Computation of Accurate Solutions when using Element-Free Galerkin Methods for Solving Structural Problems

Abstract

This paper investigates the application of adaptive integration in element-free Galerkin methods for solving problems in structural and solid mechanics in order to obtain accurate reference solutions.

An adaptive quadrature algorithm which allows user control over integration accuracy, previously developed for integrating boundary value problems, is adapted to elasticity problems. The algorithm allows the development of a convergence study procedure that takes into account both integration and discretisation errors. The convergence procedure is demonstrated using an elasticity problem which has an analytical solution and is then applied to accurately solve a soft tissue extension problem involving large deformations.

The developed convergence procedure, based on the presented adaptive integration scheme, allows the computation of accurate reference solutions for challenging problems which do not have an analytical or finite element solution.

Keywords: adaptive quadrature; element-free Galerkin methods; Total Lagrangian formulation; explicit integration; dynamic relaxation.

1. Introduction

Mesh-free methods are a class of numerical methods in which field variable approximation or interpolation is achieved without a predefined mesh, providing an attractive alternative to common mesh-based methods, such as the finite element method (FEM), in application to many complex problems in engineering analysis [1,2]. The element-free Galerkin (EFG) method was developed in [3] through corrections to the diffuse element method [4] which adopted moving least squares (MLS) approximations [5] for field variables and the Galerkin weak form to discretise the governing differential equations.

While integration of the discretised equations of the finite element method is a trivial matter, due to the polynomial shape functions defined over elements, the shape functions in the EFG method are not polynomials and are defined on local support domains that do not necessarily align with integration cells. As a result, numerical integration in EFG methods remains a problem. A possible solution to

this problem consists of an adaptive integration procedure, such as the one proposed in [6] for elliptic boundary value problems.

1.1 Numerical integration in mesh-free methods

Gaussian quadrature over either a background cell structure or a background mesh defined using the discretisation nodes remains the most common numerical integration technique in mesh-free methods [2,3,7]. However, a large number of properly distributed integration points is required to ensure integration accuracy. In [7] the authors suggested that integration accuracy could be improved by aligning the integration cells with the supports of tensor product weight functions, and proposed a “bounding-box” technique to achieve this. They also demonstrated the improvement in integration accuracy that may be achieved by subdivision of integration cells, a technique used in the development of an adaptive quadrature algorithm in [6]

In efforts to transform the weak form EFG methods into *truly meshless* methods, nodal integration schemes have been explored [8]. In [9] a residual is added to the equilibrium equations of the weak form to approximate integrals in the discretised equations by a quadrature sampling the integrand only at the nodes. This removes the necessity to construct a mesh or cell structure for integration, but the method sacrifices accuracy of the weak formulation for ease of integration. In [10] an alternate approach to achieve nodal integration is attempted by using a strain-smoothing stabilisation to avoid evaluation of shape function derivatives at the nodes, thereby eliminating spurious modes. A comparison of this stabilised conforming nodal integration technique and Gaussian quadrature for EFG is presented in [11].

Alternative methods to improve the efficiency and accuracy of numerical integration have led to the modification of the variational principle into a local weak formulation in the meshless local Petrov-Galerkin (MLPG) and the local boundary integral equation (LBIE) methods [12]. Integration of the weak form is performed locally over sub-domains defined by the supports of the test and trial functions, using specially defined Gaussian quadrature rules. This method suffers from difficulties handling intersections of the integration domains with the boundaries. The method of finite spheres [13,14] is a special case of the MLPG method in which integration domains are defined by n -dimensional spheres centred at the nodes. Specialised cubature rules are developed for interior and boundary spheres, as well as for intersections of these spheres, called “lenses”. More efficient numerical integration schemes are explored in [15], and an adaptive integration technique is presented in [16]. While the method of finite spheres shows promise in achieving the meshless condition, handling of irregular boundaries remains difficult.

The partition of unity quadrature [17] seeks to take advantage of the partition of unity property of MLS shape functions. Global integrals are computed as the sum of integrals over nodal weight function supports, weighting integrands by either the MLS shape function or Shepard's function (MLS shape function with constant basis) associated with the support. Quadrature formulas are developed for weight function supports (generally circular or square regions), and subdivision of the support into smaller integration cells is also considered. A similar moving least squares quadrature is presented in [18] using tensor product weight functions that permit integration on rectangular supports and simpler treatment of irregular boundaries by setting quadrature weights equal to zero for the integration points outside the problem domain.

Other numerical integration schemes applied to mesh-free methods include quasi-Monte Carlo quadrature [19], the use of genetic algorithms to generate integration points and weights for a given nodal configuration [20] and other novel meshless integration techniques such as the Cartesian transformation method, in which domain integrals are transformed into boundary integrals [21].

1.2 Errors in mesh-free methods

Error estimation in computational mechanics is a growing field of research that provides a quantitative basis for adapting discrete computational models to improve the quality of numerical simulations [22]. For sensitive applications, such as brain deformation simulation in computational biomechanics, guaranteed error bounds within the limits of neurosurgical accuracy are required [23].

Following the discussion in [2], we consider the two main sources of errors in mesh-free methods: *discretisation error*, related to the inability of an approximating function to exactly represent the real function sampled discretely at the nodes; and *integration error*, a result of the error in the quadrature rule used to numerically integrate approximating functions.

In finite element methods, discretisation error is controlled by the element size h and the polynomial degree of interpolating functions p , while integration error for Gaussian quadrature rules is controlled by the number of quadrature points n . The accuracy of the solution may generally be improved in three ways: increasing p , decreasing h and matching n to the shape function's polynomial degree, e.g. for 1D Gaussian quadrature $n = (p + 1)/2$. In mesh-free methods, the same may be achieved by increasing the order of consistency of the mesh-free shape functions (e.g. using high-order basis functions in MLS approximations) and increasing the density of nodes, with existing adaptive procedures addressing these methods [24–28]. However, as there is generally no prior information about the polynomial degree of MLS approximations [26] and shape function support domains are generally not aligned with the integration cells, it is difficult to explicitly address the integration error.

The standard empirical approaches for minimising integration error in mesh-free methods in the literature uses a large number of background integration cells with a sufficiently high number of Gauss points in each cell. This method assumes that for sufficiently reduced integration cell size, the strain field may be approximated by a polynomial that can be accurately integrated using a chosen Gaussian quadrature rule. However, the number of integration points required to obtain prescribed integration accuracy is presently unknown [6]. Increasing the degree of Gaussian quadrature does not guarantee an increase in integration accuracy, as the integrands in mesh-free methods are non-polynomial [7]. A better method for improving integration accuracy is through subdivision of integration cells, with the same degree quadrature applied to the resulting subdivided cells. This fact was used in the development of an adaptive quadrature algorithm for EFG methods in [6].

2. Methods

2.1 Adaptive quadrature

Numerical integration in the EFG method is achieved on either a background grid of integration cells disjoint from the nodes or on a background mesh of integration cells defined by the discretisation nodes. On each integration cell, a quadrature rule defines integration points and corresponding weights. Gaussian quadrature rules are most commonly employed for EFG methods using background integration cells.

The n point Gaussian quadrature rule on an integration cell $D \subset \mathbb{R}^d$ approximates the integral

$$I = \int_D f(\mathbf{x}) \, dD \quad (1)$$

by the weighted sum

$$Q_n(D) = \sum_{i=1}^n w_i f(\mathbf{x}_i) \quad (2)$$

where $\mathbf{x}_i \in D$ are the *integration points* and w_i the corresponding *weights*.

An important consideration in the selection of an appropriate quadrature rule is the *algebraic degree of precision* of the quadrature, referred to herein as the degree of the quadrature. We consider a quadrature to be of degree N if and only if the quadrature rule is exact for all polynomials of degree $p \leq N$ but not for some polynomial of degree $N + 1$.

The essential nature of the adaptivity presented here consists of the subdivision of integration cells. A quadrilateral or triangular integration cell $D \subset \mathbb{R}^2$ may be subdivided into 4 smaller cells using midpoints of each edge of the cell. This subdivision process may be easily extended to higher dimensions. We use the notation $Q_n^m(D)$ to indicate an n point quadrature rule applied on m

subdivided regions of the integration cell D . Here $Q_n^4(D)$ would indicate the sum of n point quadrature rules applied to the 4 smaller cells produced by subdividing D .

We will apply the adaptive quadrature algorithm developed in [6], which uses recursive subdivision of background integration cells to reduce the error of integration below a user-specified *relative integration error tolerance*, τ . This adaptive integration procedure uses a local relative error estimate of the integration error for each integration cell to guide the adaptive division of integration cells.

The generalised form of the adaptive quadrature algorithm applied to integrate function f over integration cell D with relative integration error tolerance τ is provided in Algorithm 1. A more detailed discussion of the characteristics of the adaptive algorithm is presented in [6].

Algorithm 1

```

procedure  $[Q, x_i, w_i] = \text{integrate}(f, D, \tau)$ 
     $[Q, x_i, w_i] = Q_n^1(D)$ ; //  $x_i$  = integration points,  $w_i$  = associated weights
     $[QM, x_M, w_M] = Q_n^m(D)$ ;
    if  $(|QM - Q| > \tau|QM|)$  then
         $D \rightarrow \{D_1, D_2, \dots, D_m\}$ ; // subdivision
        for  $D_j$  in  $\{D_1, D_2, \dots, D_m\}$  do
             $[Q_j, x_{ij}, w_{ij}] = \text{integrate}(f, D_j, \tau)$ ; // recursion
        end for
         $Q = \text{sum}(Q_1, Q_2, \dots, Q_m)$ ;
         $x_i = \text{concatenate}(x_{i1}, x_{i2}, \dots, x_{im})$ ;
         $w_i = \text{concatenate}(w_{i1}, w_{i2}, \dots, w_{im})$ ;
    end if
    return  $Q, x_i, w_i$ 
end procedure

```

2.2 Generation of integration points for elasticity problem

We consider the equilibrium equation for the two-dimensional elasticity problem on domain Ω bounded by Γ :

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = \mathbf{0} \quad (3)$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor and $\mathbf{b}$ the body force. The boundary conditions are:

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \bar{\mathbf{t}} \text{ on } \Gamma_t \quad (4)$$

$$\mathbf{u} = \bar{\mathbf{u}} \text{ on } \Gamma_u \quad (5)$$

where $\mathbf{n}$ is the unit normal to the boundary Γ_t , $\bar{\mathbf{t}}$ are prescribed tractions on the boundary and $\bar{\mathbf{u}}$ are the prescribed displacements. Applying the Galerkin method to the weak form of (4) we seek trial functions $\mathbf{u} \in H^1(\Omega)$ such that for all test functions $\delta \mathbf{u} \in H_0^1(\Omega)$, equation (6) is satisfied:

$$\int_{\Omega} (\nabla_s \delta \mathbf{u}^T) \mathbf{C} (\nabla_s \mathbf{u}) d\Omega = \int_{\Gamma} \delta \mathbf{u}^T \bar{\mathbf{t}} d\Gamma + \int_{\Omega} \delta \mathbf{u}^T \mathbf{b} d\Omega \quad (6)$$

where ∇_s is the symmetric gradient operator and $\mathbf{C}$ is the tensor representing the stress–strain constitutive law. For the simplicity of presentation, and without loss of generality, we will derive the discretisation for 2D problems, where

$$\nabla_s = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}. \quad (7)$$

The element-free Galerkin method uses trial functions (8) and test functions (9) based on moving least squares (MLS) shape functions to generate the discrete equations [29]:

$$\mathbf{u}(\mathbf{x}) = \sum_I^N \phi_I(\mathbf{x}) \mathbf{u}_I \quad (8)$$

$$\delta \mathbf{u}(\mathbf{x}) = \sum_I^N \phi_I(\mathbf{x}) \delta \mathbf{u}_I \quad (9)$$

The value of these functions is obtained at any point $\mathbf{x} \in \Omega$ by minimising a weighted least squares error functional where $\phi_I(\mathbf{x})$ is the MLS shape function corresponding to node I [3,5]. In the present study, as in [6], a quartic spline weight function of the normalised Euclidean distance is used, with a variable radius of influence domain assigned to each node based on a dilated mean Euclidean distance to neighbouring nodes.

By substituting into equation (6), neglecting body forces and invoking the arbitrariness of $\delta \mathbf{u}_I$, the discretised system of equations is obtained in matrix form as:

$$\mathbf{K} \mathbf{u} = \mathbf{f} \quad (10)$$

$$\mathbf{K}_{IJ} = \int_{\Omega} \mathbf{B}_I^T \mathbf{C} \mathbf{B}_J d\Omega \quad (11)$$

$$\mathbf{f}_I = \int_{\Gamma_t} \phi_I(\mathbf{x}) \bar{\mathbf{t}} d\Gamma \quad (12)$$

$$\mathbf{B}_I = \begin{bmatrix} \phi_{I,x}(\mathbf{x}) & 0 \\ 0 & \phi_{I,y}(\mathbf{x}) \\ \phi_{I,y}(\mathbf{x}) & \phi_{I,x}(\mathbf{x}) \end{bmatrix} \quad (13)$$

$$\mathbf{C} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \text{ for plane stress.} \quad (14)$$

The MLS shape functions do not possess the Kronecker delta property, and hence the essential boundary conditions (5) are enforced by adding unknown tractions $\mathbf{t}_u$ on the boundary Γ_u , and appending extra equations describing the essential boundary conditions to the existing system of equations. The resulting system of equations is given by (15) where K indexes the fixed boundary nodes, $\psi_K(s)$ is an interpolating shape function and s is the arclength along the fixed boundary.

$$\begin{bmatrix} \mathbf{K} & \mathbf{H} \\ \mathbf{G} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \mathbf{t}_u \end{Bmatrix} = \begin{Bmatrix} \mathbf{f} \\ \mathbf{0} \end{Bmatrix} \quad (15)$$

where $\mathbf{K}$ and $\mathbf{f}$ are as defined previously and

$$\mathbf{H}_{IK} = \int_{\Gamma_u} \phi_I(\mathbf{x}) \mathbf{N}_K d\Gamma \quad (16)$$

$$\mathbf{G}_{KJ} = \begin{bmatrix} \phi_J(\mathbf{x}) & 0 \\ 0 & \phi_J(\mathbf{x}) \end{bmatrix} \quad (17)$$

$$\mathbf{N}_K = \begin{bmatrix} \psi_K(s) & 0 \\ 0 & \psi_K(s) \end{bmatrix} \quad (18)$$

The degrees of freedom related to the essential boundary conditions are then eliminated from (15) through static condensation.

Examining equations (10) to (14) we notice that the integrals that must be computed using the numerical quadrature are based on products of derivatives of the shape functions:

$$I_{IJ} = \int_{\Omega} \phi_{I,i}(\mathbf{x}) \phi_{J,j}(\mathbf{x}) d\Omega \quad (19)$$

where i indicates partial spatial derivatives.

Integrands of the form above are defined on finite intersections of the local compact support domains of the shape functions. In general, over a given integration cell, multiple stiffness matrix integrands will be non-zero. Consequently, multiple different functions may need to be integrated over each integration cell. However, for a given integration cell, the number and nodal indexes of non-zero integrands at the integration points may differ. Thus, while the adaptive quadrature could be applied to each integral in the discretised equations, this may require expensive computation of shape function derivatives at a given integration point many times during the assembly of the global stiffness matrix. Should the integration cell then need to be subdivided by the adaptive algorithm, the number of function evaluations would increase further, thereby increasing the computational expense.

The solution, proposed in [6], is to construct a global function representative to the integrands in the local stiffness matrices and apply the adaptive quadrature over the entire domain using this function. Such a function must have a shape similar to the integrands and, additionally, must have a non-zero integral value over any integration cell (due to the relative integration accuracy test used for guiding cell subdivision in Algorithm 1). Following the heuristic process described in [6], we define the function used to guide the adaptive quadrature algorithm as:

$$f(\mathbf{x}) = \left[\sum_I^N (\phi_{I,x}(\mathbf{x}))^2 \right] \left[\sum_I^N (\phi_{I,y}(\mathbf{x}))^2 \right] \quad (20)$$

We note that integrals computed over the boundary Γ are not considered in the construction of function f . For simplicity, the adaptive quadrature is applied only for integrals over the domain Ω in the present study. All boundary integrals are accurately integrated using a high degree (10 points) Gaussian quadrature over the boundary segments between nodes, although a similar adaptive scheme can be employed on the boundary.

When computing the values of function f we take advantage of the fact that the shape functions and their derivatives have local compact support domains; therefore, for a given integration point only a limited number of shape function derivatives will have non-zero value, and they are all computed simultaneously using the MLS minimisation procedure.

The adaptive division of integration cells resulting from the integration of function f leads to a distribution of integration points with corresponding weights that is defined entirely by the shape functions and the integration error accuracy τ . This distribution of integration points will integrate the stiffness matrix components to the same integration error accuracy used for their generation, as it will be demonstrated in Section 3.

2.3 Convergence study

While the above procedure ensures integration accuracy under a selected level τ , it gives no indication of what accuracy level should be used for a specific problem and nodal discretisation. We propose the use of a convergence study for integration errors in order to determine the optimum integration accuracy. We designed a convergence study procedure that independently addresses the contribution of integration and discretisation errors to the overall solution error. By controlling integration error using the adaptive quadrature, and minimizing discretisation error through increasing node density, a

converged solution can be obtained for any given problem. The proposed convergence study procedure is as follows:

1. Create a low density nodal discretisation and a background mesh or grid of integration cells with the cell size approximately equal to the distance between the discretisation nodes.
2. Use the adaptive quadrature algorithm to generate integration points and weights for integration with incrementally reduced integration parameter τ (through order of magnitude decrements) until the solution converges.
3. Using the adaptive quadrature algorithm with the integration parameter τ at the value at which convergence of the integration error is observed, incrementally increase the nodal density (e.g. by adding mid-point nodes to a triangulation of the discretisation nodes) until the solution converges.

We demonstrate the application of this convergence study procedure to elasticity problems in the following section.

3. Numerical Examples

3.1 2-D elasticity example—cantilever beam with analytical solution

3.1.1 Problem description

We consider a cantilever beam of dimension $l \times 2c$ with unit depth, subject to a parabolically distributed traction force P on the free end as depicted in Fig. 1. This linear elasticity problem has an analytical solution; therefore, it allows high accuracy error computation for the integration convergence studies.

Fig. 1. Cantilever beam.

A similar problem, solved using element-free Galerkin methods, is presented in [2]. When the traction on the free end ($x = 0$) is given by:

$$t = \frac{P}{2I} (c^2 - y^2) \quad (21)$$

The displacement of this beam has the following analytical solution, as described in [30]:

$$u_x(x, y) = \frac{-Px^2y}{2EI} - \frac{\nu Py^3}{6EI} + \frac{Py^3}{6IG} + \left(\frac{Pl^2}{2EI} - \frac{Pc^2}{2IG} \right) y \quad (22)$$

$$u_y(x, y) = \frac{\nu Pxy^2}{2EI} + \frac{Px^3}{6EI} - \frac{Pl^2x}{2EI} + \frac{Pl^3}{3EI} \quad (23)$$

where $I = \frac{2c^3}{3}$ is the moment of inertia (second moment of area), E is the Young's modulus, ν the Poisson's ratio and $G = \frac{E}{2(1+\nu)}$ the shear modulus.

We used the following values for our numerical experiments: $P = 1000$, $E = 30000$ and $\nu = 0.3$, with the problem domain defined by $l = 48$ and $c = 6$.

With prescribed relative integration error tolerance $\tau = 0.001$, the node and integration cell distributions for 50 nodes and adaptively defined integration points are provided below. A 2×2 point tensor product Gaussian quadrature is used for quadrilateral integration cells (Fig. 2) and a 4-point symmetric Gaussian quadrature is used for triangular integration cells (Fig. 3). Local regions of varying nodal density that result in difficult to integrate shape functions undergo further adaptive division of integration cells resulting in automatic generation of more integration points in these areas.

Fig. 2. (a) Problem domain discretised by an irregular nodal distribution with a background grid of quadrilateral integration cells. (b) Adaptively defined integration points using tolerance $\tau = 0.001$ and a 2×2 point tensor product Gaussian rule on each cell.

Fig. 3. (a) Problem domain discretised by an irregular nodal distribution with a background mesh of triangular integration cells. (b) Adaptively defined integration points using tolerance $\tau = 0.001$ and a 4-point quadrature rule on each cell.

3.1.2 Error evaluation

Integration error is estimated by accurately computing the elements of the stiffness matrix $\mathbf{K}$ and comparing with those computed using the adaptive quadrature [6]. Since the diagonal terms of $\mathbf{K}$ are always non-zero, we define the relative integration error for each element of $\mathbf{K}$ as:

$$RE_{ij}^{int} = \left| \frac{\mathbf{K}_{ij}^* - \mathbf{K}_{ij}}{\mathbf{K}_{ii}} \right| \quad (24)$$

where $\mathbf{K}_{ij}^*$ is computed using the adaptive quadrature and $\mathbf{K}_{ij}$ is computed using a high degree (10 x 10 points) Gaussian quadrature over a large number (100 x 100) of integration cells. By varying the imposed integration accuracy τ and observing the effect on the average and maximum values of the above error measure, we may evaluate the control of integration error using the adaptive quadrature.

In order to assess the error of the numerical solution, we define the relative strain energy error norm as:

$$RE^{energy} = \frac{\left[\int_{\Omega} \frac{1}{2} (\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^h)^T \mathbf{C} (\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^h) d\Omega \right]^{\frac{1}{2}}}{\left[\int_{\Omega} \frac{1}{2} \boldsymbol{\epsilon}^T \mathbf{C} \boldsymbol{\epsilon} d\Omega \right]^{\frac{1}{2}}} \quad (25)$$

where $\boldsymbol{\epsilon}$ refers to strains determined using analytical expressions, while $\boldsymbol{\epsilon}^h$ are strains computed numerically using the element-free Galerkin method with the adaptive quadrature. The accurate evaluation of the integrals involved in computing this error norm is performed by using a regular background grid with a very large number (100 x 100) of integration cells and a 10×10 tensor product Gaussian quadrature on each cell.

3.1.3 Performance assessment

We used the adaptive integration algorithm with both degree 3 and degree 5 quadrature rules. For quadrilateral cells, $n \times n$ point tensor product quadrature rules “G n ” are used, while m -point symmetric quadrature rules “T m ” are used for triangular cells.

Fig. 4. Variation of maximum (a) and average (b) integration errors with the relative integration error tolerance τ . The G2 and G3 quadrature rules are used on a background grid of quadrilateral cells and the T4 and T7 quadrature rules on a background mesh of triangular cells.

Fig. 5. Number of integration points against relative integration error tolerance τ .

Fig. 4 demonstrates the good correlation between requested relative integration error tolerance and achieved integration accuracy. This strong correlation suggests the adaptive quadrature algorithm is highly reliable and responsive to decreases in relative integration error tolerance. Fig. 5 shows the increase in number of integration points generated by the adaptive quadrature as the requested tolerance is reduced. Increases in number of integration points results in a greater number of function evaluations during assembly of the stiffness matrix and therefore greater computational expense for the solution method.

3.1.4 Convergence study

The application of the convergence study procedure to the cantilever beam example is demonstrated in Fig. 6.

Fig. 6. (a) Integration error convergence: relative strain energy error variation against the relative integration error tolerance τ . (b) Node density convergence: relative strain energy error variation against nodal spacing h , computed as the maximum distance between neighbouring nodes in the discretisation. The relative integration error tolerance is $\tau = 10^{-3}$ for each experiment, as determined from (a).

From the results in Fig. 6.a we draw the following conclusions:

- For sufficiently reduced integration error (obtained by imposing a sufficiently small relative integration error tolerance τ), the solution errors converge towards the discretisation error, influenced only by the number and distribution of nodes and independent of the integration cell structure or quadrature rule.
- It is not necessary to obtain a higher accuracy of integration to ensure a converged solution. Adaptively reducing integration error beyond convergence of the solution errors will increase computational expense (greater number of integration points) without yielding a more accurate solution.

Fig. 6.b shows the convergence of solution errors with increasing nodal density. Our experiments have shown that there is no discernible improvement in solution convergence with a more accurate integration. In each case, as node density is increased, the starting integration cells are decreased in size to match the new nodal spacing. This is required for the adaptive quadrature algorithm to be effective, with starting integration cell size desired to be of the order of magnitude of distance between the nodes in the discretisation [6].

3.2 Application example—soft tissue sample extension

We use the adaptive quadrature algorithm with the meshless Total Lagrangian explicit dynamics (MTLED) solution method in an example of soft tissue extension simulation involving large deformations. The MTLED algorithm was developed for computing deformation of soft tissue for surgical simulations [31]. This algorithm combines the Total Lagrangian explicit dynamics (TLED) algorithm [32] with a meshless displacement field approximation based on EFG. Example applications of the algorithm for simulating mechanical responses of soft tissue are provided in [33,34]. This example demonstrates that the integration procedure can be used for the solution of nonlinear elasticity problems. It also demonstrates the application of the convergence study, with separation of integration and discretisation errors, for a problem which does not have an analytical solution.

The method for solving the continuum mechanics incremental equations of motion used in the MTLED algorithm uses a Total Lagrangian formulation, in which the displacements in the current configuration of a body are determined with reference to the original, undeformed configuration. Using this method in an explicit dynamics framework requires the computation of shape function derivatives only once, in the undeformed configuration, and these shape functions remain unchanged during the time integration. This is a very important feature of the Total Lagrangian formulation, which allows us to apply the adaptive quadrature algorithm only once, at the beginning of the simulation, generating a distribution of integration points and weights that will remain unchanged in all following time-steps. Explicit integration makes the algorithm very well suited for parallel integration [35,36].

3.2.1 Problem description

We simulate the behaviour of a 2-D sample of soft tissue under extension, similar to the problem presented in [37]. The problem domain is initially discretised using 57 nodes and a background grid of quadrilateral cells is created for integration (Fig. 7). Linear basis functions and a variable influence domain radius with dilatation parameter $M = 2$ are used to generate MLS shape functions. The nodes on the right side of the geometry ($x = 10$ cm) are displaced 3 cm in the positive x -direction. A simple Neo-Hookean hyperelastic constitutive model is used, with a Young's modulus of 3000 Pa in the undeformed state and a Poisson's ratio of 0.49.

The steady state deformation of the sample is obtained using dynamic relaxation [38–40]. A very low value for the imposed displacement accuracy has been used ($\epsilon = 10^{-10}$) [39], to ensure convergence to

the steady state solution. Essential boundary conditions were imposed using regularised weight functions, as in [37]. This leads to almost interpolating shape functions, while reducing their smoothness; this poses an additional challenge for the adaptive integration algorithm.

Fig. 7. Soft tissue extension. (a) Problem domain, initial nodal discretisation and background grid of quadrilateral integration cells. (b) Nodal position in the deformed configuration.

3.2.2 Convergence study

Without an analytical solution for the problem, convergence of the solution is evaluated by tracking the maximum vertical nodal displacement. The convergence study for the integration error is presented in Fig. 8. A 2x2 Gaussian quadrature is used at each integration cell. We notice that even for $\tau = 10^{-1}$ the relative solution error is approximately 1%, which may be sufficiently accurate for the sample problem; nevertheless, for demonstration purposes, we consider that an accurate solution has been obtained for a relative integration error tolerance $\tau = 10^{-3}$. We decreased the relative integration error tolerance 10 times for each simulation; a smaller decrease may result in finer control of accuracy, leading to a higher value for τ and therefore reducing the number of integration points.

Fig. 8. Convergence study for the integration error. (a) Variation of maximum vertical displacement against relative integration error tolerance. (b) Resulting position of integration points for $\tau = 10^{-3}$.

Compared to displacement-based finite element formulations, where under-integration generally produces less stiff elements, the effect of integration error on the solution of mesh-free methods is unpredictable, as shown by the non-uniform convergence in Fig. 8(a).

The convergence study for the discretisation error is presented in Fig. 9. The adaptive quadrature with the relative integration error tolerance $\tau = 10^{-3}$, as identified from the integration error convergence study, is used. Node density is increased at each step by adding a node between each pair of neighbouring nodes, with mesh spacing h taken as the maximum distance between adjacent nodes in the discretisation. We notice that a converged solution is obtained after the node density is increased twice.

Fig. 9. Convergence study for the nodal density. (a) Variation of maximum vertical displacement against nodal spacing when using an adaptive quadrature with relative integration error tolerance $\tau = 10^{-3}$. (b) Discretisation that ensures a converged solution (after the second increase in nodal density). (c) Integration point distribution for the converged solution.

3.3 Application example—brain shift simulation

During open brain surgery, after the opening of the skull (craniotomy), the brain changes its shape and position. Therefore, the pre-operative images used for planning the surgery become obsolete.

Biomechanical models can be used to predict the deformation of the brain and register the pre-operative images to the current position and shape of the brain, by using displacements measured on the brain surface in the craniotomy area [23,41,42]. The computational model can be created based on the acquired intra-operative images, but the actual computations can only be performed during surgery, after the opening of the skull; the solution method needs to be quick and accurate. Because they can handle large deformations, meshless methods are well suited for computing the solution. Nevertheless, we must first investigate if a chosen integration scheme leads to acceptable solution accuracy. This example demonstrates how the proposed adaptive integration algorithm can be used to answer this question.

3.3.1 Problem description

We simulate the brain shift using a 2D slice through the brain (Fig. 10). The model parameters are chosen based on our previous modelling experience [23,41,42]. The Young's modulus is set to 3000 Pa for the brain parenchyma and 6000 Pa for the tumour. The Poisson's ratio is set to 0.49 for both parenchyma and tumour, to account for the almost incompressible behaviour of brain tissue. The ventricles are modelled as a cavity, because the cerebrospinal fluid can freely move in and out of them. Displacements are enforced on the nodes of the brain surface exposed by craniotomy. The interaction between skull and brain is modelled as finite sliding, frictionless contact and the skull is assumed to be rigid as it is orders of magnitude stiffer than the brain tissue. The brain model is discretised using 707 nodes, and a triangular background grid is used for integration. Linear basis functions and a constant influence domain (circles of radius $R=8$) is used in the creation of the meshless shape functions. The same solution method as in the previous example is used – the MTLED algorithm and dynamic relaxation. We would like to use a low number of integration points in order to reduce the computation time. Therefore, we investigate if 3 integration points for each triangular integration cell leads to sufficient solution accuracy.

3.3.2 Integration error convergence study

The adaptive quadrature with the relative integration error tolerance $\tau = 10^{-3}$ was used to obtain a reference solution with high integration accuracy (using more than 100 thousands integration points). Three different integration schemes (3 integration points per triangular integration cell, adaptive integration with $\tau = 0.01$, and adaptive integration with $\tau = 0.1$) were then used to solve the problem and the difference in computed nodal displacements between the obtained solutions and the reference solution was recorded; the maximum differences in nodal displacements are presented in Fig. 10(c).

The results show that using 3 integration points per cell leads to a maximum difference in nodal displacements compared to the reference solution of 0.048 mm. The variation of the displacement difference over the problem domain is presented in Fig. 10(b); it shows that in the area of interest (tumor area) the displacement difference is even smaller than the maximum. Given that the medical images used for guiding the surgery rarely have sub-millimetre accuracy, we can conclude that, for this specific problem, using 3 integration points in each background integration cell leads to a sufficiently accurate solution.

Fig. 10. Influence of integration accuracy on solution accuracy. (a) Problem domain, meshless discretisation using nodes and background integration cells. (b) Difference in displacements between the solution computed using 3 integration points per integration cell and the reference solution computed using relative integration error tolerance $\tau = 10^{-3}$. (c) Variation of the maximum difference in displacements between solutions computed using different integration accuracies and the reference solution.

4. Discussion and Conclusions

This paper investigates the application of adaptive quadrature to solid mechanics problems in engineering analysis using the element-free Galerkin method in order to obtain accurate reference solutions. The presented adaptive quadrature algorithm provides a method for controlling the integration error in Galerkin mesh-free methods. The generation of a distribution of integration points and weights that will accurately integrate stiffness matrices is automatic - no analyst input is required beyond the imposition of a relative integration error tolerance. The algorithm adaptively responds to large variation in the integrand by dividing integration cells, with a simple local relative error estimate directing the algorithm to generate additional integration points only in local regions where the integrand varies significantly. The adaptive algorithm can be applied in other quadrature methods which require computation of an integral over a given integration cell, such as the partition of unity quadrature, which uses overlapping integration cells.

The key feature of the adaptive quadrature algorithm is the ability for the user to control integration accuracy using the algorithm parameter τ . This results in a reliable and responsive numerical integration scheme, permitting efficient and accurate integration of stiffness matrices through the selection of an appropriate relative integration error tolerance. This forms the basis for the convergence study proposed in this paper, which allows the user to independently examine and control the contribution of integration and discretisation errors to the overall solution error. Through selection of an optimal integration error tolerance (which guarantees integration accuracy) and node density (which guarantees discretisation accuracy), an accurate solution can be achieved, as demonstrated by the numerical examples. The presented procedure allows the computation of reference solutions for very challenging problems which do not have an analytical or even a finite element solution (such as very large deformation problems).

References

- [1] Liu GR, Gu YT. An Introduction to Meshfree Methods and Their Programming. Dordrecht: Springer; 2005.
- [2] Liu G. Meshfree Methods: Moving Beyond the Finite Element Method. 2nd ed. CRC Press; 2009.
- [3] Belytschko T, Lu YY, Gu L. Element-free Galerkin methods. *Int J Numer Meth Engng* 1994;37:229–56. doi:10.1002/nme.1620370205.

- [4] Nayroles B, Touzot G, Villon P. Generalizing the finite element method: Diffuse approximation and diffuse elements. *Computational Mechanics* 1992;10:307–18. doi:10.1007/BF00364252.
- [5] Lancaster P, Salkauskas K. Surfaces generated by moving least squares methods. *Math Comp* 1981;37:141–58. doi:10.1090/S0025-5718-1981-0616367-1.
- [6] Joldes GR, Wittek A, Miller K. Adaptive numerical integration in Element-Free Galerkin methods for elliptic boundary value problems. *Engineering Analysis with Boundary Elements* 2015;51:52–63. doi:10.1016/j.enganabound.2014.10.007.
- [7] Dolbow J, Belytschko T. Numerical integration of the Galerkin weak form in meshfree methods. *Computational Mechanics* 1999;23:219–30. doi:10.1007/s004660050403.
- [8] 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. doi:10.1016/S0045-7825(96)01078-X.
- [9] Beissel S, Belytschko T. Nodal integration of the element-free Galerkin method. *Computer Methods in Applied Mechanics and Engineering* 1996;139:49–74. doi:10.1016/S0045-7825(96)01079-1.
- [10] Chen J-S, Wu C-T, Yoon S, You Y. A stabilized conforming nodal integration for Galerkin mesh-free methods. *Int J Numer Meth Engng* 2001;50:435–66. doi:10.1002/1097-0207(20010120)50:2<435::AID-NME32>3.0.CO;2-A.
- [11] Quak W, Boogaard AH van den, González D, Cueto E. A comparative study on the performance of meshless approximations and their integration. *Comput Mech* 2011;48:121–37. doi:10.1007/s00466-011-0577-6.
- [12] Atluri SN, Kim H-G, Cho JY. A critical assessment of the truly Meshless Local Petrov-Galerkin (MLPG), and Local Boundary Integral Equation (LBIE) methods. *Computational Mechanics* 1999;24:348–72. doi:10.1007/s004660050457.
- [13] De S, Bathe K-J. The method of finite spheres. *Computational Mechanics* 2000;25:329–45. doi:10.1007/s004660050481.
- [14] De S, Bathe K-J. Towards an efficient meshless computational technique: the method of finite spheres. *Engineering Computations* 2001;18:170–92. doi:10.1108/02644400110365860.
- [15] De S, Bathe K-J. The method of finite spheres with improved numerical integration. *Computers & Structures* 2001;79:2183–96. doi:10.1016/S0045-7949(01)00124-9.

- [16] Racz D, Bui TQ. Novel adaptive meshfree integration techniques in meshless methods. *Int J Numer Meth Engng* 2012;90:1414–34. doi:10.1002/nme.4268.
- [17] Carpinteri A, Ferro G, Ventura G. The partition of unity quadrature in meshless methods. *Int J Numer Meth Engng* 2002;54:987–1006. doi:10.1002/nme.455.
- [18] Duflo M, Nguyen-Dang H. A truly meshless Galerkin method based on a moving least squares quadrature. *Commun Numer Meth Engng* 2002;18:441–9. doi:10.1002/cnm.503.
- [19] Rosca VE, Leitão VMA. Quasi-Monte Carlo mesh-free integration for meshless weak formulations. *Engineering Analysis with Boundary Elements* 2008;32:471–9. doi:10.1016/j.enganabound.2007.10.015.
- [20] BaniHani S, De S. Development of a genetic algorithm-based lookup table approach for efficient numerical integration in the method of finite spheres with application to the solution of thin beam and plate problems. *Int J Numer Meth Engng* 2006;67:1700–29. doi:10.1002/nme.1678.
- [21] Khosravifard A, Hematiyan MR. A new method for meshless integration in 2D and 3D Galerkin meshfree methods. *Engineering Analysis with Boundary Elements* 2010;34:30–40. doi:10.1016/j.enganabound.2009.07.008.
- [22] Oden JT, Belytschko T, Babuska I, Hughes TJR. Research directions in computational mechanics. *Computer Methods in Applied Mechanics and Engineering* 2003;192:913–22. doi:10.1016/S0045-7825(02)00616-3.
- [23] Miller K, Wittek A, Joldes GR, Horton A, Roy TD, Berger J, et al. Modelling brain deformations for computer-integrated neurosurgery. *International Journal for Numerical Methods in Biomedical Engineering* 2010;26:117 – 138. doi:10.1002/cnm.1260.
- [24] Duarte CA, Oden JT. An h-p adaptive method using clouds. *Computer Methods in Applied Mechanics and Engineering* 1996;139:237–62. doi:10.1016/S0045-7825(96)01085-7.
- [25] Häussler-Combe U, Korn C. An adaptive approach with the Element-Free-Galerkin method. *Computer Methods in Applied Mechanics and Engineering* 1998;162:203–22. doi:10.1016/S0045-7825(97)00344-7.
- [26] Liu GR, Tu ZH. An adaptive procedure based on background cells for meshless methods. *Computer Methods in Applied Mechanics and Engineering* 2002;191:1923–43. doi:10.1016/S0045-7825(01)00360-7.

- [27] Rossi R, Alves MK. An h-adaptive modified element-free Galerkin method. *European Journal of Mechanics - A/Solids* 2005;24:782–99. doi:10.1016/j.euromechsol.2005.03.010.
- [28] Rabczuk T, Belytschko T. Adaptivity for structured meshfree particle methods in 2D and 3D. *Int J Numer Meth Engng* 2005;63:1559–82. doi:10.1002/nme.1326.
- [29] Nguyen VP, Rabczuk T, Bordas S, Duflot M. Meshless methods: A review and computer implementation aspects. *Mathematics and Computers in Simulation* 2008;79:763–813. doi:10.1016/j.matcom.2008.01.003.
- [30] Timoshenko S, Goodier JN. *Theory of Elasticity*. McGraw-Hill Book Company, Inc.; 1951.
- [31] Horton A, Wittek A, Joldes GR, Miller K. A meshless Total Lagrangian explicit dynamics algorithm for surgical simulation. *International Journal for Numerical Methods in Biomedical Engineering* 2010;26:977–98. doi:10.1002/cnm.1374.
- [32] Miller K, Joldes G, Lance D, Wittek A. Total Lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation. *Commun Numer Meth Engng* 2007;23:121–34. doi:10.1002/cnm.887.
- [33] Zhang G, Wittek A, Joldes GR, Jin X, Miller K. A three-dimensional nonlinear meshfree algorithm for simulating mechanical responses of soft tissue. *Engineering Analysis with Boundary Elements* 2014;42:60–6. doi:10.1016/j.enganabound.2013.08.014.
- [34] Miller K, Horton A, Joldes GR, Wittek A. Beyond finite elements: a comprehensive, patient-specific neurosurgical simulation utilizing a meshless method. *Journal of Biomechanics* 2012;45:2698–701.
- [35] Joldes GR, Wittek A, Miller K. Real-Time Nonlinear Finite Element Computations on GPU - Application to Neurosurgical Simulation. *Computer Methods in Applied Mechanics and Engineering* 2010;199:3305–14. doi:10.1016/j.cma.2010.06.037.
- [36] Joldes GR, Wittek A, Miller K. Real-time nonlinear finite element computations on gpu – handling of different element types. In: Wittek A, Nielsen PMF, Miller K, editors. *Computational Biomechanics for Medicine: Soft Tissues and Musculoskeletal System*, Springer New York; 2011, p. 73–80.
- [37] Chowdhury HA, Joldes GR, Wittek A, Doyle B, Pasternak E, Miller K. Implementation of a Modified Moving Least Squares Approximation for Predicting Soft Tissue Deformation Using a

Meshless Method. In: Doyle B, Miller K, Wittek A, Nielsen PMF, editors. Computational Biomechanics for Medicine, Springer International Publishing; 2015, p. 59–71.

- [38] Joldes GR, Wittek A, Miller K. Computation of intra-operative brain shift using dynamic relaxation. *Computer Methods in Applied Mechanics and Engineering* 2009;198:3313–20. doi:10.1016/j.cma.2009.06.012.
- [39] Joldes GR, Wittek A, Miller K. An adaptive Dynamic Relaxation method for solving nonlinear finite element problems. Application to brain shift estimation. *International Journal for Numerical Methods in Biomedical Engineering* 2011;27:173–85. doi:10.1002/cnm.1407.
- [40] Joldes GR, Wittek A, Miller K. Suite of finite element algorithms for accurate computation of soft tissue deformation for surgical simulation. *Medical Image Analysis* 2009;13:912–9. doi:10.1016/j.media.2008.12.001.
- [41] Zhang JY, Joldes GR, Wittek A, Miller K. Patient-specific computational biomechanics of the brain without segmentation and meshing. *International Journal for Numerical Methods in Biomedical Engineering* 2013;29:293–308.
- [42] Garlapati RR, Roy A, Joldes GR, Wittek A, Mostayed A, Doyle B, et al. More accurate neuronavigation data provided by biomechanical modeling instead of rigid registration. *Journal of Neurosurgery* 2014;120:1477–83.