INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2002; **54**:987–1006 (DOI: 10.1002/nme.455)

The partition of unity quadrature in meshless methods

A. Carpinteri, G. Ferro and G. Ventura*,†

Politecnico di Torino, Department of Structural Engineering and Geotechnics, Corso Duca degli Abruzzi 24, 10129 Torino, Italy

SUMMARY

In dealing with mesh-free formulations a major problem is connected to the computation of the quadratures appearing in the variational principle related to the differential boundary value problem. These integrals require, in the standard approach, the introduction of background quadrature subcells which somehow make these methods not 'truly meshless'. In this paper a new general method for computing definite integrals over arbitrary bounded domains is proposed, and it is applied in particular to the evaluation of the discrete weak form of the equilibrium equations in the framework of an augmented Lagrangian element-free formulation. The approach is based on splitting the integrals over the entire domain into the sum of integrals over weight function supports without modifying in any way the variational principle or requiring background quadrature cells. The accuracy and computational cost of the technique compared to standard Gauss subcells quadrature are discussed. Copyright © 2002 John Wiley & Sons, Ltd.

KEY WORDS: element free; meshless; quadrature; partition of unity; augmented Lagrangian

1. INTRODUCTION

The problem of the spatial integration in element-free methods is probably one of the major difficulties in the application of this class of methods. In fact, these methods present substantial advantages over the classical finite elements but, on the other hand, the shape functions constructed by meshless formulations have, in general, a far more complicated mathematical structure compared to the finite elements and are therefore more difficult to integrate. The problem of quadrature of the weak form is commonly resolved by introducing a background mesh of integration cells where Gauss quadrature is employed. At the same time, this cell

*Correspondence to: G. Ventura, Politecnico di Torino, Department of Structural Engineering and Geotechnics, Corso Duca degli Abruzzi 24, 10129 Torino, Italy
†E-mail: gventura@polito.it

Contract/grant sponsor: C.N.R., M.U.R.S.T.
Contract/grant sponsor: E.C.–T.M.R.; contract/grant number: ERBFMRXCT960062

Received 10 December 2000
Revised 25 June 2001

structure describes the physical domain of the body. This approach, while quite effective in practical applications, has the drawback of introducing a mesh in the discretization of the problem. This mesh is not related in any way to the approximation of the solution, nor does it impair the characteristics of the meshless approximation, such as the insensitivity to distortion in large deformation problems, the continuity and differentiability of the solution or the high convergence rate. Two different approaches to eliminating the necessity of introducing a background quadrature mesh have been proposed in the literature:

- the use of nodal integration, where the quadrature points are assumed coincident with the discretization nodes and the variational principle is modified so that the numerical integration is able to pass the patch test by adding a residual of the equilibrium equations [1] or by strain-smoothing stabilization [2];
- the modification of the variational principle into a local weak form, defined on the support of each weight function that converts the integration from the global domain into a series of integrations over the supports [3, 4].

The approach presented in this paper follows a rather different conceptual path, and is based on the partition of unity property of the moving least-squares shape functions [5]. This property states that, at an arbitrary point of the domain, the sum of all the shape functions is equal to one. As the MLS shape function for the Ith node differs only from zero on the support of its weight function, the integral over the whole domain can be replaced by the sum of the integrals over each weight function support, each computed with a quadrature weight function given by the corresponding MLS shape function. This quadrature scheme, relying on the partition of unity property of the shape functions, will be called the partition of unity quadrature (PUQ).

It must be noted that even if the quadrature is performed at each weight function support, as in the MLPG method of Atluri and Zhu [3, 4], the present method is conceptually very different. This is because the variational principle is not affected by the quadrature method, and the resulting system of equilibrium equations is banded and symmetric, in contrast to MLPG, which has a non-symmetrical structure.

The paper is structured as follows. In Section 2, the moving least-squares approximation is briefly illustrated and, in Section 3, the variational formulation of the elastic solid problem is presented using the augmented Lagrangian element-free (ALEF) method introduced in Reference [6]. This method is equivalent, from the approximation point of view, to the EFG method proposed by Belytschko *et al.* [7], but uses an augmented Lagrangian variational principle for enforcing the essential boundary condition and any other constraints [8–10]. It results in an iterative, fast algorithm characterized by a symmetric, positive-definite and banded system of equations whose dimension is equal to the number of approximation variables (i.e. to the number of nodes in the spatial discretization times the dimension of the space where the problem is set), independent of the number of constraints. The algorithm is globally convergent [11, 12] in very few iterations, consuming a negligible fraction of the total computing time [6]. In Section 4, the partition of unity quadrature approach is demonstrated. In Section 5, the results of the patch test for both regular and irregular nodal arrangements, and a Timoshenko beam problem [7, 13] are presented. The results are analysed in comparison with the classical subcell Gauss quadrature and the results obtained on the same tests by Belytschko *et al.* [7] and Atluri and Zhu [4]. A final example with a larger number of nodes and some concluding remarks conclude the section.

2. MLS APPROXIMATION

In this section, a brief introduction to the moving least-squares (MLS) approximation is reported. The approximation of a scalar function $\delta(x)$ in a domain Ω is considered. The technique uses a set of nodal points, a set of weight functions, and a set of basis functions.

At each point of the domain, a function is approximated by a linear combination of the basis functions. A set of nodes x_I, $I = 1 \ldots N$ is considered, and the linear combination coefficient for the node I is denoted by η_I. The coefficients η_I differ from point to point and are computed by minimizing a local least-squares form. This is accomplished by associating a weight function $w_I = w(x - x_I)$ with the compact support Ω_I to each node of the domain. The support is generally circular or rectangular and the weight function assumes the shape of a ball-function. Generally, exponential functions or splines are used [5]. In the present paper, reference will be made to conical, cubic and quartic splines weight functions, whose expressions in the one-dimensional case are, respectively,

$$w(s) = \begin{cases} 1 - s^{2k}, & s \leqslant 1 \\ 0, & s > 1 \end{cases} \tag{1}$$

$$w(s) = \begin{cases} \frac{2}{3} - 4s^2 + 4s^3, & s \leqslant \frac{1}{2} \\ \frac{4}{3} - 4s + 4s^2 - \frac{4}{3}s^3, & \frac{1}{2} < s \leqslant 1 \\ 0, & s > 1 \end{cases} \tag{2}$$

$$w(s) = \begin{cases} 1 - 6s^2 + 8s^3 - 3s^4, & s \leqslant 1 \\ 0, & s > 1 \end{cases} \tag{3}$$

s being the normalized distance and k being a positive real number. The moving least-square interpolant $\eta(x)$ of the function $\delta(x)$ in the domain Ω is defined by the following linear combination of basis functions $\{p(x)\}$:

$$\eta(x) = \sum_{i=1}^{m} p_i(x) a_i(x) = \underset{(1\times m)}{\{p(x)\}^{\mathrm{T}}} \underset{(m\times 1)}{\{a(x)\}} \tag{4}$$

where m is the number of the terms in the basis and $\{a(x)\}$ is the vector of the coefficients, which are functions of the spatial co-ordinates x. The choice of the basis depends on the boundary value problem to be solved. In the two-dimensional case, the linear and quadratic bases are given by

$$\{p(x)\}^{\mathrm{T}} = \{1, x, y\} \tag{5}$$

$$\{p(x)\}^{\mathrm{T}} = \{1, x, y, xy, x^2, y^2\} \tag{6}$$

The coefficients $\{a(x)\}$ in Equation (4) are obtained, at any point x, by minimizing the square of the difference between the local approximation $\eta(x)$ and the value δ_I of the function

$\delta(x)$ at $x = x_I$. Therefore, the following functional is minimized:

$$J(x) = \frac{1}{2} \sum_{I=1}^{n} w_I(x - x_I) \left[\sum_{i=1}^{m} p_i(x_I) a_i(x) - \delta_I \right]^2 \tag{7}$$

n being the number of the nodes whose supports contain the point x. It can be observed that, as the weight functions have a compact support, the sum over I is limited to the nodes for which the associated weight function has the property $x \in \Omega_I$.

The following final form is obtained [5]:

$$\eta(x) = \sum_{I=1}^{n} \phi_I(x) \delta_I \equiv \underset{(1\times n)}{\{\phi(x)\}^{\mathrm{T}}} \underset{(n\times 1)}{\{\delta\}} \tag{8}$$

$\{\phi(x)\}$ being the vector of the shape functions for the MLS approximation and δ the vector of nodal values. The approximation $\eta(x)$ is such that $\eta(x_I) \neq \delta_I$ and this results in the necessity to enforce explicitly Dirichlet (essential) boundary conditions.

In the following, the MLS shape functions will be used to solve the elastic problem. Hence Equation (8) will be applied to each displacement component. Nevertheless, to avoid unnecessarily involved notation, the displacement field will be denoted by $\{\eta\}$ as well.

3. VARIATIONAL FORMULATION

The shape functions derived in the former section are introduced into a variational formulation for the linear elastic solid. Equation (8) refers to the approximation of a scalar variable and the vector $\{\delta\}$ has a dimension equal to the MLS nodes number, N. For an elastic solid in an n_{d}-dimensional space, let $\{\eta\}$ be the displacement field, with n_d components and $N_d = n_d N$. The MLS approximation can be applied for each component of the displacement field in the form

$$\underset{(n_d\times 1)}{\{\eta\}} = \underset{(n_d\times N_d)}{[\Phi_\eta]} \underset{(N_d\times 1)}{\{\delta\}} \tag{9}$$

where $[\Phi_\eta]$ is the matrix of the shape functions and $\{\delta\}$ is the vector of all the nodal components of the approximation variables. To simplify the notation let us introduce the following shape function vector for the deformations:

$$\underset{(d\times 1)}{\{\varepsilon\}} = \underset{(d\times n_d)}{[\partial]} \underset{(n_d\times 1)}{\{\eta\}} = \underset{(d\times n_d)}{[\partial]} \underset{(n_d\times N_d)}{[\Phi_\eta]} \underset{(N_d\times 1)}{\{\delta\}} = \underset{(d\times N_d)}{[\Phi_\varepsilon]} \underset{(N_d\times 1)}{\{\delta\}} \tag{10}$$

with $d = 1$ for $n_d = 1$, $d = 3\times(n_d - 1)$ for $n_d = 2, 3$ and $\{\varepsilon\}$ is the deformation vector. The matrix $[\Phi_\varepsilon]$ is determined as the symmetric part of the gradient of the displacement field, Equation (9), through the differential operator $[\partial]$.

Let us consider the domain Ω of a solid subjected to volume forces $\{\mathscr{F}\}$, surface forces $\{q\}$ and prescribed displacements $\{\bar{\eta}\}$, acting, respectively, on Ω, $\partial\Omega_q$ and $\partial\Omega_\eta$ ($\partial\Omega = \partial\Omega_q \cup \partial\Omega_\eta$, $\partial\Omega_q \cap \partial\Omega_\eta = \emptyset$). The solution of the Cauchy elastic problem is given by the minimum of the

potential energy functional

$$\Pi(\{\eta\}) = \frac{1}{2}\int_{\Omega} \underset{(1\times d)}{\{\varepsilon\}^{\mathrm{T}}}\ \underset{(d\times d)}{[C]}\ \underset{(d\times 1)}{\{\varepsilon\}}\ \mathrm{d}\Omega - \int_{\Omega} \underset{(1\times n_d)}{\{\mathscr{F}\}^{\mathrm{T}}}\ \underset{(n_d\times 1)}{\{\eta\}}\ \mathrm{d}\Omega$$
$$- \int_{\partial\Omega_q} \underset{(1\times n_d)}{\{q\}^{\mathrm{T}}}\ \underset{(n_d\times 1)}{\{\eta\}}\ \mathrm{d}S + \int_{\partial\Omega_u} \mathrm{ind}\left(\underset{(n_d\times 1)}{\{\bar{\eta}\}} - \underset{(n_d\times 1)}{\{\eta\}}\right)\mathrm{d}S \tag{11}$$

$[C]$ being the elastic operator. The last part of Equation (11) is added as the MLS shape functions do not satisfy the essential boundary conditions, being $\{\eta(x_I)\} \neq \{\delta_I\}$, and the indicator function is equal to

$$\mathrm{ind}(\{\bar{\eta}\} - \{\eta\}) = \begin{cases} 0 & \text{if } \{\bar{\eta}\} - \{\eta\} = \{0\} \\ +\infty & \text{if } \{\bar{\eta}\} - \{\eta\} \neq \{0\} \end{cases} \tag{12}$$

The total potential energy functional, Equation (11), is of course not differentiable due to the presence of the indicator function for the kinematically admissible displacements. To transform it into a differentiable equivalent one, it will be replaced by its augmented Lagrangian regularization [6, 10].

Introducing the hypothesis that the assigned displacements $\{\bar{\eta}\}$ are interpolated by a set of delta Dirac functions placed to correspond with each node x_I on $\partial\Omega_\eta$ (which is equivalent to enforcing the essential boundary conditions only on a discrete number of points), the following expression is obtained:

$$\int_{\partial\Omega_u} \mathrm{ind}(\{\bar{\eta}\} - \{\eta\})\,\mathrm{d}S = \sum_{i=1}^{n_r} \sup_{\{r\}_i} \left[\{r\}_i^{\mathrm{T}}(\{\bar{\eta}_r\}_i - \{\eta_r\}_i) + \frac{1}{2}\alpha(\{\bar{\eta}_r\}_i - \{\eta_r\}_i)^{\mathrm{T}}(\{\bar{\eta}_r\}_i - \{\eta_r\}_i)\right] \tag{13}$$

where $\{r\}_i$ and $\{\bar{\eta}_r\}_i$ are, respectively, the reactions (which have the mathematical meaning of Lagrangian multipliers) and the assigned displacements on the n_r constrained points, while $\alpha > 0$ is the penalty parameter. To improve the precision in the discretization of the essential boundary conditions, other points, not necessarily discretization nodes, can be constrained along the boundary with the same technique.

By introducing the MLS approximation, Equations (9), (10), and the augmented Lagrangian regularization (13) into the potential energy functional (11), an augmented Lagrangian functional $\Pi(\{\delta\}, \{r\})$ is obtained, whose saddle point gives the problem solution

$$\min_{\{\delta\}} \max_{\{r\}} \Pi(\{\delta\}, \{r\}) \tag{14}$$

The present variational formulation allows the elimination of all the drawbacks related to the use of the Lagrangian and penalty methods, and represents a computationally effective approach for the introduction of constraints on direct variables in the MLS approximation. Full details are reported in Reference [6], while an extension to crack problems is given in Reference [9].

4. THE PARTITION OF UNITY QUADRATURE METHOD

In this section, a new quadrature method for the weak form of the equilibrium equations is introduced. This method will be called the partition of unity quadrature (PUQ), as it is conceptually based on the partition of unity property of the MLS shape functions. This property states that at any point of the domain the MLS shape functions are such that [5, 14, 15].

$$\sum_{I=1}^{n} \phi_I(x) = 1 \quad \forall x \in \Omega \tag{15}$$

To this end let f, an integrable function, be defined over the domain Ω. In the same domain, a set of N MLS nodes is considered, and the relevant shape functions ϕ_I, $I = 1 \ldots N$ are computed.

Let Ω_I the support for the weight function w_I and H, the function that indicates if a generic point x is located inside the domain Ω, be defined by

$$H(x) = \begin{cases} 1 & \text{if } x \in \Omega \\ 0 & \text{if } x \notin \Omega \end{cases} \tag{16}$$

Observing that $\phi_I(x) = 0 \ \forall x \notin \Omega_I$ and that the value of the integral does not change if the integrand function is multiplied by one

$$\int_\Omega f \, \mathrm{d}\Omega = \int_\Omega fH \sum_{I=1}^{n} \phi_I \, \mathrm{d}\Omega = \sum_{I=1}^{N} \int_{\Omega_I} fH\phi_I \, \mathrm{d}\Omega = \sum_{I=1}^{N} \int_{\Omega_I} fHg_I \ \mathrm{d}\Omega \tag{17}$$

where the functions ϕ_I have been renamed g_I for the purpose of making a clear distinction between MLS approximation functions and partition of unity weights.

Equation (17) is the basis of the PUQ method and it states that it is possible to evaluate any integral subdividing the total domain into the union of the weight function supports, and evaluate the integral as the sum of the integrals computed within each support, weighting the integrand function by the MLS shape function associated with the support. The function H cuts the supports if they extend outside the domain Ω, as it happens when the supports intersect the boundary $\partial\Omega$. The PUQ can be used in the evaluation of definite integrals without any connection with element-free methods by generating a cloud of points for the construction of the partition of unity. Its application arises naturally in element-free methods, where a set of points and associated supports is used for constructing the shape functions, and it can be conveniently used at the same time for applying the PUQ.

The fundamental feature of the PUQ is, therefore, that there is no need to introduce a background quadrature mesh since the quadrature cells are determined by the weight supports, and there is no need to modify the variational formulation as suggested by Atluri and Zhu in the MLPG [4]. Moreover, as observed by Dolbow and Belytschko [16], the coincidence between quadrature cells and weight supports reduces the error in the integral evaluation. Finally, it should be noted that there is no need for the MLS approximation shape functions ϕ_I and for the partition of unity weight functions g_I to coincide. They can be constructed w.r.t. different bases and weights. This observation is particularly noteworthy when enriched bases are used for the approximation (e.g. crack problems), as the presence of the weight function in the integral makes it more difficult to evaluate. Therefore, it seems a good idea to

Table I. Integration regions depending on the shape of the support and on the intersecting/non-intersecting boundary.

Option	Support	Boundary intersection	Subdivision applied on
A	Circular	Yes	Arch of circle and triangles
B	Circular	No	Circle
C	Square	Yes	Triangles, quadrilateral
D	Square	No	Square

assume the simplest basis (e.g. the linear basis) and weights for the partition of unity weight function.

A key point in the application of the method is the quadrature rule adopted in each support. When the support Ω_I, Equation (17), is circular and fully contained in Ω, specific quadrature formulas [17, 18] can be used. For square and general quadrilateral supports, classical Gaussian quadrature can be used as well. On the contrary, if a support Ω_I is not fully contained in Ω but intersects the external boundary, its shape changes accordingly, so that two alternatives are possible:

1. dividing the support into integration cells of standard shape (e.g. triangles and quadrilaterals); and
2. leaving the shape of the support unmodified and assuming the integrand function to be equal to zero when evaluated at points outside Ω. This alternative preserves the meshless nature of the approach near the boundary, although somewhat larger error in the quadrature is to be expected.

A second issue in the quadrature at each support Ω_I is whether to consider the support itself as a cell for integration, or to divide it into smaller subcells. If approach (2) is used, this does not impair the meshless nature of the quadrature as the support has a standard subdivision into subcells repeatable on each support. Conceptually, the division of the support into subcells is equivalent to generating a quadrature formula over a standard domain where the location of quadrature points and the relevant weights are determined on the basis of formulas with a lower number of Gauss points.

The subdivision of a support into smaller integration cells has been carried out in different ways, depending on the shape of the support (circular or square) and on the choice (1) or (2) being used for carrying out the quadrature. The various possibilities are reported in Table I, and depicted in Figures 1 and 2.

For each option, in Table I, the following subcell structure is generated:

(A) the subcells are polar triangles (one side curved), polar rectangles (two sides curved) triangles and rectangles. The subdivision depends on the external boundary intersection;
(B) the subcells are polar triangles and polar rectangles. The subdivision is independent of the external boundary intersection;
(C) the subcells are triangles and rectangles. The subdivision depends on the external boundary intersection;
(D) the subcells are a uniform arrangement of rectangles. The subdivision is independent of the external boundary intersection.

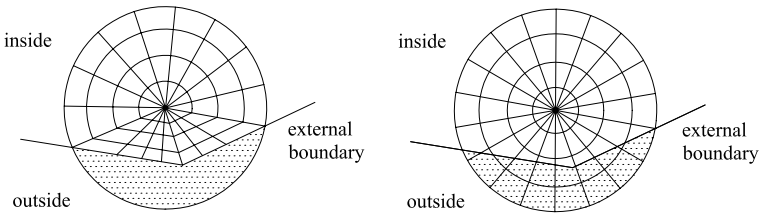

Figure 1. Circular weight support and subcell division for non-intersecting and intersecting external boundary.

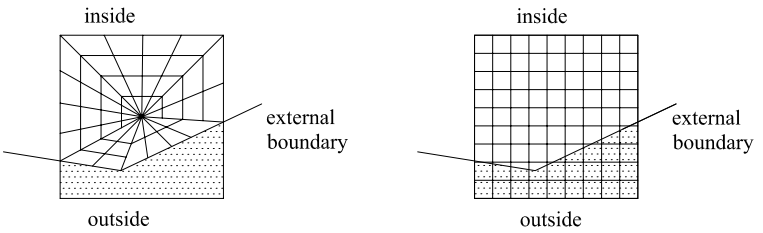

Figure 2. Square weight support and subcell division for non-intersecting and intersecting external boundary.

To apply the quadrature formulas, the triangles are mapped onto standard triangles, while quadrilaterals, polar triangles and polar rectangles are mapped onto the standard square, introducing the relevant transformation Jacobians. In the elementary regions T (standard triangle) and C (standard square) the following quadrature formulas have been implemented (see Reference [18] for details in naming notation):

- degree 1
  - T_n: 1-1, 1 point;
  - C_n: 1-1, 1 point;
- degree 2
  - T_n: 2-1, 3 points;
  - C_n: 2-1, 3 points;
- degree 3
  - T_n: 3-6, 6 points;
  - C_n: 3-4, 4 points;
- degree 5
  - T_2: 5-1, 7 points;
  - C_2: 5-4, 9 points;
- degree 7
  - T_2: 7-1, 16 points;
  - C_2: 7-4, 16 points;

where a formula is of degree n if it exactly integrates all polynomials up to degree n and there exists a polynomial of degree $n+1$ for which the formula is not exact. The interested reader can find details about these formulas, the location of points and the relevant weights in Reference [18].

Given a nodal point x_I and the support Ω_I of the weight function, it must be pointed out that the subdivisions sketched in Figures 1 and 2, are non-optimal, in the sense they tend to generate a quadrature point distribution which is denser near x_I and coarse near the boundary of Ω_I. Moreover, the following additional issues concerning the quadrature formulas must be considered to improve the efficiency of the method. Some of these points are currently under development:

- the use of quadrature formulas for circular regions;
- the study of different subcell divisions to improve the distribution of the quadrature points in the support;
- the study of quadrature formulas for polar triangles and polar rectangles, so that the Jacobian of the transformation into a standard square does not come into play and degrade the order of the formula; and
- at a given point x_I, the relevant weight function $w(x-x_I)$ appears as a 'constant presence' both in the evaluation of the shape functions and in the determination of the PUQ weight g_I. So special quadrature formulas for the entire support Ω_I, allowing for the exact quadrature of polynomials, could be developed.

5. NUMERICAL EXAMPLES

The PUQ is demonstrated in this section using some examples previously reported in the literature for assessing the convergence of the EFG method [7] and the MLPG formulation [3, 4]. For the purpose of error estimation and comparison with the literature results, the following displacement and energy norms are introduced:

$$||u|| = \left(\int_\Omega \{\eta\}^{\mathrm{T}} \{\eta\} \, \mathrm{d}\Omega \right)^{1/2} \tag{18}$$

$$||\varepsilon|| = \left(\frac{1}{2} \int_\Omega \{\varepsilon\}^{\mathrm{T}} [C] \{\varepsilon\} \, \mathrm{d}\Omega \right)^{1/2} \tag{19}$$

From these the relative errors in displacement and energy are computed by

$$r_u = \frac{||u^{\mathrm{NUM}} - u^{\mathrm{EXACT}}||}{||u^{\mathrm{EXACT}}||} \tag{20}$$

$$r_e = \frac{||\varepsilon^{\mathrm{NUM}} - \varepsilon^{\mathrm{EXACT}}||}{||\varepsilon^{\mathrm{EXACT}}||} \tag{21}$$

The first two examples examined are the regular and irregular patch test in a domain 2×2 as reported in Figure 3 [7]. Here the elastic constants $E=1$, $v=0.2$ are assumed and the

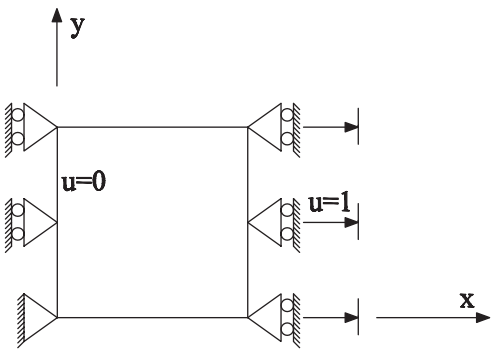

Figure 3. Geometry and boundary conditions for the patch test problem.

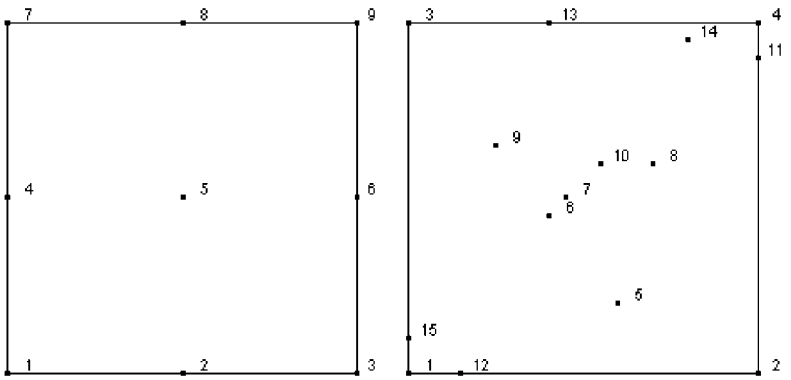

Figure 4. Nodal discretization for the regular and irregular patch test problem.

Table II. Nodal coordinates for the irregular patch test problem.

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x	0.0	2.0	2.0	0.0	1.2	0.8	0.9	1.4	0.5	1.1	2.0	0.3	0.8	1.6	0.0
y	0.0	0.0	2.0	2.0	0.4	0.9	1.0	1.2	1.3	1.2	1.8	0.0	2.0	1.9	0.2

displacements are set equal to zero for the nodes at $x=0$ and equal to one for the nodes at $x=2$. In Figure 4, only the discretization nodes are reported, while the boundary conditions are enforced at 11 equally spaced points per side.

The nodal coordinates for the irregular patch test are reported in Table II. For both patch tests the linear basis $\{p\}^{\mathrm{T}}=\{1,x,y\}$ has been used, along with the quartic spline weight function, Equation (3), with square supports and considering the external boundary intersection (option D). The assumed length for the size of the supports is 6. In Tables III and IV, the results for the standard Gauss quadrature on 4 cells (each 1×1 in size) are reported in comparison with the PUQ results. Of course the errors are very small for both Gauss and PU quadrature, since the MLS approximation is able to represent the solution exactly. It is apparent

Table III. Base 10 logarithm of the relative error norms for the regular patch test.

Quadrature rule	Subcells size	Quadrature points	log r_u	log r_e	log r_u Shephard's weight	log r_e Shephard's weight
Gauss 2×2		16	−24.69	−24.58		
Gauss 4×4		64	−24.38	−24.27		
PUQ degree 1	0.1	1672	−24.47	−24.35	−24.46	−24.35
PUQ degree 1	0.5	96	−25.81	−25.58	−25.90	−25.65
PUQ degree 1	1.0	24	−24.55	−24.32	−24.71	−24.46

Table IV. Base 10 logarithm of the relative error norms for the irregular patch test.

Quadrature rule	Subcells size	Quadrature points	log r_u	log r_e	log r_u Shephard's weight	log r_e Shephard's weight
Gauss 2×2		16	−18.53	−18.13		
Gauss 4×4		64	−18.80	−18.42		
PUQ degree 1	0.1	3001	−19.33	−18.94	−19.49	−19.10
PUQ degree 1	0.5	166	−19.88	−19.44	−18.84	−18.41
PUQ degree 1	1.0	49	−19.01	−18.46	−18.74	−18.17

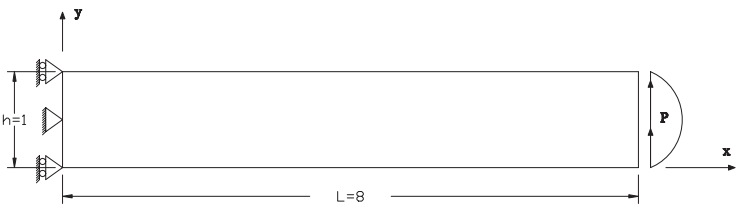

Figure 5. Clamped beam problem.

that in these two cases the PUQ method is able to reach the same precision of the standard Gauss quadrature with a nearly equivalent computational cost. There is no difference between using the same shape functions for both the approximation and the quadrature weights or by using Shephard's shape functions (i.e. the shape functions constructed by using the constant basis $\{p\} = \{1\}$) as quadrature weights.

The third example examined is the cantilever beam problem, Figure 5, for which the exact solution is given by Reference [13]

$$u_x = -\frac{P}{6\bar{E}J}\left(y - \frac{h}{2}\right)[(6L - 3x)x + (2 + \bar{\nu})(y^2 - hy)] \tag{22a}$$

$$u_y = \frac{P}{6\bar{E}J}\left[\frac{1}{4}h^2(4 + 5\bar{\nu})x + (3L - x)x^2 + 3\bar{\nu}(L - x)\left(y - \frac{h}{2}\right)^2\right] \tag{22b}$$

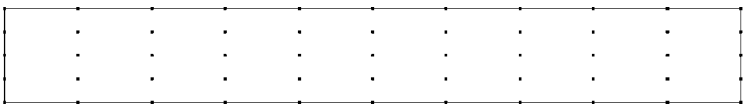

Figure 6. MLS nodal discretization for the clamped beam problem.

where $J = h^3/12$ and for the plane strain case analysed in References [4, 7] $\bar{v} = v/(1 - v)$, $\bar{E} = E/(1 - v^2)$. The stresses corresponding to (22) are

$$\sigma_x = \frac{P}{2J}(L - x)(h - 2y) \tag{23a}$$

$$\sigma_y = 0 \tag{23b}$$

$$\tau_{xy} = \frac{P}{2J}(h - y)y \tag{23c}$$

The data used in the example are $h = 1$, $L = 8$, $E = 3.0 \times 10^7$, $v = 0.25$. The load P is assumed parabolic and the exact displacement field is assigned at 21 equally spaced points at the clamped end. A regular mesh of 10×5 nodes, Figure 6, is used in the computation, with cubic spline weights, Equation (2), whose supports are squares with a side equal to 3.5. The use of circular supports, with the actually implemented quadrature formulas, has shown in general less numerical efficiency. For this reason, no results with this choice are presented here.

The quadratic basis $\{p\}^{\mathrm{T}} = \{1, x, y, xy, x^2, y^2\}$ is used in the construction of the MLS shape functions. In order to study the influence of the kind of partition of unity chosen to perform the PUQ (17), three cases have been considered, each corresponding to a different partition of unity weight function g used for the quadrature:

- g is coincident with the MLS shape functions used for the approximation (i.e. the PUQ weights are coincident with the MLS shape functions);
- g is taken as the Shephard's functions constructed by utilizing the same set of nodes and weights used for the approximation (i.e. the PUQ weights are the MLS shape functions computed assuming the constant basis $\{p(x)\} = \{1\}$); and
- g is taken as the Shephard's functions constructed by utilizing the same set of nodes used for the approximation and a linear ($k = 1/2$) conical weight function, Equation (1).

The best results have been obtained with rectangular supports, accounting for the intersection with the external boundary, and using a degree of quadrature 7 corresponding to the 4×4 Gauss integration rule on the rectangular subcells. In Table V, the ratio of the end deflection to the exact one (computed at the barycenter) is reported for the three partition of unity weights and for three different subcell sizes, respectively, equal to 2.0, 2.5, 3.0. For the smaller subcells, the result of the PUQ is coincident with the classical Gauss quadrature carried out on a regular 10×4 quadrature background mesh, Figure 7. The larger the subcells are in the PUQ, the more apparent the improvement is in using simpler partition of unity weights in the quadrature, namely the ones using Shephard's functions. This is a direct consequence of simplifying the integrand function in Equation (17).

Table V. Deflection ratio w/w_{exact} for the clamped beam problem considering the external boundary intersection.

Points	2×2	4×4	6×6
Gauss	0.998	0.998	0.998
PU type	$g=$MLS	$g=$Shephard	$g=$Shephard, $w=$Conical
PUQ 7-2.0	0.998	0.998	0.998
PUQ 7-2.5	1.009	0.998	0.998
PUQ 7-3.0	1.042	1.005	1.005

Table VI. Deflection ratio w/w_{exact} for the clamped beam problem. External boundary intersection not considered.

Points	2×2	4×4	6×6
Gauss	0.998	0.998	0.998
PU type	$g=$MLS	$g=$Shephard	$g=$Shephard, $w=$Conical
PUQ 7-1.0	1.106	1.052	1.054
PUQ 7-0.5	0.987	1.019	1.016
PUQ 3-1.0		1.106	1.117
PUQ 3-0.5	0.963	1.062	1.054

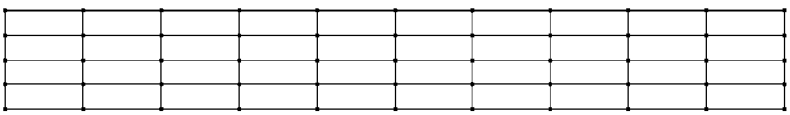

Figure 7. Background mesh used for Gaussian quadratures in the clamped beam example.

Similar results, but showing less accuracy, are obtained when the intersection of the supports with the external boundary is not considered, Table VI, even if the subcell size has been decreased to 1.0 and 0.5.

This result is observed again in comparing the computed tangential stresses τ_{xy} to the exact ones at the midspan for the various quadrature schemes, and considering the external boundary intersection. In Figure 8, the results for standard Gaussian quadrature on the background mesh are reported, showing progressively better results as the order of integration increases. The PUQ quadrature with a subcell size equal to 2.0 gives more accurate results, Figure 9, than the standard one. When using larger subcell sizes, Figures 10 and 11, the results show less precision, which can be improved by acting on the choice of the partition of unit weight g. The computation of the tangential stress when the boundary intersection is not considered shows again a sensible loss of precision, Figure 12.

Finally, the base 10 logarithms of the relative error norms r_u and r_e are reported in Figures 13 and 14. The integrals appearing in the norms have been computed using the 6×6 Gauss rule on the background quadrature mesh. In the legends of Figures 13 and 14, the different kinds of quadrature used are indicated, and the value enclosed in parentheses is the total number of points used for the quadrature. It is observed again that the use of Shephard's functions as PUQ weights is especially attractive when larger support subcells are considered. The precision of the results obtained as measured by the relative error norms r_u and r_e is very

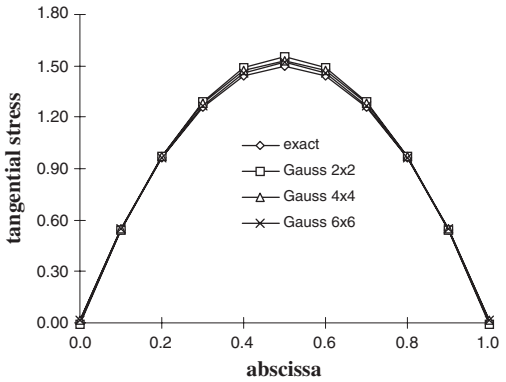

Figure 8. Tangential stress τ_{xy} computed using Gaussian quadrature.

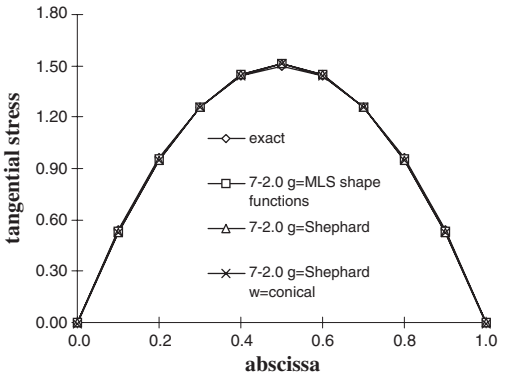

Figure 9. Tangential stress τ_{xy} computed by PUQ, degree 7, cell size 2.0.

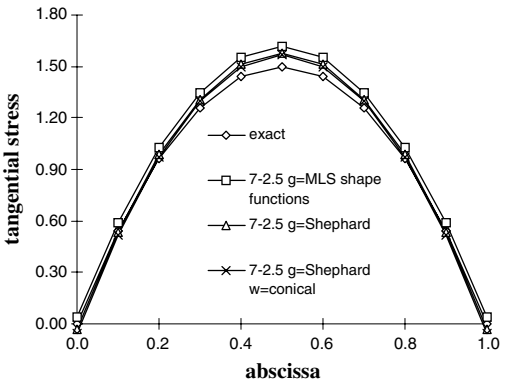

Figure 10. Tangential stress τ_{xy} computed by PUQ, degree 7, cell size 2.5.

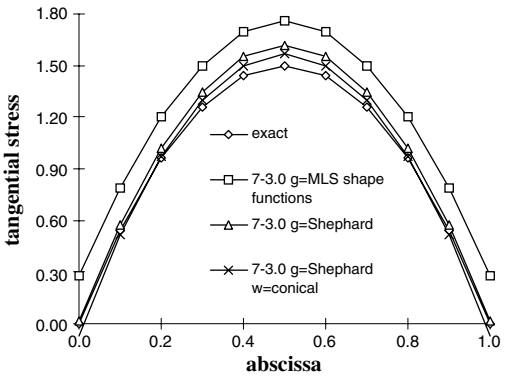

Figure 11. Tangential stress τ_{xy} computed by PUQ, degree 7, cell size 3.0.

high. In fact, when comparing these results with those obtained by Atluri and Zhu [3, 4], it is observed that the used mesh size h (defined as the nodal spacing in the x direction) is such that $\log_{10} h = -0.097$, and so it is larger than those considered by them. The extrapolation of the results obtained by Atluri and Zhu to this mesh size is a value of about -1.5 for both $\log_{10} r_u$ and $\log_{10} r_e$, a value that is obtained here only in the worst case for $\log_{10} r_e$. A similar comparison with the work by Belytschko *et al.* [7], shows comparable results for the relative error in energy, but less accurate results when comparing the relative error in displacements $\log_{10} r_u$.

The numerical efficiency of the presented approach, measured by the number of quadrature points necessary to reach a given accuracy in the results, is strongly affected by various factors. In the patch test it has been shown to be perfectly comparable to the traditional Gauss quadrature on a background mesh. Elongated domains, like the beam problem just presented, exhibit less numerical efficiency. In fact, the comparison between the standard

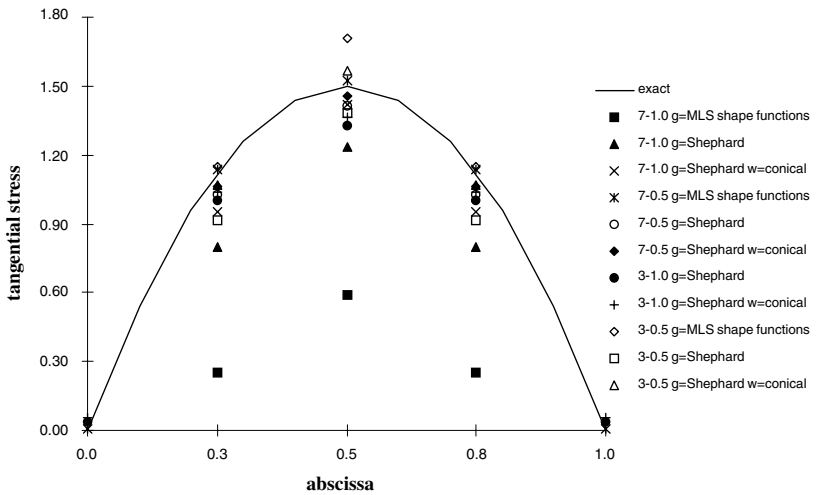

Figure 12. Tangential stress τ_{xy} computed by PUQ not accounting for the external boundary intersection.

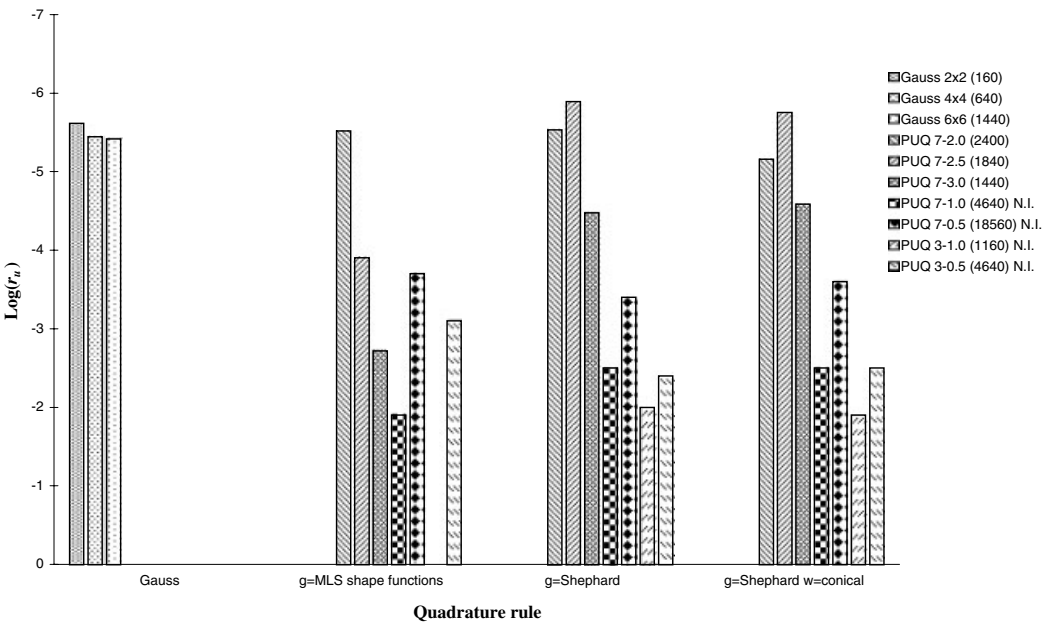

Figure 13. Base 10 logarithm of the relative error in displacement norm for the clamped beam example. The labels N.I. refer to the cases where the intersection with the external boundary is not considered.

Gauss quadrature and the PUQ, presented in Figures 13 and 14, made by observing the number of quadrature points for each method reported in parentheses, gives an advantage to the background mesh quadrature. However, the results for subcell of size 3.0 and Shephard's functions computed with linear conical weight are quite similar to standard quadrature results.

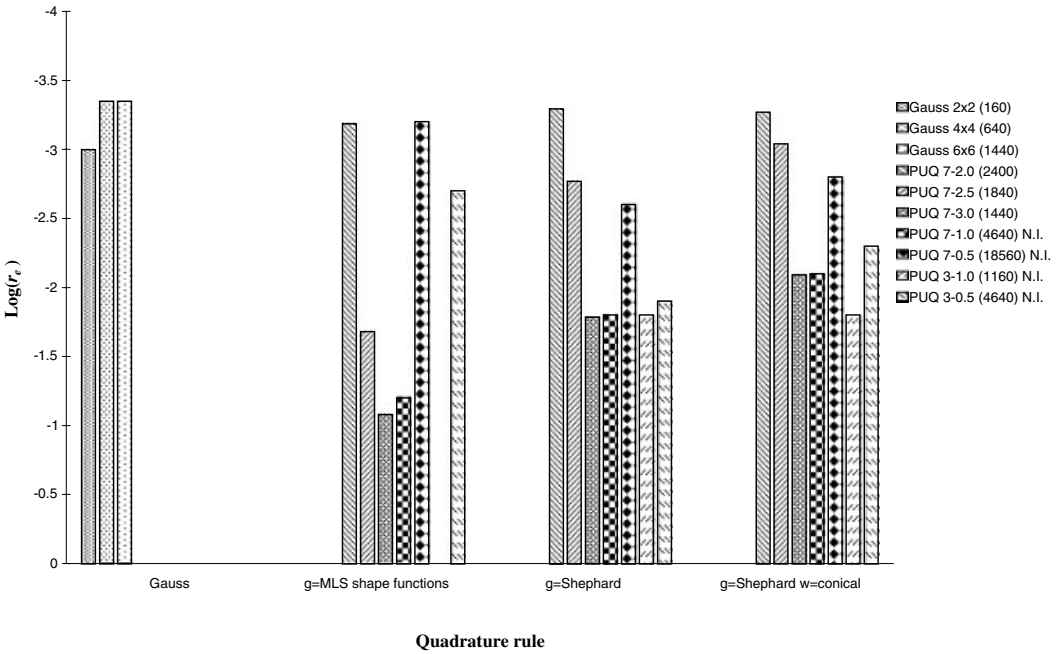

Figure 14. Base 10 logarithm of the relative error in energy norm for the clamped beam example. The labels N.I. refer to the cases where the intersection with the external boundary is not considered.

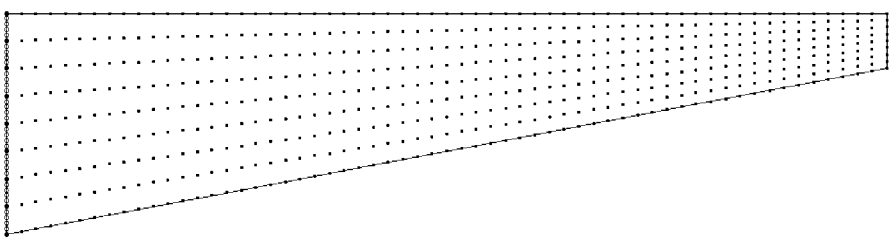

Figure 15. Nodal discretization for the variable height beam problem.

The PUQ has also been tested on a more complex example. A variable height beam has been discretized using 549 nodes with variable spacing, and 6×6 Gaussian quadrature has been defined over 481 cells. Figures 15 and 16 illustrate the nodal discretization and quadrature cells. The beam is clamped at the left end, and loaded by a uniform unit load acting on the right-end side. The beam length is 8.0, while the initial and ending heights are, respectively, 2.0 and 0.5. The elastic constants, bases and weights for constructing the MLS shape functions are assumed as in the previous example. The radius of the supports is variable and assumed as the minimum radius containing 8 times the number of basis functions, i.e. 48 nodes being the chosen basic quadratic. A contour plot of the horizontal stress σ_x is reported in Figure 17 for the Gaussian quadrature on subcells, while the same plot for PUQ of degree 7, subcells size

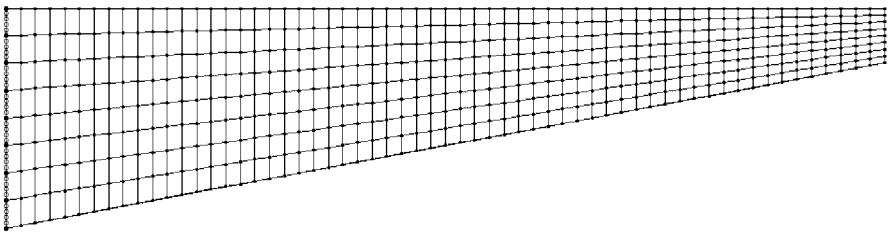

Figure 16. Gauss quadrature cells in the variable height beam problem.

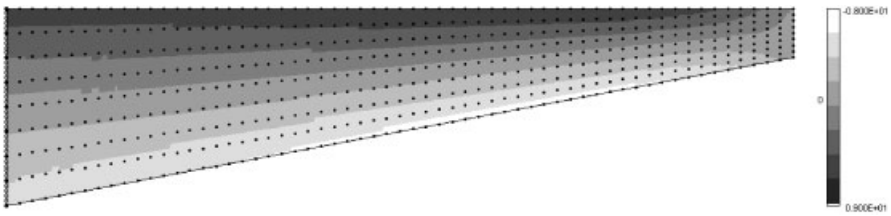

Figure 17. Horizontal stress σ_x computed by standard Gauss quadrature.

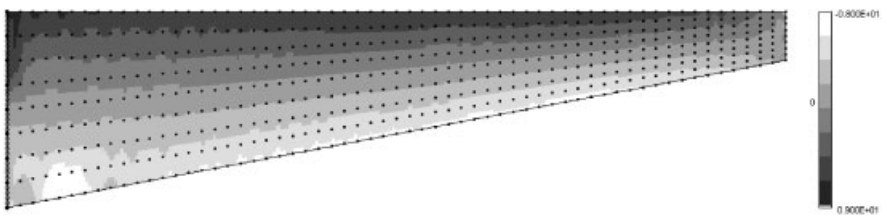

Figure 18. Horizontal stress σ_x computed by PUQ, degree 7, quadrature with subcell size 0.5.

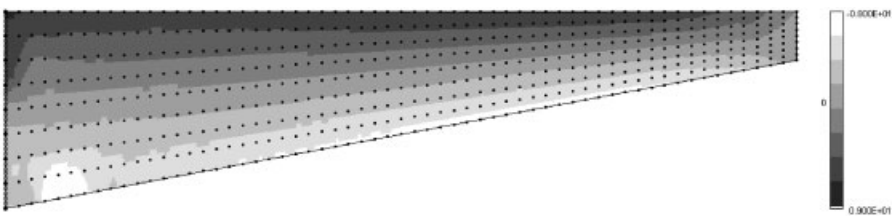

Figure 19. Horizontal stress σ_x computed by PUQ, degree 7, quadrature with subcell size 0.1.

0.5 and Shephard's weights is reported in Figure 18. In this figure, the number of PUQ points is 15 888, very close to the number of points used for the standard quadrature on subcells, equal to $36 \times 481 = 17\,316$, and the stress field is reproduced to a reasonable accuracy. The vertical displacement at the right end differs by 2.8%. Finally, in Figure 19 the subcell size is decreased to 0.1, generating 140 272 PUQ points. In this case, the stress field in the beam

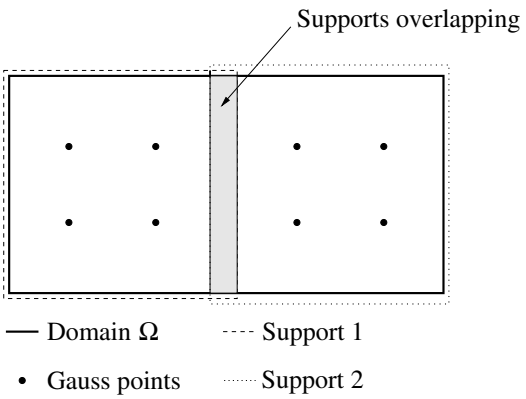

Figure 20. Computation of the area of a rectangular domain by PUQ.

appears quite similar to the one computed by Gauss quadrature even if some alterations of the stress field appear close to the clamp.

The reason why the PU method of quadrature requires a larger number of Gauss points compared to the standard one is not due to an intrinsic property of the method, but due to the problem that the equation upon which the PUQ is based, Equation (17), holds in the exact case, but does not strictly hold for numerical quadrature, where the integrand function is evaluated at a discrete number of (quadrature) points.

The problem is apparent in the computation of a domain area and considering that the integrations are performed by Gaussian quadrature over the supports Ω_I. Suppose that the area of the rectangular domain depicted qualitatively in Figure 20 is to be evaluated, and consider two rectangular intersecting supports. The domain Jacobians and PU weights are evaluated at the Gauss points, and the weights should be able to account for the intersection of the supports. This condition is, in general, not respected and produces an error in the determination of the domain area. The error decreases as the quadrature subcells size decreases and the order of integration is increased. The error vanishes if each support Ω_I covers the entire domain, as verified by numerical experiments in the patch test examples. Of course the latter solution is not feasible in practice.

In Table VII the area of the domain for the cantilever beam example has been computed using different subcell sizes and quadrature rules. It can be noted how, by refining the subcell divisions and increasing the order of the quadrature rule, the measure of the domain converges to the exact value.

Despite this consideration, the PUQ shows interesting results even in non-trivial discretizations, like the one used in the last numerical example. The results can be improved by studying and calibrating the several parameters involved in the choice of the PU weights computation. In fact, the following points are to be considered:

- the strategy used for carrying out the quadrature subdividing each weight function support very strongly affects the overall efficiency, and the algorithm used here has been found not to perform very well in this task;
- the use of a more specialized partition of unity weights and quadrature rules for the PUQ can result in significant improvements; and

Table VII. Computed area for the cantilever beam domain. The exact value is 8.0.

PU type	Gauss points	$g =$ MLS	$g =$ Shephard	$g =$ Shephard, $w =$ Conical
PUQ 7-2.0	2400	8.0009	7.9998	8.0051
PUQ 7-2.5	1840	7.9986	8.0013	7.9951
PUQ 7-3.0	1440	7.9944	8.0059	7.9837
PUQ 5-2.0	1350	8.0042	8.0018	7.9862
PUQ 5-2.5	1035	8.0188	7.9965	8.0033
PUQ 5-3.0	810	7.9644	7.9605	8.0019
PUQ 3-2.0	600	7.9712	7.9911	8.0026
PUQ 3-2.5	460	7.7909	8.0428	7.9577
PUQ 3-3.0	360	9.0651	8.3039	8.1119

- the numerical efficiency of the presented method need not be inferior to the MLPG approach proposed by Atluri and Zhu [4], as in both methods the quadrature is carried out on the weight function domains. But while MLPG requires a modified local variational principle, in the PUQ the variational principle is not affected at all, and a symmetric and banded system of equations is generated, in contrast to the unsymmetric one of the MLPG.

For these reasons, the PUQ method can be improved in several ways to combine its truly meshless nature and numerical efficiency.

6. CONCLUSIONS

A new method for the evaluation of the spatial integrals in meshless methods has been introduced. It is based on the partition of unity property of the moving least-squares shape functions. The method does not use any background quadrature cells and is as accurate as the classical Gauss quadrature over subcells. No modification of the variational principle is required for its application, and the generated system of equations is equal to the one generated by classical subcells integration. In the proposed examples, the method is generally more computationally demanding compared to the classical quadrature on subcells, but its numerical performance can be improved in several ways. Further studies on improving its numerical efficiency are currently being carried out.

REFERENCES

1. Beissel S, Belytschko T. Nodal integration of the element free Galerkin method. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:49–74.
2. Chen JS, Wu CT, Yoon S, You Y. A stabilized conforming nodal integration for Galerkin meshfree methods. *International Journal for Numerical Methods in Engineering* 2001; **50**:435–466.
3. Atluri SN, Zhu T. A new meshless local Petrov–Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics* 1998; **22**:117–127.
4. Atluri SN, Zhu T. New concepts in meshless methods. *International Journal for Numerical Methods in Engineering* 2000; **47**:537–556.
5. Belytschko T, Krongauz Y, Organ D, Fleming M, Krysl P. Meshless methods: an overview and recent developments. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:3–47.

6. Ventura G. An augmented Lagrangian approach to essential boundary conditions in meshless methods. *International Journal for Numerical Methods in Engineering* 2002; **53**:825–842.
7. Belytschko T, Lu YY, Gu L. Element free Galerkin methods. *International Journal for Numerical Methods in Engineering* 1994; **37**:229–256.
8. Carpinteri A, Ferro G, Ventura G. Material and crack discontinuities: application of an element free augmented Lagrangian method. In *Damage and Fracture Mechanics*, Brebbia CA, Carpinteri A (eds). Computational Mechanics Publications: Southampton, U.K., 1998; 237–246.
9. Carpinteri A, Ferro G, Ventura G. An augmented Lagrangian element free (ALEF) approach for crack discontinuities. *Computer Methods in Applied Mechanics and Engineering* 2001; **191**:941–957.
10. Cuomo M, Ventura G. Complementary energy formulation of no-tension masonry-like solids. *Computer Methods in Applied Mechanics and Engineering* 2000; **189**:313–339.
11. Fletcher R. *Practical Methods of Optimization*. Wiley: New York, 1987.
12. Bertsekas DP. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press: London, 1982.
13. Timoshenko SP, Goodier JN. *Theory of elasticity* (3rd edn). McGraw-Hill: New York, 1987.
14. Melenk JM, Babuska I. The partition of unity finite element methods: basic theory and application. *Seminar for Applied Mathematics*, *ETH Zurich*, *Research Report*, vol. **96**(01), 1996.
15. Melenk JM, Babuska I. The partition of unity method. *International Journal for Numerical Methods in Engineering* 1997; **40**:727–758.
16. Dolbow J, Belytschko T. Numerical integration of the Galerkin weak form in meshfree methods. *Computational Mechanics* 1999; **23**:219–230.
17. Engels H. *Numerical Quadrature and Cubature*. Academic Press: London, 1980.
18. Stroud AH. *Approximate Calculation of Multiple Integrals*. Prentice-Hall: Englewood Cliffs, NJ, 1971.

