Computer Assisted Mechanics and Engineering Sciences, 5: 479–501, 1998.

Consistency approach and diffuse derivation in element free methods based on moving least squares approximation

Piotr Breitkopf
Codiciel, UPS 856 CNRS, Mont-Sain-Aignan, France
e-mail: Piotr.Breitkopf@codiciel.fr

Gilbert Touzot
INSA de Rouen, Rouen, France,
e-mail: Gilbert.Touzot@insa-rouen.fr,

Pierre Villon
Université de Technologie de Compiègne, Laboratoire LG2MS Compiègne, France,
e-mail: Pierre.Villon@utc.fr.

(Received February 11, 1998)

The paper concerns shape functions formulations in the scope of the recent methods generalizing finite elements and whose common feature is the absence of a mesh. These methods may also be interpreted as a generalization of the finite difference approach for irregular grids. The shape functions obtained by the Moving Least Squares an by the GFDM (Generalized Finite Difference Method) approach exhibit a number of interesting properties, the most interesting being a local character of the approximation, high degree of continuity and the satisfaction of consistencu contraints neccesary for exact reproduction of polynomials. In the present work we formulate the shape functions directly as solutions of the minimization of a weighted quadratic form subjected to the consistency contraints explicity introduced by Lagrange multipliers. This approach gives similar results as the standard moving least squares algorithm applied to the Taylor series expansion where the consistency is automatically satisfied but is more general in the sense, that an explicit specification of wished properties permits an introduction of additional arbitrary conraints other than consistency. It also leads to faster and more robust algorithms by avoiding matrix inversion. On the other hand, the consistency based formulations naturally lead to diffuse (or incomplete) derivatives of the shape functions. They are obtained at a significantly lower cost than full derivatives and their convergence to extact derivatives is demonstrated.

1. Introduction

In the recent years, interest is groving for a class of methods challenging the finite element method in the field of computational mechanics. The common feature of these new methods is the absence of explicit finite element mesh, the continuous media being discretized in terms of nodes rather than in terms of nodes and elements. In these methods, the interactions between nodes are governed by vicinity criteria, rather than by finite element connectivities. This common starting point led to the development of a number of methods: Diffuse Element Method [12], elemenet Free Galerkin Method [1], Finite Points Method [13], h-p Clouds Method [4, 5], Boundary Nodes Method [11]. Bel 96] provides an extensive panorama of the field. On the other hand, the work [8] gave rise to an extensive research on Finite Differences on Irregular Grids [6, 9, 10, 14, 15].

Besides the absence of explicit finite element mesh, the second feature common to the majority of these methods is the approximation technique. The actual work is centered on the class of methods based on different variants of the Moving Least Squares, subsequently developed in [6, 9, 10] and whose mathematical foundations are given in [1]. In the present paper, we shall focus on some

improvements of the approximation method, independently of the particular technique used for building and solving the equations. After a brief section dedicated to the Moving Least Squares, we use the centered version of the approximation in oredr to obtain all the shape functions and their diffuse derivatives in one operation. Next, we present an alternative technique of constructing the shape functions directly based on consistency conditions introduced by Lagrange multipliers. It naturally leads to the diffuse derivatives of the shape functions. The obtained linear system exhibits a particular structure which may be solved by a multi-step algorithm, avoiding expicit matrix inversion. Finally, we illustrate the proposed approach by several examples, comparing in particular diffuse and full derivatives.

2. Moving least squares approximation

We begin by introducing the Moving Least Squares (MLS) method following the approach [7] which may be intrpreted [12] as a generalization of the finite elements. An alternative method [9] based on a local Taylor expansion presents numerous advantages and could also be used. We are looking for a local approximation of a function u_{ex} at a point x, based on the nodal values u_i of u_{ex} at a limited number of points close to x. The unknown function u_{ex} is approximated in the vicinity of x by:

$$u_{\text{ex}}(x) \approx u_{\text{app}}(x) = \mathbf{p}^T(x)\mathbf{a}(x). \tag{1}$$

The basis functions of the approximation may be polynomial':

$$\mathbf{p}^T(x) = [1x \dots x^n].$$

The coefficients a_i of the approximation are related to the nodal values u_i by minimizing a norm of the weighted difference between the estimated values at nodes and the nodal values u_i.

$$J_x(\mathbf{a}) = \frac{1}{2}\sum_i w_i(x_i, x)(\mathbf{p}^T(x_i)\mathbf{a} - u_i)^2. \tag{2}$$

The contribution of each nodal value to the approximation is influenced by a weighting function $w(x_i, x)$ such that $w(x_i, .) > 0$ inside the domain of influence of the node i et $w(x_i, .) = 0$ othervise, providing a local character to the approximation.

Generally MLS does not interpolate data, so the relation

$$u_{\text{app}}(x_i) = u_{\text{ex}}(x_i) \tag{3}$$

is not satisfied unless the weighting function is infinite at the node

$$x \to x_i \;\Rightarrow\; w(x_i, x) \to \infty.$$

In this case the influence of other nodes vanishes, the approximation becomes interpolating and Eq. (3) holds. In the forthcoming paper [3] we present also a method of obtaining non-singular interpolating weght functions.

The derivatives of $\tilde{u}(x)$ may be approximated in two ways, either by derivating voth $\mathbf{p}$ and $\mathbf{a}$ (full derivative):

$$\frac{du}{dx} = \frac{d\mathbf{p}^T}{dx}(x)\mathbf{a}(x) + \mathbf{p}^T(x)\frac{d\mathbf{a}}{dx} \tag{4}$$

or by considering $\mathbf{a}$ as a constant like in finite elemenet approximation, leading to the "diffuse derivative" denoted by $\delta u/\delta x$

$$\frac{\delta u_{\text{app}}}{\delta x}(x) = \frac{d\mathbf{p}^T}{dx}(x)\mathbf{a}(x). \tag{5}$$

The former approach is used in the Element Free Galerkin method [1] and the latter one is analogous to the derivatives obtained by the GFDM method [6, 9, 10] where second order diffuse derivatives are employed. The first order diffuse derivative was reintroduced by [12] along with the Diffuse Element method. Both derivatives converge to exact derivatives when the discretization size tends to zero. The two derivatives are equivalent in several cases:

- the evaluation point x is located at a node and the interpolating condition (3) is verified,
- the weights $w(x_i, x)$ are constant over a vicinity of x: in this case the coefficients $\mathbf{a}$ are constant, the term $\dfrac{d\mathbf{a}}{dx}$ vanishes and $\dfrac{\delta u_{\text{app}}}{\delta x}(x) \equiv \dfrac{du_{\text{app}}}{dx}(x)$, which is in particular the case in the finite element method [12],
- u may be expressed as a linear combination of the basis functions p_i: the coefficients $\mathbf{a}$ are constant in this case too; for an arbitrary function this is verified at the limit, when the discretization size tends to zero and the function converges to the first terms of its Taylor expansion

In this paper, we discuss different aspects of MLS approximation, by considering convergence, consistency and computational effectiveness.

2.1. Centered moving least squares

In this section we will show that the diffuse derivatives (5) correspond to an approximation of a Taylor series expansion [9]. Let us consider the function $u_{\text{ex}}(.)$ to be approximated, and suppose it is $k+1$ times continuously derivable. The Taylor expansion of $u_{\text{ex}}(x)$ in the vicinity of point x gives

$$u(x_i) = \sum_{l=0,\ldots,k} \frac{(x_i - x)^l}{l!} \frac{d^l}{dx^l} u_{\text{ex}}(x) + \ldots .$$

Let us introduce a centered polynomial basis $\mathbf{q}$

$$\mathbf{q}^T(x_i - x) = \left[1 \;\; (x_i - x) \;\; \ldots \;\; \frac{(x_i - x)^k}{k!}\right].$$

For $k = 2$, the new basis $\mathbf{q}$ is related to the basis $\mathbf{p}$ by the following relationship

$$\mathbf{Q}\mathbf{p}(x_i) = \mathbf{q}(x_i - x), \qquad \mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 \\ -x & 1 & 0 \\ \frac{1}{2}x^2 & -x & \frac{1}{2} \end{bmatrix}$$

and inversely

$$\mathbf{p}(x_i) = \mathbf{Q}^{-1}\mathbf{q}(x_i - x), \qquad \mathbf{Q}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ x & 1 & 0 \\ x^2 & 2x & 2 \end{bmatrix}.$$

The matrix $\mathbf{Q}$ is nonsingular as it corresponds to a basis change in a polynomial vector space. The nodal approximation (1) becomes

$$u_{\text{app}}(x) = \mathbf{q}^T(x_i - x)\mathbf{Q}^T(x)\mathbf{a}(x) = \mathbf{q}^T(x_i - x)\boldsymbol{\alpha}(x), \tag{6}$$

where

$$\boldsymbol{\alpha}(x) = \mathbf{Q}^{-T}(x)\mathbf{a}(x). \tag{7}$$

Inserting (6) into (2) gives the modified criterion which depends now on the vector $\boldsymbol{\alpha}$

$$\mathrm{J}_x(\boldsymbol{\alpha}) = \frac{1}{2}\sum_i w_i(x_i, x)(\mathbf{q}^T(x_i - x)\boldsymbol{\alpha} - u_i)^2. \tag{8}$$

The minimization of (8) yields

$$\boldsymbol{\alpha}(x) = \mathbf{A}(x)^{-1}\mathbf{B}(x)\mathbf{u}, \tag{9}$$

where

$$\mathbf{A}(x) = \sum_i w_i \mathbf{q}(x_i - x)\mathbf{q}^T(x_i - x),$$

$$\mathbf{B}(x) = [\ldots w_i\mathbf{q}(x_i - x)\ldots]. \tag{10}$$

The centered approach presents better conditioning properties then using the global coordinates. The conditioning may be further improved by introduction of adimensional coordinates ξ_i

$$\begin{pmatrix} 1 \\ \xi_i \\ \vdots \\ \xi_i^n \end{pmatrix} = \begin{bmatrix} 1 & & & 0 \\ & \frac{1}{r} & & \\ & & \ddots & \\ 0 & & & \frac{1}{r^n} \end{bmatrix} \begin{pmatrix} 1 \\ x_i - x \\ \vdots \\ (x_i - x)^n \end{pmatrix}, \quad \text{so } \mathbf{D}(r)\mathbf{p}(x_i - x) = \mathbf{p}(\xi_i)$$

with $r = r(x_i, x)$ chosen in the way that $0 \leq \xi \leq 1$. The cost function (8) becomes

$$\mathrm{J}_x(\beta) = \frac{1}{2}\sum_i w_i(x_i, x)(\mathbf{q}^T(\xi)\beta - u_i)^2$$

and the coefficients α are given by

$$\alpha = \mathbf{D}(r)\beta.$$

Rewritting (7) one demonstates that the coefficients $\boldsymbol{\alpha}$ are the successive diffuse derivatives of the approximation

$$\alpha_0 = a_0 + a_1 x + a_2 x^2 = \mathbf{pa} = u_{\mathrm{app}}(x),$$

$$\alpha_1 = a_1 + 2a_2 x = \frac{d\mathbf{p}}{dx}\mathbf{a} = \frac{\delta u_{\mathrm{app}}}{\delta x}(x),$$

$$\alpha_2 = 2a_2 = \frac{d^2\mathbf{p}}{dx^2}\mathbf{a} = \frac{\delta^2 u_{\mathrm{app}}}{\delta x^2}(x).$$

The convergence of $\boldsymbol{\alpha}$ to the vector

$$\left[u_{\mathrm{ex}}(x), \frac{du_{\mathrm{ex}}}{dx}(x), \ldots, \frac{d^k u_{\mathrm{ex}}}{dx^k}(x)\right]$$

has been demonstrated by [19] and recently in the scope of MLS by [18] (see Appendix 1):

$$\left|\alpha_l(x) - \frac{d^l u_{\text{ex}}}{dx^l}\right| < \frac{r(x)^{k-l+1}}{(l+1)!}\left\|\frac{d^{k+1}u_{\text{ex}}}{dx^{k+1}}\right\| K(x)$$

where k is the degree of the polynomial basis, r is the radius of the domain of fitting, the term $\left\|\frac{d^{k+1}u_{\text{ex}}}{dx^{k+1}}\right\|$ depends on the regularity of the function $u_{\text{ex}}(x)$ and the term $K(x)$ is related to the local topological quality of the pattern. With a fixed pattern of points and decreasing r, the order of convergence of the approximation of the k-th derivative is $0(k-l)$.

The term $K(x)$ expresses the conditioning of the system $\mathbf{A}(x)^{-1}\mathbf{B}(x)$ and depends on the local nodal pattern taken into account in the approximation at point x. The following figures illustrate different cases:

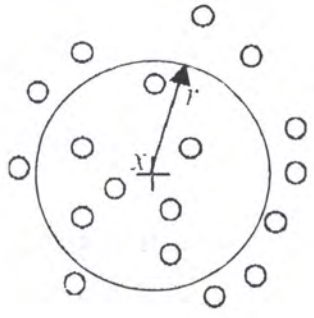


Fig. 1. Well conditioned pattern with the polynomial basis $p =< 1, x, y >$

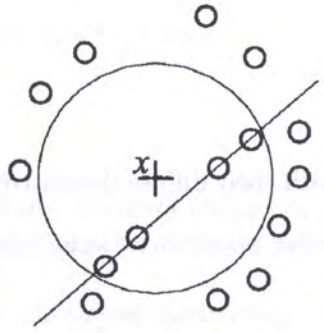


Fig. 2. Pathological pattern with $p =< 1, x, y >$

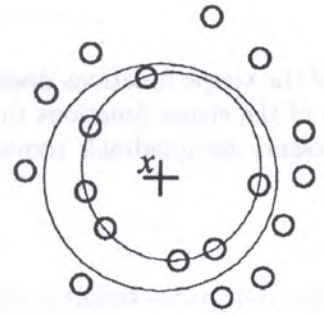


Fig. 3. Pathological pattern with $p =< 1, x, y, x2, y2, xy >$

In the above examples, we chose $w(x_i, x) = w_{\text{ref}}\left(\frac{\|x_i - x\|}{r}\right)$ where w_{ref} is a bell shaped window function, $\|.\|$ is the Euclidian norm L2 and $r(x)$ is the distance from x to the $k+2$ th neighbor node of x. Figure 1 presents a well conditioned case with a linear basis. Figures 2 and 3 show cases where the matrix $\mathbf{A}$ becomes singular, respectively for colinear points with a linear basis and for cocircular points with a quadratic basis. These pathological situations are limit cases in which the approximation may not be performed. More frequently, the patterns would be close to these singular ones and will produce ill-conditioned matrices, spoiling convergence. One has to note, that these results do not depend on the choice of the weight function in MLS so they are similar to those obtained by [16] for a finite difference scheme on irregular mesh.

3. Shape functions

Relation (9) may be written

$$\begin{Bmatrix} u_{\text{app}}(x) \\ \frac{\delta u_{\text{app}}}{\delta x}(x) \\ \vdots \\ \frac{\delta u^k_{\text{app}}}{\delta x^k}(x) \end{Bmatrix} = \mathbf{A}^{-1}\mathbf{B}\{u_i\} \tag{11}$$

or, with the usual finite element notation

$$u_{\text{app}}(\mathbf{x}) = \sum_i N_i(\mathbf{x}) u_i, \tag{12}$$

$$\frac{\delta u^l_{\text{app}}}{\delta x^l}(\mathbf{x}) = \sum_i \frac{\delta^l N(\mathbf{x})}{\delta x^l} u_i.$$

All the shape functions N_i, along with their diffuse derivatives are expressed in a compact form as

$$\begin{bmatrix} \mathbf{N}^T \\ \frac{\delta \mathbf{N}^T}{\delta x} \\ \frac{\delta \mathbf{N}^T}{\delta y} \\ \vdots \end{bmatrix} = \mathbf{A}^{-1}\mathbf{B}. \tag{13}$$

Computing the diffuse derivatives of the shape functions does not present any computational overhead compared with the evaluation of the shape functions themselves.

The cancellation conditions, necessary for quadratic convergence of $u_{\text{app}}(\mathbf{x}) = \sum_i N_i(\mathbf{x}) u_i$ are

$$\sum_i N_i = 1,$$

$$\sum_i N_i(x_i - x) = 0. \tag{14}$$

The cubic convergence of u_{app} requires also

$$\sum_i N_i(x_i - x)^2 = 0.$$

These consistency contraints may be presented compactly

$$\mathbf{PN} = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} = \mathbf{e}^1 \tag{15}$$

with

$$\mathbf{P} = \begin{bmatrix} 1 & \dots & 1 \\ x_1 - x & \dots & x_n - x \\ \frac{1}{2}(x_1 - x)^2 & \dots & \frac{1}{2}(x_n - x)^2 \end{bmatrix}.$$

Linear convergence of $\dfrac{du_{\text{app}}}{dx}(\mathbf{x}) = \sum_i N_i'(\mathbf{x})u_i$ requires

$$\begin{aligned} &\sum_i N_i' = 0, \\ &\sum_i N_i'(x_i - x) = 1 \end{aligned} \tag{16}$$

and for quadratic convergence we have to satisfy also

$$\sum_i N_i'(x_i - x)^2 = 0.$$

These properties, well known in finite elements under the name of consistency conditions, are necessary for the convergence of a variational formulation based on first derivatives, such as Finite Elements or Diffuse Elements.

In collocation formulations based on second derivatives, the linear convergence of the terms $\dfrac{d^2u_{\text{app}}}{dx^2}(\mathbf{x}) = \sum_i N_i'(\mathbf{x})u_i$ appearing in the equilibrium equations, requires:

$$\begin{aligned} &\sum_i N_i'' = 0, \\ &\sum_i N_i''(x_i - x) = 0, \\ &\sum_i N_i'' \frac{(x_i - x)^2}{2} = 1. \end{aligned} \tag{17}$$

These conditions are automatically satisfied when derving shape functions by MLS [7] or by GFDM [9].

4. Consistency based approach

Rather than generate the shape functions and then analyze their properties, we now introduce an alternative technique of shape function construction explicitly based on the wished consistency conditions. This approach presents several advantages both from computational and from formal point of view. The obtained algorithm is efficient and supplementary conditions, different from consistency, may also be used.

Let us consider an evaluation point x and a set of associated nodes $\{x_1, \dots x_n\}$ close to x. We note

$$\mathbf{W} = \begin{bmatrix} w_1 & & 0 \\ & \ddots & \\ 0 & & w_n \end{bmatrix}.$$

We introduce the objective function

$$J(\mathbf{N}) = \frac{1}{2}\mathbf{N}^T\mathbf{W}^{-1}\mathbf{N}. \tag{18}$$

The shape functions $\mathbf{N}$ are solutions of Min $(J(\mathbf{N}))$ subjected to contraints (15).
The associated Lagrangian is

$$L(\mathbf{N}, \lambda) = \frac{1}{2}\mathbf{N}^T\mathbf{W}^{-1}\mathbf{N} + \lambda^T(\mathbf{P}\mathbf{N} - \mathbf{e}^1)$$

and the optimality conditions are

$$\mathbf{N}^T\mathbf{W}^{-1} + \lambda^T\mathbf{P} = 0, \tag{19}$$

$$\mathbf{P}\mathbf{N} - \mathbf{e}^1 = 0,$$

leading to the following linear system

$$\begin{bmatrix} \mathbf{W}^{-1} & \mathbf{P}^T \\ \mathbf{P} & 0 \end{bmatrix} \begin{Bmatrix} \mathbf{N} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} 0 \\ \mathbf{e}^1 \end{Bmatrix}. \tag{20}$$

The solution of (19) is given by $\mathbf{N} = -\mathbf{W}\mathbf{P}^T\lambda$, so $\mathbf{P}(-\mathbf{W}\mathbf{P}^T\lambda) = \mathbf{e}^1$, and

$$\lambda = -(\mathbf{P}\mathbf{W}\mathbf{P}^T)^{-1}\mathbf{e}^1.$$

and finally

$$\mathbf{N}^T = \mathbf{e}^{1T}(\mathbf{P}\mathbf{W}\mathbf{P}^T)^{-1}\mathbf{P}\mathbf{W} \tag{21}$$

where we recognize the matrices

$$\mathbf{A} = \mathbf{P}\mathbf{W}\mathbf{P}^T,$$

$$\mathbf{B} = \mathbf{P}\mathbf{W}.$$

It easy to see that (21) corresponds to the Moving Least Squares shape functions.

For the first derivatives of the shape functions one may also state that

$$\mathbf{N}' = \begin{Bmatrix} N_1'(x) \\ \vdots \\ N_n'(x) \end{Bmatrix}$$

is solution of $\mathrm{Min}(J_x(\mathbf{N}'))$ under the constraint (16) written as

$$\mathbf{PN}' = \left\{ \begin{array}{c} 0 \\ 0 \\ 1 \end{array} \right\} = \mathbf{e}^2$$

leading to the first and second diffuse derivatives of the shape functions

$$\mathbf{N}^T = \mathbf{e}^{2T}(\mathbf{PWP}^T)^{-1}\mathbf{PW},$$

$$\mathbf{N}^T = \mathbf{e}^{3T}(\mathbf{PWP}^T)^{-1}\mathbf{PW}$$

and we establish in this way that

$$N'(x) \equiv \frac{\delta N}{\delta x}(x),$$

$$N''(x) \equiv \frac{\delta^2 N}{\delta x^2}(x).$$

Finally we find again formula (13)

$$\left[\begin{array}{ccc} N_1(x) & \dots & N_n(x) \\ \dfrac{\delta N_1}{\delta x}(x) & \dots & \dfrac{\delta N_n}{\delta x}(x) \\ \dfrac{\delta^2 N_1}{\delta x^2} N_1(x) & \dots & \dfrac{\delta^2 N_n}{\delta x^2}(x) \end{array} \right] = \mathbf{A}(x)^{-1}\mathbf{B}.$$

It is important to note that

- the derivatives are obtained here as a result of the minimization of (18) subjected to different consistency contraints,
- these consistency based derivatives are the diffuse derivatives. It implies that the diffuse derivatives are sufficient for the convergence of the approximation.

This alternative presentation of the shape functions based on explicit constraints is also useful for taking into account other kinds of contraints.

5. Step by step algorithm for computing shape functions

The particular structure of the system (20) leads to an efficient algorithm of shape function computation without explicit matrix inversion.

While minimizing the expression (18)

$$\min\left(\frac{1}{2}\mathbf{N}^T\mathbf{W}^{-1}\mathbf{N}\right) \tag{22}$$

we separate the consraint

$$\sum_i N_i = \mathbf{e}^T\mathbf{N} = 1, \qquad \mathbf{e} = \left\{ \begin{array}{c} 1 \\ \vdots \\ 1 \end{array} \right\} \tag{23}$$

and the other constraints

$$\gamma^T \mathbf{N} = 0, \quad \gamma = \begin{Bmatrix} \vdots & \vdots \\ (x_i - x) & \dfrac{(x_i - x)^2}{2} \\ \vdots & \vdots \end{Bmatrix}. \tag{24}$$

Let us introduce a Lagrange multiplier λ for the constraint (23) and a vector of Lagrange multipliers $\boldsymbol{\mu}$ for the constraints (24)

$$\begin{bmatrix} \mathbf{W}^{-1} & \mathbf{e} & \gamma \\ \mathbf{e}^T & 0 & 0 \\ \gamma^T & 0 & 0 \end{bmatrix} \begin{Bmatrix} \mathbf{N} \\ \lambda \\ \boldsymbol{\mu} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix}. \tag{25}$$

The solution of the latter system is

$$\mathbf{N} = \frac{\tilde{\mathbf{W}}\mathbf{e}}{\mathbf{e}^T \tilde{\mathbf{W}} \mathbf{e}} \tag{26}$$

or, in index notation

$$N_i = \frac{\sum_j \tilde{W}_{ij}}{\sum_k \sum_l \tilde{W}_{kl}}, \tag{27}$$

where $\tilde{\mathbf{W}}$ stands for the modified weight matrix

$$\tilde{\mathbf{W}} = \mathbf{W}(\mathbf{I} - (\gamma^T \mathbf{W} \gamma)^{-1} \gamma \gamma^T \mathbf{W}). \tag{28}$$

The derivative are computed by the same algorithm, with the constraints

$$\sum_i N_i'(x_i - x) = \mathbf{N}^T(\mathbf{x}_i - \mathbf{x}) = 1, \tag{29}$$

$$\gamma'^T \mathbf{N}' = 0, \quad \gamma' = \begin{Bmatrix} \vdots & \vdots \\ 1 & \dfrac{(x_i - x)^2}{2} \\ \vdots & \vdots \end{Bmatrix} = \left[\mathbf{e} \ \frac{(\mathbf{x}_i - \mathbf{x})^2}{2} \right].$$

Leading to the system

$$\begin{bmatrix} \mathbf{W}^{-1} & (\mathbf{x}_i - \mathbf{x}) & \gamma' \\ (\mathbf{x}_i - \mathbf{x})^T & 0 & 0 \\ \gamma'^T & 0 & 0 \end{bmatrix} \begin{Bmatrix} \mathbf{N}' \\ \lambda \\ \boldsymbol{\mu} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix} \tag{30}$$

The derivatives of the shape function take the form

$$\mathbf{N}' = \frac{\tilde{\mathbf{W}}'(\mathbf{x}_i - \mathbf{x})}{(\mathbf{x}_i - \mathbf{x})^T \tilde{\mathbf{W}}'(\mathbf{x}_i - \mathbf{x})} \tag{31}$$

with the modified weight matrix

$$\tilde{\mathbf{W}}' = \mathbf{W}(\mathbf{I} - (\gamma'^T \mathbf{W} \gamma')^{-1} \gamma' \gamma'^T \mathbf{W}) \tag{32}$$

In order to avoid the inversion of the matrix $\gamma^T \mathbf{W} \gamma$ in the expressions of $\tilde{\mathbf{W}}$ and $\tilde{\mathbf{W}}'$, we shall adopt a step by step technique. First, we noitce that the different systems corresponding to the shape functions and to their derivatives are particular cases built from the following general system

$$\begin{bmatrix} \begin{bmatrix} \ddots & & 0 \\ & w_i & \\ 0 & & \ddots \end{bmatrix}^{-1} & \begin{Bmatrix} \vdots \\ 1 \\ \vdots \end{Bmatrix} & \begin{Bmatrix} \vdots \\ (x_i - x) \\ \vdots \end{Bmatrix} & \begin{Bmatrix} \vdots \\ \dfrac{(x_i - x)^2}{2} \\ \vdots \end{Bmatrix} & \cdots \\ \langle \ldots 1 \ldots \rangle & 0 & 0 & 0 & \ldots \\ \langle \ldots (x_i - x) \ldots \rangle & 0 & 0 & 0 & \ldots \\ \left\langle \ldots \dfrac{(x_i - x)^2}{2} \ldots \right\rangle & 0 & 0 & 0 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{Bmatrix} \begin{Bmatrix} \vdots \\ N_i \\ \vdots \end{Bmatrix} \\ \mu_1 \\ \mu_2 \\ \mu_3 \\ \vdots \end{Bmatrix} \tag{33}$$

$$= \begin{Bmatrix} \begin{Bmatrix} \vdots \\ 0 \\ \vdots \end{Bmatrix} \\ 1 \\ 0 \\ 0 \\ \vdots \end{Bmatrix}$$

by swapping the lines and the column corresponding to the multipliers μ_i in such a way to place in the first position the equation with a unit term on its right hand side. Following this procedure, the first derivative system 1-30 is obtained by the permutation $(\mu_2, \mu_1, \mu_3, \mu_4 \ldots)$. By analogy, the second derivative is obtained by $(\mu_3, \mu_1, \mu_2, \mu_4, \ldots)$. The same technique is also applicable in 2 and 3 dimensions. So, for calculating the shape functions and their derivatives, we propose an efficient algorithm to solve the system with k Lagrange multipliers and with the same RHS

$$\begin{bmatrix} \mathbf{W}^{-1} & \gamma'_1 & \ldots & \gamma'_k \\ \gamma'^T_1 & 0 & & 0 \\ \vdots & & \ddots & \\ \gamma'^T_k & 0 & & 0 \end{bmatrix} \begin{Bmatrix} \mathbf{N} \\ \boldsymbol{\mu}_1 \\ \vdots \\ \boldsymbol{\mu}_k \end{Bmatrix} = \begin{Bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{Bmatrix}.$$

The last multiplier μ_k may be eliminated by sunstituting

$$\mathbf{N} = -\mathbf{W} \left(\gamma'_k \mu_k + \sum_{i=1}^{k-1} \gamma'_i \mu_i \right)$$

Into the last equation

$$\gamma_k'^T \mathbf{N} = 0$$

The system (33) becomes then

$$\begin{bmatrix} \tilde{\mathbf{W}}_1^{-1} & \gamma_1' & \cdots & \gamma_{k-1}' \\ \gamma_1'^T & 0 & & 0 \\ \vdots & & \ddots & \\ \gamma_{k-1}'^T & 0 & & 0 \end{bmatrix} \begin{Bmatrix} \mathbf{N} \\ \mu_1 \\ \vdots \\ \mu_{k-1} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{Bmatrix}$$

with

$$\tilde{\mathbf{W}}_1 = \mathbf{W}(\mathbf{I} - (\gamma_k^T \mathbf{W} \gamma_k)^{-1} \gamma_k \gamma_k^T \mathbf{W})$$

or, more simply, the term $\gamma_k'^T \mathbf{W} \gamma_k$ being a scalar

$$\tilde{\mathbf{W}}_1 = \mathbf{W}\left(\mathbf{I} - \frac{\gamma_k \gamma_k^T \mathbf{W}}{\gamma_k^T \mathbf{W} \gamma_k}\right).$$

Repeating the procedure for all the multipliers corresponding to the zero RHS, we get a step by step algorithm for $\tilde{\mathbf{W}}$

$$\begin{aligned} &\tilde{\mathbf{W}}_0 = \mathbf{W}, \\ &\tilde{\mathbf{W}}_i = \tilde{\mathbf{W}}_{i-1}\left(\mathbf{I} - \frac{\gamma_i \gamma_i^T \mathbf{W}_{i-1}}{\gamma_i^T \mathbf{W}_{i-1} \gamma}\right), \quad i = 1, \ldots, k. \end{aligned} \tag{34}$$

We obtain finally

$$\mathbf{N} = \frac{\tilde{\mathbf{W}}_k \gamma_1}{\gamma_1^T \tilde{\mathbf{W}}_k \gamma_1}. \tag{35}$$

The shape functions are computed in this way without matrix inversion. The first example in the next section illustrates the algorithm for a simple 2-nodes linear finite element in one dimension.

Remark The calculations may be accelerated by setting

$$\boldsymbol{\eta}_i = \tilde{\mathbf{W}}_{i-1} \gamma_i.$$

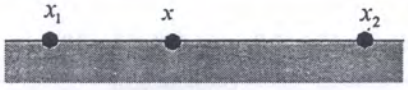
Now, the algorithm only contains products of vectors

$$\begin{aligned} \tilde{\mathbf{W}}_0 &= \mathbf{W}, \\ \tilde{\mathbf{W}} &= \tilde{\mathbf{W}}_{i-1} - \frac{\boldsymbol{\eta}_i \boldsymbol{\eta}_i^T}{\gamma_i^T \boldsymbol{\eta}_i}, \quad i = 1, \ldots, k, \\ \mathbf{N} &= \frac{\tilde{\mathbf{W}}_k \boldsymbol{\eta}_1}{\boldsymbol{\eta}_1^T \tilde{\mathbf{W}}_k \boldsymbol{\eta}_1}. \end{aligned} \tag{36}$$

6. EXAMPLES

6.1. Finite elements as special case

Finite element approximation is obtained as a special case of MLS, with constant weght functions $w_i = 1$ and with a number of nodes equal to the number of contraints. In one dimension and with 2 nodes



the system (33) becomes

$$\begin{bmatrix} 1 & 0 & 1 & (x_1 - x) \\ 0 & 1 & 1 & (x_2 - x) \\ 1 & 1 & 0 & 0 \\ (x_1 - x) & (x_2 - x) & 0 & 0 \end{bmatrix} \begin{Bmatrix} N_1 \\ N_2 \\ \mu_1[2ex]\mu_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{Bmatrix}$$

whose solution corresponds to usual shape functions for linear finite elements

$$N_1 = \frac{x - x_2}{x_1 - x_2}, \quad N_2 = \frac{x - x_1}{x_2 - x_1}.$$

The diffuse derivates are obtained by swapping multipliers μ_1 and μ_2

$$\begin{bmatrix} 1 & 0 & (x_1 - x) & 1 \\ 0 & 1 & (x_2 - x) & 1 \\ (x_1 - x) & (x_2 - x) & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \frac{\delta N_1}{\delta x} \\ \frac{\delta N_2}{\delta x} \\ \mu_1 \\ \mu_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{Bmatrix}.$$

They are equivalent to the standard derivatives of the shape functions (diffuse and full derivatives are equal in this particular case)

$$\frac{\delta N_1}{\delta x} = \frac{1}{x_1 - x_2} = \frac{dN_1}{dx}, \quad \frac{\delta N_2}{\delta x} = \frac{1}{x_2 - x_1} = \frac{dN_2}{dx}.$$

6.2. Performance of shape functions evaluation

The performance ,easures of the recursive algorithm (36) are presented in Table 1 along with those of the usual algorithm (13). Averaged timings for evaluating N_i, $\delta N_i/\delta x$, $\delta N_i/\delta y$, are given in miliseconds for different numbers of nodes ($n = 3, 5, 7, 9, 12$). For algorithm (13), the timings for computing N_i, $\partial N_i/\partial x$, $\partial N_i/\partial y$ are also given. The timings were obtained running Matlab 5 on a Pentium 266MHz computer.

The values in Table 1 show, that for connectivities useful in practical applications, the mean gain is about 50% as compared with the standard algorithm.

Table 1. Performance measurements for shape function computation using various algorithms

Computation of:	$n=3$	$n=5$	$n=7$	$n=9$	$n=12$
$N_i, \frac{\delta N_i}{\delta x}, \frac{\delta N_i}{\delta y}$; algorithm 1–36	2.4 ms	3.1 ms	3.6 ms	4.1 ms	4.7 ms
$N_i, \frac{\delta N_i}{\delta x}, \frac{\delta N_i}{\delta y}$; algorithm 1–13	3.1 ms	6.1 ms	7.6 ms	8.9 ms	10.6 ms
$N_i, \frac{\delta N_i}{\delta x}, \frac{\delta N_i}{\delta y}$; algorithm 1–13	5.2 ms	8.0 ms	10.1 ms	11.8 ms	14.3 ms

6.3. Convergence of the approximation

Given the displacement values $\mathbf{u}$ at nodes $\mathbf{x}_i$, one may get the strains at any evaluation point $\mathbf{x}$ by

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_x(x) \\ \varepsilon_y(x) \\ \gamma_{xy}(x) \end{Bmatrix} = [\mathbf{B}(x)] \begin{Bmatrix} \vdots \\ u(x_i) \\ \nu(x_i) \\ \vdots \end{Bmatrix}.$$

With the operation $\mathbf{B}$ well known in the finite element method

$$[\mathbf{B}] = \begin{bmatrix} & N_{i,x} & 0 & \\ \dots & 0 & N_{i,y} & \dots \\ & N_{i,y} & N_{i,x} & \end{bmatrix}.$$

The stresses are then given by the constitutive relation $\{\boldsymbol{\sigma}(x)\} = [\mathbf{D}\{\varepsilon(x)\}$. The convergence of the operator $\mathbf{B}$ is required for any method of solving governing equations (Diffuse Elements, Element Free Galerkin, Collocation, ...).

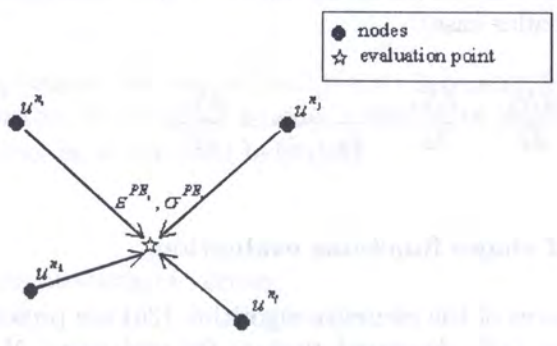


Fig. 4. Computing stresses and strains at an evaluation point

In the following two examples we compare the convergence of stress functions on regular and irregular grids using $[B(x)]$ buil from both "full" and "diffuse" derivatives of shape functions generated from linear consistency constraints.

6.4. Cantilever beam

The analytical solution of the following cantilever beam problem

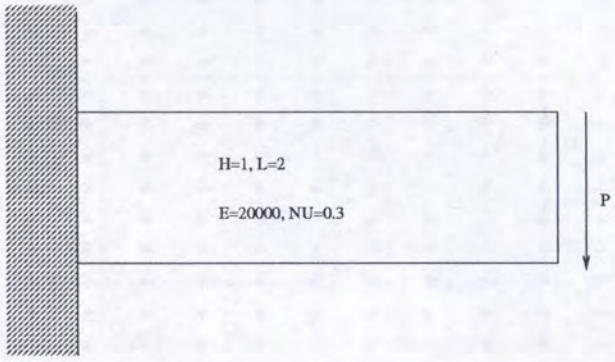


Fig. 5. Cantilever beam

is given by Timoshenko [17]

$$u = \frac{-Py}{EI}\left(Lx - \frac{x^2}{2}\right) + \frac{P}{I}\left(\frac{\boldsymbol{\nu}}{E} - \frac{1}{G}\right)\frac{y^3}{6}$$

$$\nu = \frac{P}{EI}\left(\left(L - \frac{x}{3}\right)\frac{x^2}{2} + \boldsymbol{\nu}(L-x)\frac{y^2}{2}\right) + \frac{PH^2x}{8GI}$$

$$G = \frac{E}{2(1+\boldsymbol{\nu})}$$

$$\boldsymbol{\sigma}_x = \frac{-Py(L-x)}{I}, \qquad \boldsymbol{\sigma}_y = 0, \qquad \tau = \frac{P}{2I}\left(\frac{H^2}{4} - y^2\right)$$

The following tests study the convergence of the approximations of displacements and stresses while increasing the density of a constant pattern of nodes and of evaluation points. In the first example a regular pattern is used (Figure 6).

The results reported in Figure 7 are given in a relative norm error$= \dfrac{\|\mathbf{u}_{ex} - \mathbf{u}_{app}\|}{\|\mathbf{u}_{ex}\|}$ and were obtained for a Poisson ratio de 0.3, a unit pressure. The approximation at the evaluation points was performed using 5 closest nodes.
We note that:

- the "diffuse" stresses and displacements have the same numerical stability,
- the "diffuse" derivatives are more stable than the "exact" ones
- the starting values of stresses are slightly worse for $\delta u/\delta x$ than those of stresses computed with $\partial u/\partial x$,
- the rate of convergence is the same in both cases,
- the roundoff errors extremely small h seem to more affect the "exact" stresses.

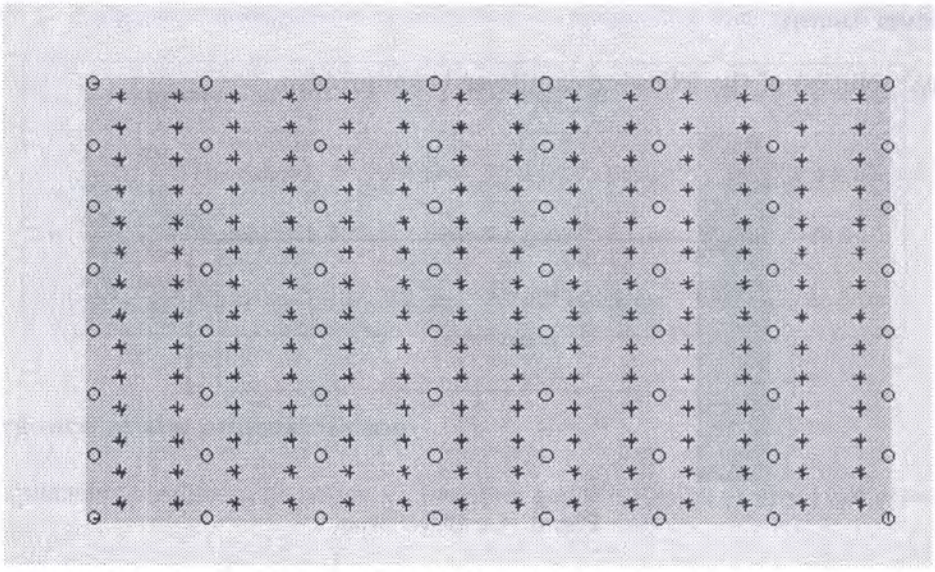


Fig. 6. Regular pattern of nodes (circles) and evaluation points (stars) used for the cantilever beam example

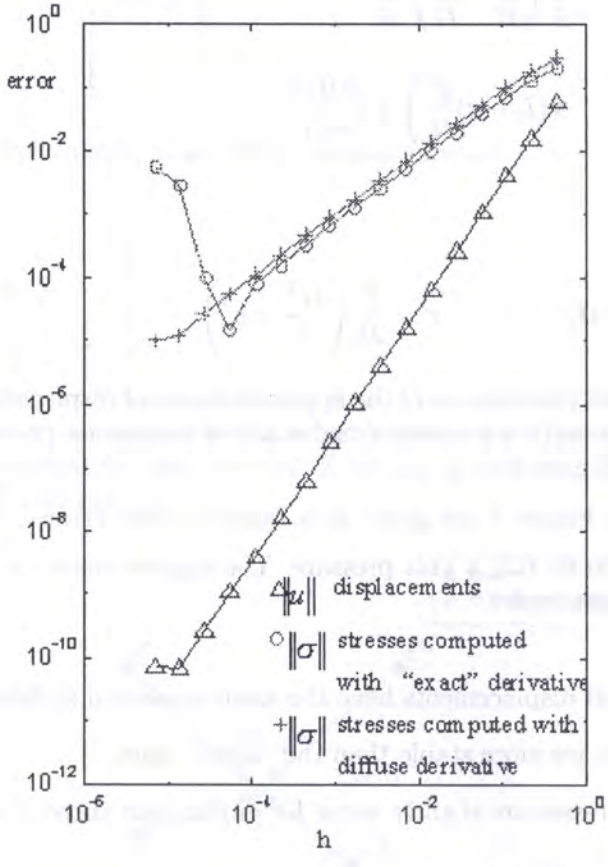


Fig. 7. Convergence of the approximation of displacements at nodes and of stresses at evaluation points using the relative norm

6.5. Plate with a hole

An infinite plate with a hole of radius a subject to traction $\boldsymbol{\sigma}_0$

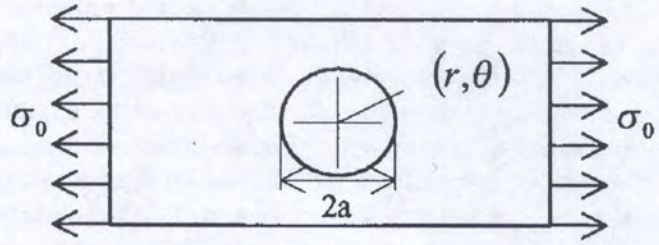


Fig. 8. Plate with a hole problem

The analytical solution was given by [17]

$$\boldsymbol{\sigma}_{xx} = \boldsymbol{\sigma}_0 \left\{ 1 - \frac{a^2}{r^2} \left[\frac{3}{2}\cos(2\theta) + \cos(4\theta) \right] + \frac{3a^4}{2r^4}\cos(4\theta) \right\},$$

$$\boldsymbol{\sigma}_{yy} = -\boldsymbol{\sigma}_0 \left\{ \frac{a^2}{r^2} \left[\frac{1}{2}\cos(2\theta) - \cos(4\theta) \right] + \frac{3a^4}{2r^4}\cos(4\theta) \right\},$$

$$\boldsymbol{\sigma}_{xy} = -\boldsymbol{\sigma}_0 \left\{ \frac{a^2}{r^2} \left[\frac{1}{2}\sin(2\theta) + \sin(4\theta) \right] - \frac{3a^4}{2r^4}\sin(4\theta) \right\},$$

$$u_r = \frac{\boldsymbol{\sigma}_0}{4G} \left\{ r \left[\frac{\kappa - 1}{2} + \cos(2\theta) \right] + \frac{a^2}{r^2}[1 + (1+\kappa)\cos(2\theta)] - \frac{a^4}{r^3}\cos(2\theta) \right\},$$

$$u_\theta = \frac{\boldsymbol{\sigma}_0}{4G} \left[(1-\kappa)\frac{a^2}{r^2} - r - \frac{a^4}{r^3} \right] \sin(2\theta)$$

where

$$G = \frac{E}{2(1+\nu)}, \qquad \kappa = \begin{cases} 3 - 4\nu & \text{in plane strain} \\ \dfrac{3-\nu}{1+\nu} & \text{in plane stress} \end{cases}$$

and (r, θ) are the polar coordinates with origin at the center of the hole. In this example we focus on the convergence of the approximation on a random pattern of nodes. We use regular grid of evaluation points (Figure 9). Figure 10 illustrates the results obtained while refining this pattern. As in the previous example, the results reported in Figure 10 correspond to a relative norm and were obtained for a Poisson ratio de 0.3, a unit pressure $\boldsymbol{\sigma}_0$. The approximation at the evaluation points was performed using 7 closest nodes. We note, that for this strongly irregular pattern of nodes, the diffuse derivative performs slightly better than the "exact" one. The numerical stability of both "diffuse" and "exact" stresses is the same.

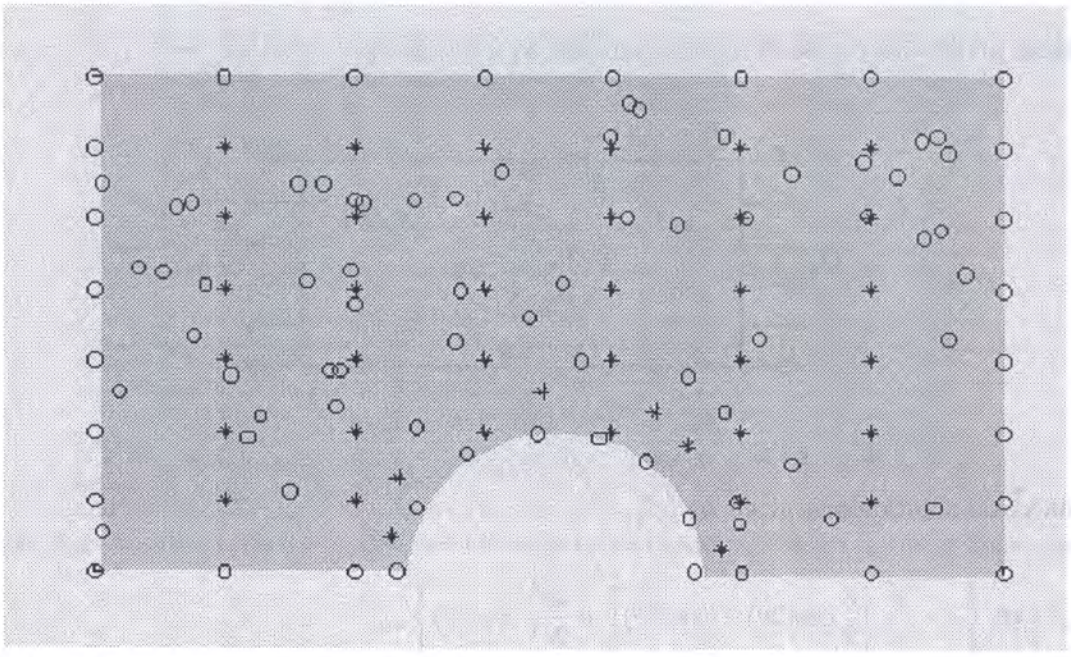


Fig. 9. Random nodal pattern (circles) and regular grid of evaluation points (stars) used for the plate with a hole problem

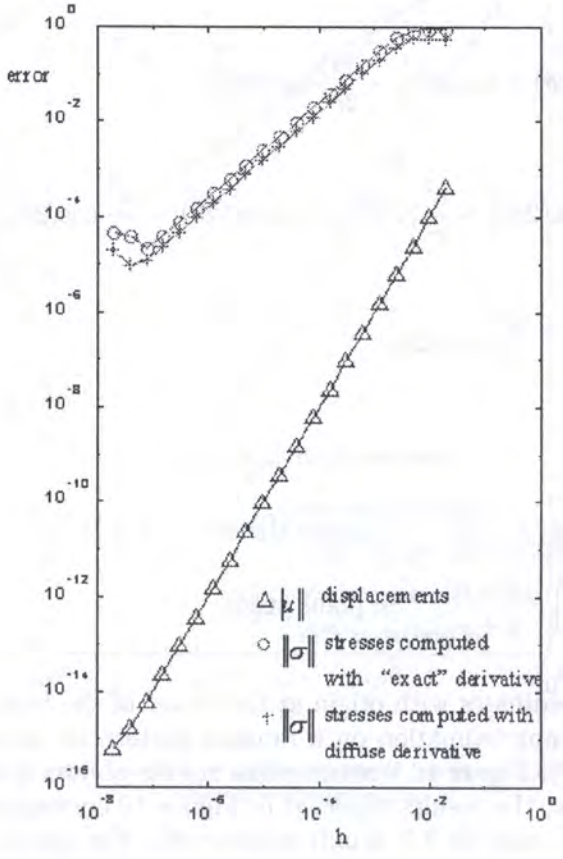


Fig. 10. Convergence results for a plate with a hole

7. Conclusion

The consistency approach to the construction of shape functions leads to more effective algorithms by exploiting the particular form of the equation set. The numerical results being similar, one may see the proposed method as an alternative to the generalized finite differences. The proposed framework is however more general in the sense that it permits to introduce supplementary constaints on the shape functions, other than consistency. One has to be conscious that in variational formulations consistency is not sufficient. Future work will have to address such issues as the validity of the integration by parts technique for inexact derivatives, the Green formula, numerical integration... . These problems do not arise in the methods based on collation. In these methods, the interest of the diffuse derivative resides in its high computational efficiency especially for high order derivatives and its good numerical stability. From the programming point of view it seems worthwhile to keep the two possibilities in the code and to leave the choice to the user, keeping in mind that the cost of obtaining the "exact" derivatives is significantly higher.

Appendix: convergence of centered MLS

We give here the demonstration of convergence by [18] in 2D. The work [19] may be also consulted. Let us consider once again the centered scheme 1–8

$$\mathbf{J}_x(\boldsymbol{\alpha}) = \frac{1}{2}\sum_i w_i(\mathbf{x}_i, \mathbf{x})(\mathbf{q}^T(\mathbf{x}_i - \mathbf{x})\boldsymbol{\alpha} - u_i)^2 \tag{A.1}$$

taking into account k terms we have

$$\mathbf{q}^T(\mathbf{x}_i - \mathbf{x}) = \left(1, x_i - x, y_i - y, \frac{1}{2}(x_i - x)^2, (x_i - x)(y_i - y), \frac{1}{2}(y_i - y)^2, \ldots,\right.$$

$$\left.\frac{1}{k!}(x_i - x)^k, \ldots, \frac{(x_i - x)^l}{l!}\frac{(y_i - y)^{k-1}}{(k-l)!}, \ldots, \frac{1}{k!}(y_i - y)^k\right)$$

we assume, as previously, that $u_i = u_{\text{ex}}(\mathbf{x}_i)$, where $u_{\text{ex}}(\mathbf{x})$ is $k+1$ times continously differentiable. The Taylor formula may be written

$$\mathbf{u}_{\text{ex}}^T(\mathbf{x}) = \mathbf{q}^T(\mathbf{x}_i - \mathbf{x})\mathbf{U}_{\text{ex}}(\mathbf{x}) + r(\mathbf{x}_i, \mathbf{x})$$

with

$$\mathbf{U}_{\text{ex}}^T(\mathbf{x}) = \left(u_{\text{ex}}(\mathbf{x}), \frac{\partial u_{\text{ex}}}{\partial x}, \frac{\partial u_{\text{ex}}}{\partial y}, \frac{\partial^2 u_{\text{ex}}}{\partial x^2}, \ldots, \frac{\partial^2 u_{\text{ex}}}{\partial x^2}, \ldots, \frac{\partial^k u_{\text{ex}}}{\partial x^k}, \frac{\partial^k u_{\text{ex}}}{\partial x^l \partial y^{k-l}}, \ldots, \frac{\partial^k u_{\text{ex}}}{\partial y^k}\right)$$

and

$$r(\mathbf{x}_i, \mathbf{x}) = (k+1)\sum_{l=0}^{k+1}\frac{(x_i - x)^l}{l!}\frac{(y_i - y)^{k-l+1}}{(k-l)!}\int_0^1 \frac{\partial^{k+1} u_{\text{ex}}}{\partial x^l \partial y^{k-l}}(\mathbf{x} + t(\mathbf{x}_i - \mathbf{x}))(1-t)^k dt \tag{A.2}$$

(A.1) becomes

$$\mathbf{J}_x(\alpha) = \frac{1}{2}\sum_i w_i(\mathbf{x}_i, \mathbf{x})(\mathbf{q}^T(\mathbf{x}_i - \mathbf{x})(\mathbf{U}_{\text{ex}} - \alpha) + r(\mathbf{x}_i\mathbf{x}))^2 \tag{A.3}$$

$$= \frac{1}{2}(\mathbf{U}_{\text{ex}} - \alpha)^T\mathbf{A}(\mathbf{x})(\mathbf{U}_{\text{ex}} - \alpha) - \mathbf{b}(\mathbf{x})^T(\mathbf{U}_{\text{ex}} - \alpha) + \text{rest}$$

where

$$\mathbf{A}(\mathbf{x}) = \sum_i w(\mathbf{x}_i - \mathbf{x})\mathbf{q}(\mathbf{x}_i - \mathbf{x})\mathbf{q}^T(\mathbf{x}_i - \mathbf{x}),$$

$$\mathbf{b}(\mathbf{x}) = \sum_i w(\mathbf{x}_i - \mathbf{x})r(\mathbf{x}_i, \mathbf{x})\mathbf{q}(\mathbf{x}_i - \mathbf{x}).$$

The optimum of (A.3) is then attained for $\boldsymbol{\alpha}$ such that

$$\mathbf{A}(\mathbf{x})(\mathbf{U}_{\text{ex}} - \boldsymbol{\alpha}) = \mathbf{b}(\mathbf{x}) \tag{A.4}$$

or

$$(\mathbf{U}_{\text{ex}} - \boldsymbol{\alpha}) = \mathbf{A}^{-1}(\mathbf{x})\mathbf{b}(\mathbf{x}) = \sum_i r(\mathbf{x}_i, \mathbf{x})w(\mathbf{x}_i, \mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{q}(\mathbf{x}_i - \mathbf{x})$$

what may be noted

$$(\mathbf{U}_{\text{ex}} - \boldsymbol{\alpha}) = [N(\mathbf{x})]\{r(\mathbf{x})\} \tag{A.5}$$

where

$$[N(\mathbf{x})] = \begin{bmatrix} N_1(\mathbf{x}) & \dots & N_n(\mathbf{x}) \\ \dfrac{\partial N_1}{\partial x}(\mathbf{x}) & & \dfrac{\partial N_n}{\partial x}(\mathbf{x}) \\ \vdots & & \vdots \\ \dfrac{\partial^k N_1}{\partial y^k}(\mathbf{x}) & \dots & \dfrac{\partial^k N_n}{\partial y^k}(\mathbf{x}) \end{bmatrix}$$

and

$$r(\mathbf{x})^T = (r(\mathbf{x}_1, \mathbf{x}), \dots, r(\mathbf{x}_n, \mathbf{x})).$$

In order to establish convergence we consider the expression (A.5) taking ν a radius of a sphere contaning the nodes x_i. We take $\theta = (\xi_i, \eta_i)$ with

$$\xi_i = (x_i - x)/\nu$$

$$\eta_i = (y_i - y)/\nu,$$

so $\nu\boldsymbol{\theta}_i = (\mathbf{x}_i - \mathbf{x})$

$$\mathbf{q}(\mathbf{x}_i - \mathbf{x})^T = \mathbf{q}(\boldsymbol{\theta}_i)[D],$$

where

$$D = \begin{bmatrix} 1 & & & & & & \\ & \nu & & & & 0 & \\ & & \nu & & & & \\ & & & \nu^2 & & & \\ & & & & \nu^2 & & \\ & & & & & \nu^2 & \\ & 0 & & & & & \ddots \\ & & & & & & & \nu^k \end{bmatrix}$$

and

$$\mathbf{A}(\mathbf{x}) = \mathbf{D}\mathbf{A}(\boldsymbol{\theta})\mathbf{D},$$
$$\mathbf{b}(\mathbf{x}) = \mathbf{D}\mathbf{b}(\boldsymbol{\theta})$$

then (A.4) becomes

$$\mathbf{D}(\mathbf{U}_{\text{ex}} - \boldsymbol{\alpha}) = \mathbf{A}^{-1}(\boldsymbol{\theta})\mathbf{b}(\boldsymbol{\theta}) \tag{A.6}$$

but, (A.2) transforms to

$$r(\mathbf{x}_i, \mathbf{x}) = \boldsymbol{\nu}^{k+1} \sum_{l=0}^{k+1} \frac{\xi_i^l}{l1} \frac{\eta_i^{k-l+1}}{(k-l)!} \int_0^1 (1+k)(1-t)^k \frac{\partial^{k+1} u_{\text{ex}}}{\partial x^l \partial y^{k-l}}(\mathbf{x} + t(\mathbf{x}_i - \mathbf{x}))dt = \boldsymbol{\nu}^{k+1} r(\boldsymbol{\theta}_i)$$

and we obtain from (A.6)

$$(\mathbf{U}_{\text{ex}} - \boldsymbol{\alpha}) = \begin{bmatrix} \boldsymbol{\nu}^{k+1} & & & & \\ & \boldsymbol{\nu}^k & & 0 & \\ & & \boldsymbol{\nu}^k & & \\ & & & \boldsymbol{\nu}^{k-1} & \\ & 0 & & & \ddots \\ & & & & \boldsymbol{\nu} \end{bmatrix} \mathbf{A}^{-1}(\boldsymbol{\theta})\mathbf{d}(\boldsymbol{\xi}) \tag{A.7}$$

where

$$\mathbf{d}(\boldsymbol{\theta}) = \sum_i w(\mathbf{x}_i, \mathbf{x}) \cdot (\boldsymbol{\theta}_i)\mathbf{q}(\boldsymbol{\theta}_i).$$

We may introduce the non dimensional shape functions by the same technique that was used in (A.6)

$$\begin{Bmatrix} n_i(\boldsymbol{\theta}) \\ \dfrac{\partial n_i}{\partial x}(\boldsymbol{\theta}) \\ \vdots \\ \dfrac{\partial^k n_i}{\partial x^k}(\boldsymbol{\theta}) \end{Bmatrix} = w(\mathbf{x}_i, \mathbf{x})\mathbf{A}^{-1}(\boldsymbol{\theta})\mathbf{q}(\boldsymbol{\theta}_i)$$

(A.7) may be written

$$(\mathbf{U}_{\text{ex}} - \boldsymbol{\alpha}) = \begin{bmatrix} \boldsymbol{\nu}^{k+1} & & & & \\ & \boldsymbol{\nu}^k & & 0 & \\ & & \boldsymbol{\nu}^k & & \\ & & & \boldsymbol{\nu}^{k-1} & \\ & 0 & & & \ddots \\ & & & & \boldsymbol{\nu} \end{bmatrix} \mathbf{n}(\boldsymbol{\xi}) \begin{Bmatrix} r(\boldsymbol{\theta}_1) \\ \vdots \\ r(\boldsymbol{\theta}_n) \end{Bmatrix} \tag{A.8}$$

where

$$\mathbf{n}(\boldsymbol{\xi}) = \begin{bmatrix} \mathbf{n}_1(\boldsymbol{\theta}) & \dots & \mathbf{n}_n(\boldsymbol{\theta}) \\ \dfrac{\partial \mathbf{n}_1}{\partial x}(\boldsymbol{\theta}) & & \dfrac{\partial \mathbf{n}_n}{\partial x}(\boldsymbol{\theta}) \\ \vdots & & \vdots \\ \dfrac{\partial^k \mathbf{n}_1}{\partial y^k}(\boldsymbol{\theta}) & \dots & \dfrac{\partial^k \mathbf{n}_n}{\partial y^k}(\boldsymbol{\theta}) \end{bmatrix}$$

Bounding of $r(\theta_i)$

We put

$$D_l^{k+1} u_{\text{ex}} = \underset{\theta \in S(\mathbf{x},\boldsymbol{\nu})}{\text{Max}} \left| \frac{\partial^{k+1} u_{\text{ex}}}{\partial x^l \partial y^{k-l}}(\boldsymbol{\theta}) \right|$$

where $S(\mathbf{x}, \boldsymbol{\nu})$ is a sphere of center $\mathbf{x}$ and radius $\boldsymbol{\nu}$ (for a chosen topology) we obtain then

$$\left| \int_0^1 (1+k)(1-t)^k \frac{\partial^{k+1} u_{\text{ex}}}{\partial x^l \partial y^{k-l}}(\mathbf{x} + t(\mathbf{x}_i - \mathbf{x}))dt \right| \leq D_l^{k+1} u_{\text{ex}} \tag{A.9}$$

on the other hand $|\xi_i|, |\eta_i| \leq 1$ so

$$r(\theta_i) \leq \sum_{l=0}^{k+1} \frac{1}{l!(k-l)!} D_l^{k+1} u_{\text{ex}} \overset{\text{def}}{=} D^{k+1} u_{\text{ex}} \tag{A.10}$$

Bounding of (A.1) line by line

For the first line (A.1) and taking into account (A.9), we have

$$|u_{\text{ex}}(\mathbf{x}) - \boldsymbol{\alpha}_0(\mathbf{x})| \leq \boldsymbol{\nu}^{k+1} \left(\sum_i |\mathbf{n}_i(\boldsymbol{\theta})| \right) D^{k+1} u_{\text{ex}}$$

for the second line

$$\left| \frac{\partial u_{\text{ex}}}{\partial x}(\mathbf{x}) - \boldsymbol{\alpha}_1(\mathbf{x}) \right| \leq \boldsymbol{\nu}^k \left(\sum_i \left| \frac{\partial \mathbf{n}_i}{\partial x}(\boldsymbol{\theta}) \right| \right) D^k u_{\text{ex}}$$

and for any subsequent line: $L(m,l)$

$$\left| \frac{\partial^m u_{\text{ex}}}{\partial x^l \partial y^{m-l}}(\mathbf{x}) - \boldsymbol{\alpha}_{L(m,l)}(\mathbf{x}) \right| \leq \boldsymbol{\nu}^{k+1-m} \left(\sum_i \left| \frac{\partial^m \mathbf{n}_i}{\partial x^l \partial y^{m-l}}(\boldsymbol{\theta}) \right| \right) D^{k+1} u_{\text{ex}} \tag{A.11}$$

where the line index $L(m,l)$ is given for the reason, that for the global derivation order m, there are $m+1$ terms of partial derivatives to be calculated, so

$$L(m,l) = 1 + 2 + \ldots + m + m - l + 1, \quad l = 0 \ldots m$$

The total number of lines of the matrix $\mathbf{n}(\boldsymbol{\theta})$ is $\frac{1}{2}(k+1)(k+2)$.

Example

We take a quadratic polynomial basis $p(x)^T = (1 \;\; x \;\; y \;\; x^2 \;\; xy \;\; y^2)$. We have 6 coefficients α_i and the error formula gives for (A.10)

$$D^3 u_{\text{ex}} = \frac{1}{6}(D_0^3 u_{\text{ex}} + 3D_1^3 u_{\text{ex}} + 3D_2^3 u_{\text{ex}})$$

and the formulas (A.11) may be written

$$|u_{\mathrm{ex}}(\mathbf{x}) - \boldsymbol{\alpha}_0(\mathbf{x})| \leq \boldsymbol{\nu}^3 \left(\sum_i |\mathbf{n}_i(\boldsymbol{\theta})| \right) D^3 u_{\mathrm{ex}},$$

$$\left| \frac{\partial u_{\mathrm{ex}}}{\partial x}(\mathbf{x}) - \boldsymbol{\alpha}_1(\mathbf{x}) \right| \leq \boldsymbol{\nu}^2 \left(\sum_i \left| \frac{\partial \mathbf{n}_i}{\partial x}(\boldsymbol{\theta}) \right| \right) D^3 u_{\mathrm{ex}},$$

$$\left| \frac{\partial^2 u_{\mathrm{ex}}}{\partial x^2}(\mathbf{x}) - \boldsymbol{\alpha}_5(\mathbf{x}) \right| \leq \boldsymbol{\nu} \left(\sum_i \left| \frac{\partial^2 \mathbf{n}_i}{\partial x^2}(\boldsymbol{\theta}) \right| \right) D^3 u_{\mathrm{ex}}.$$

Remark the conditioning of $\eta(\xi)$ depends on the topological configuration of nodes (Figures 1, 2 and 3).

REFERENCES

[1] T. Belytschko, L. Gu, Y.Y. Lu. Element-free Galerkin methods. *Int. Journ. Num. Meth. Engrg.*, **37**: 229–256, 1994.

[2] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, P. Krysl. Meshless Methods: An overview and recent developments. *Comput. Methods Appl. Mech. Engrg.* **139**, 1996.

[3] Breitkopf P., Touzot G., Villon P. *Explicit form and efficient computation of MLS shape functions and their derivatives*, submitted to IJNME.

[4] C.A. Duarte, J.T. Oden. H-p clouds: an h-p meshless method. *Num. Meth. Partial Diff. Equa.*, **12**: 673–706, 1996.

[5] C.A. Duarte, J.T. Oden. An h-p adaptive method using clouds. *Comput. Meth. Applied Mech. Engrg.* **139**: 237–262, 1996.

[6] J. Krok, J. Orkisz. A unified Approach to the FE and Generalized Variational FD in Nonlinear Mechanics, Concepts and Numerical Approach. *Int. Symp. on Discretization Methods in Structural Mechanics IUTAM/ IACM*, Vienna, Austria, 1989, Springer-Verlag, Berlin-Heidelberg, 353–362, 1990.

[7] P. Lancaster, K. Salkauskas. Surfaces generated by moving least squares methods. *Math Comput.* **37**: 155, 1981.

[8] T. Liszka, J. Orkisz. Finite Difference Method of Arbitrary Irregular Meshes in Non-Linear Problems of Applied Mechanics. *4th Int. Conf. on Structural Mechanics in Reactor Technology*, San Francisco, California, 1977.

[9] T. Liszka, J. Orkisz. Finite difference method at arbitrary irregular grids and its application in applied mechanics. *Comput. Struct.* **11**, 1980.

[10] T. Liszka. An interpolation method for an irregular net nodes. *IJNME*, **20**: 1599–1612, 1984.

[11] Yu Xie Mukherjee and S. Mukherjee. The boundary node method for potential problems. *Int. Journ. Num. Meth. Engrg.* **40**: 797–815, 1997.

[12] B. Nayroles, G. Touzot, P. Villon. Generalizing the finite element method: Diffuse approximation and diffuse elements. *Comput. Mech.*, 10, 1992.

[13] E. Oñate, S. Idelsohn, O.C. Zienkiewicz, L. Taylor, C. Sacco. A stabilized finite point method for analysis of fluid mechanics problems. *Comput. Meth. Applied Mech. Engrg.*, **139**: 315–346, 1996.

[14] J. Orkisz. Meshless finite difference method. I basic approach. *Proc. of the IACM-Fourth World Congress on Computational Mechanics*, Buenos Aires, 1998.

[15] J. Orkisz. Meshless finite difference method. II Adaptative approach. *Proc. of the IACM-Fourth World Congress on Computational Mechanics*, Buenos Aires, 1998.

[16] M. Syczewski, R. Tribillo. Singularities of sets used in the mesh method. *Computer and Structures*, **14**(5–6): 509–511, 1981.

[17] S.P. Timoshenko, J.N. Goodier. Theory of elasticity. 3rd edition, McGraw-Hill, New York, 1987.

[18] P. Villon. Contribution à l'Optimisation UTC, 1991.

[19] M.J. Wyatt, G. Davies, C. Snell. Truction error control in generalized finite element method. *J. Engng Mech. Div., Proc. ASCE*, EM4, 736–741, 1976.