \) and \( dm = \text{rinf} \)

b) Polynomial weight function (Cubic Spline):

\[
w(d) = \begin{cases} 
\frac{2}{3} - 4 \left( \frac{d}{dm} \right)^2 - 4 \left( \frac{d}{dm} \right)^3 & \text{for } d \leq \frac{1}{2} dm \\
\frac{4}{3} - 4 \left( \frac{d}{dm} \right)^2 + 4 \left( \frac{d}{dm} \right)^3 - \frac{4}{3} \left( \frac{d}{dm} \right)^3 & \text{for } \frac{1}{2} dm < d \leq dm \\
0 & \text{for } d > dm
\end{cases} \tag{33}
\]

and \( dm = \text{rinf} \)

5.1 Laplace equation

Firstly, the Dirichlet problem for Laplace equation is chosen as model problem. Let us consider the Laplace equation in 2-D

\[
\nabla^2 u = 0 \quad \text{in } \Omega \tag{34}
\]
with boundary conditions

$$u = u_d \quad \text{on } \partial \Omega$$

(35)

where $\partial \Omega$ is the boundary of $\Omega$, being $\Omega = \{0.01, 1.01\} \times \{0.01, 1.01\}$.

We shall consider as solution of Laplace equation:

$$u = \log(x^2 + y^2)$$

(36)

The EFG method with linear shape functions and Penalty functions to enforce essential boundary conditions was used. Different clouds of nodes were considered as it is shown in figure 2. The integration space was defined according with each cloud of nodes.

Figure 2: Clouds of points and integration cells.
Next, some results have been obtained for Laplace equation with the case defined above and using the Quartic Spline weighting function. The results are shown in table 1, for different Gauss quadratures using fixed radius of influence, rinf.

<table>
<thead>
<tr>
<th>NODIS</th>
<th>NODES</th>
<th>CELLS</th>
<th>rinf=0.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODES</td>
<td>9X9</td>
<td>CELLS</td>
<td>4X4</td>
</tr>
<tr>
<td>ORDER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INTEGRATION ORDER</td>
<td>% Error in energy</td>
<td>Efectiv. Index</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Exact</td>
<td>Aproxim.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>25.35</td>
<td>30.60</td>
<td>1.41</td>
</tr>
<tr>
<td>5</td>
<td>26.87</td>
<td>36.34</td>
<td>1.61</td>
</tr>
<tr>
<td>6</td>
<td>27.81</td>
<td>29.39</td>
<td>1.23</td>
</tr>
<tr>
<td>7</td>
<td>28.12</td>
<td>30.90</td>
<td>1.28</td>
</tr>
<tr>
<td>8</td>
<td>28.16</td>
<td>33.92</td>
<td>1.42</td>
</tr>
<tr>
<td>9</td>
<td>28.16</td>
<td>30.69</td>
<td>1.27</td>
</tr>
<tr>
<td>10</td>
<td>28.18</td>
<td>31.04</td>
<td>1.29</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NODIS</th>
<th>NODES</th>
<th>CELLS</th>
<th>rinf=0.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODES</td>
<td>9X9</td>
<td>CELLS</td>
<td>8X8</td>
</tr>
<tr>
<td>ORDER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INTEGRATION ORDER</td>
<td>% Error in energy</td>
<td>Efectiv. Index</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Exact</td>
<td>Aproxim.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>27.70</td>
<td>29.89</td>
<td>1.26</td>
</tr>
<tr>
<td>5</td>
<td>28.13</td>
<td>29.35</td>
<td>1.21</td>
</tr>
<tr>
<td>6</td>
<td>28.16</td>
<td>30.80</td>
<td>1.28</td>
</tr>
<tr>
<td>7</td>
<td>28.18</td>
<td>29.34</td>
<td>1.21</td>
</tr>
<tr>
<td>8</td>
<td>28.16</td>
<td>31.11</td>
<td>1.29</td>
</tr>
<tr>
<td>9</td>
<td>28.17</td>
<td>29.54</td>
<td>1.22</td>
</tr>
<tr>
<td>10</td>
<td>28.16</td>
<td>31.26</td>
<td>1.30</td>
</tr>
</tbody>
</table>

Table 1: Quartic Spline, Fixed radius

The global effectivity index represents the performance of the global error indicator. It is, however, desirable that the error indicator should provide not only reliable error approximation but also reliable cell error approximation, which is particularly important for point refinement in the adaptive analysis.

Next, a group of studies has been carried out with models consecutively more refined. We have try to check the performance of the indicator. Figure 2 shows every different used meshes. In all the cases quartic spline weighting functions and variable radius of influence have been used. The radius of domain of influence, rinf, was computed by rinf = α dI, with dI chosen to be the distance to the third closest node from node I; α was chosen to be 2. Each model has been designated with a code pointing the degree of refinement (see fig. 2). Results are given in Table 2.
<table>
<thead>
<tr>
<th>Node Configuration</th>
<th>% Error in energy</th>
<th>Effectiv. Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>81 nodes (9x9); 4x4 cell.</td>
<td>26.49  27.95</td>
<td>1.22</td>
</tr>
<tr>
<td>81 nodes (9x9); 8x8 cell.</td>
<td>29.07  27.75</td>
<td>1.10</td>
</tr>
<tr>
<td>97 nodes (9x9+16); 8x8 cell.</td>
<td>17.08  22.95</td>
<td>1.45</td>
</tr>
<tr>
<td>109 nodes (9x9+28); 8x8 cell.</td>
<td>16.35  22.13</td>
<td>1.45</td>
</tr>
<tr>
<td>118 nodes (9x9+37); 8x8 cell.</td>
<td>16.30  22.05</td>
<td>1.45</td>
</tr>
<tr>
<td>289 nodes (17x17); 8x8 cell.</td>
<td>15.71  21.49</td>
<td>1.47</td>
</tr>
</tbody>
</table>

Table 2: Quartic Spline, Variable radius

Figure 3: 81 nodes (9x9); 4x4 cells. Errors in every cell.

Figure 4: 97 nodes (9x9+16); 4x4 cells. Errors in every cell.
Figure 5: 81 nodes (9x9); 8x8 cells. Errors in every cell.

Figure 6: 97 nodes (9x9+16); 8x8 cells. Errors in every cell.

Figure 7: 109 nodes (9x9+28); 8x8 cells. Errors in every cell.
Model T30904 has provided a very good error in the function, not in the gradients. As it is shown, the energy error distribution is very irregularly scattered in the domain. Especially, see in fig 3, the first cell (in the top of figure) has higher error. This is a logical result since there is a singularity in the co-ordinates origin. However, what we are trying to do is to spread the error uniformly over the domain. That will provide better gradient results.

The previous conclusion leads to the next model, T30904r1. This model does not provide the desired results. The errors are very high and the error propagates to the nearest cells, without any error decreasing in the first cell, see fig.4. The reason is that the model obtained has a very irregularly node density over the domain. It has a sharp transition from one cell to another. It will be necessary to add nodes in the surroundings of cell n° 1, but because the model is too coarse it has been considered as a better solution to refine uniformly the cells firstly, than the node refinement.

Then models T30908 and T90908r1 come up. They are very similar to the previous ones but using 64 integration cells. The results obtained, although better than with model T30904 and T30904r1, lead to the similar conclusions. As we desire, the global error has decreased, especially in cell n°1, although an increase has been produced in the nearest cells, see figures 5 and 6.
Nevertheless, the most important fact is that the cell, with error has risen the most, are what we can call the second ring of cells. These are the cells nearest to the last refined ones. It is necessary to create smooth transition of nodes.

In T30908r2 model, (see figures 2 and 7), a node has been added in the centre of the cells belonging to the third and fourth rings. The results strongly improve. The error in the function is much better than in the former model and in the gradient decrease smoothly. The error indicator decrease in the domain and it is homogenised. As in the other cases it increase a little in the ring when the node refinement changes. Then, a new row has to be refined. With model T30908r3 error drops a little although the results are very similar, see fig 8.

Obviously a uniform refinement as in the model T31708 leads to a better results, especially in gradients, but the error distribution is very irregular, see figure 9. However the computational requirements are higher (289 nodes versus 118) and the results are quite similar to the T30908r3 model.

Although, all the results have been shown consecutively, the successively refined models has been chosen by making use of the new error indicator with the results obtained in previous models.

In the following table 3, we can see the results obtained using at same time variable radius of influence and variable size of integration cells, depending of the local nodal density, with 4x4 integration order.

<table>
<thead>
<tr>
<th>% Error in the Energy</th>
<th>Efectivit. Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>97 nodes(9x9+16);37 cells</td>
<td>17,14 22,96 1,44</td>
</tr>
<tr>
<td>109 nodes(9x9+28);40 cells</td>
<td>17,22 21,87 1,36</td>
</tr>
<tr>
<td>118 nodes(9x9+37);52 cells</td>
<td>17,11 21,65 1,36</td>
</tr>
</tbody>
</table>

Table 3 : Quartic Spline. Variable number of integration cells depending of nodal density

5.2 Elasticity equations

Next, the problem of a plate with a hole is examined. The solution for an infinite plate with a hole subject to a far-field unit traction in the x direction is given in Timoshenko and Goodier\textsuperscript{22} as

\[
\sigma_{xx} = 1 - \frac{a^2}{r^2} \left( \frac{3}{2} \cos(2\theta) + \cos(4\theta) \right) + \frac{3a^4}{2r^4} \cos(4\theta) \]

\[
\sigma_{yy} = -\frac{a^2}{r^2} \left( \frac{1}{2} \cos(2\theta) - \cos(4\theta) \right) - \frac{3a^4}{2r^4} \cos(4\theta) \]

\[
\sigma_{xy} = -\frac{a^2}{r^2} \left( \frac{1}{2} \sin(2\theta) + \sin(4\theta) \right) + \frac{3a^4}{2r^4} \sin(4\theta) \]

(37)
Only the upper quadrant of the problem is modelled as it is shown in Figure 10 with dimensions a=1 and b=5. The following parameters were used E=1000; \( \nu = 0.3 \). The tractions from the exact solution are applied to the outer boundaries of the model, and zero normal displacements are prescribed on the symmetry boundaries. Three discretizations of 30, 72 and 255 nodes are used. The three discretizations are shown in Figure 11: the background meshes are such that the nodes are located at the vertices of the integration elements. In this case triangular integration cells have been used. The radius of domain of influence, r_{inf}, was computed by r_{inf} = \alpha \ d_{t}, with d_{t} chosen to be the distance to the fourth closest node from node I; \( \alpha \) was chosen to be 1.4. Weighting functions (32) and (33) were tested. Results obtained are given in Tables 4 and 5.

<table>
<thead>
<tr>
<th>QUARTIC SPLINE</th>
<th>INTEGRATION ORDER</th>
<th>% Error in energy</th>
<th>Efectiv. Index</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Exact</td>
<td>Approx.</td>
</tr>
<tr>
<td>30 nodes</td>
<td>7</td>
<td>17.96</td>
<td>18.27</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>16.67</td>
<td>17.25</td>
</tr>
<tr>
<td>72 nodes</td>
<td>7</td>
<td>9.04</td>
<td>11.10</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>7.75</td>
<td>9.57</td>
</tr>
<tr>
<td>255 nodes</td>
<td>7</td>
<td>5.00</td>
<td>6.19</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>5.18</td>
<td>5.89</td>
</tr>
</tbody>
</table>

Table 4: Plate with a hole. Quartic Spline.
In both tables 4 and 5 results obtained are very similar. A good convergence is obtained for the two different weighting functions considered. The three models given in figure 11 have a smooth transition of nodes.

6. CONCLUSIONS

In EFG method, it is obviously important that the error of approximation should be assessed and
an indication given to the user how this error can be reduced to acceptable values.

An easy method to estimate the error has been proposed. It can not be considered as a real estimator, as those used in FEM, but it is a good approach, and very easy to implement and use. The procedure is inexpensive and it can be applied to many different partial differential equations. The performance of the error indicator has been demonstrated on two different problems with known analytical solution. The calculation of the error indicator is performed in a cell by cell basis, and is found to be accurate and efficient.

The method can be used in complex domains, removing problems with the grid. Changes in the model has to be done simply adding nodes where the indicator shows but taking into account that better results are obtained with uniform densities of nodes. It is recommended to use smooth transition of nodes, creating areas of decreasing nodal densities.

REFERENCES