

# **A New Method for Essential Boundary Conditions Imposition in Explicit Meshless Methods**

Grand Roman Joldes\*, Habib Chowdhury, Adam Wittek and Karol Miller  
Intelligent Systems for Medicine Laboratory, School of Mechanical and Chemical Engineering,  
The University of Western Australia, 35 Stirling Highway, Crawley/Perth WA 6009, AUSTRALIA

## **Abstract**

In this paper we present a new technique of enforcing Essential Boundary Conditions (EBC) in Meshless Methods (MM) based on the Element Free Galerkin (EFG) principles. Imposing EBC is a fundamental issue in MMs. The imposition of prescribed displacement values on the boundary in MM based on approximating shape functions is not as straightforward as in the Finite Element Method (FEM) because the meshless shape functions are generally not interpolating at nodes. Furthermore, many techniques of enforcing EBC are not compatible with explicit time integration schemes. This paper describes a new method of imposing EBC in EFG based MM suitable for explicit time integration, named Essential Boundary Conditions Imposition in Explicit Meshless (EBCIEM). The effectiveness of the proposed method is demonstrated using both 2D and 3D numerical examples.

**Keywords:** essential boundary conditions; meshless methods; explicit time integration; Element Free Galerkin; non-interpolating shape functions

---

\* Corresponding author. Tel: +61-8-6488-3125, Fax: +61-8-6488-1024

Email: grand.joldes@uwa.edu.au

## 1. Introduction

Meshless Methods (MM) [1, 2] have evolved over the years as an alternative approach which addresses some of the shortcomings of the Finite Element Method (FEM). MMs use only a set of scattered nodes in the problem domain and on its boundary to discretize the problem space. Therefore, the need for creating a complex predefined mesh is eliminated in MMs. Furthermore, MMs are suitable for many applications involving large deformations where traditional FEM have difficulties in obtaining a solution due to element distortion.

However, despite rapid progress in the development of MM in recent years, some aspects still require further research; one such aspect is the imposition of Essential Boundary Conditions (EBC) in the Element Free Galerkin (EFG) meshless methods [3]. The difficulty in imposing EBC in MMs arises from the properties of the MM shape functions. Shape functions are necessary to approximate the unknown field functions and to find the approximate solution to a problem governed by given differential equations and boundary conditions. Using the shape functions, the values of the field variable computed at the discretisation nodes are used to approximate the values at non-nodal points using interpolation. In FEM, the shape functions are interpolating and the values of the field variables are explicitly calculated at nodes; therefore, prescribing the required values to the field variables corresponding to the essential boundary nodes is sufficient to enforce EBC. However, in MMs, the shape functions are created with overlapping support domains and in many cases are not interpolating at nodes; prescribing values to the field variables corresponding to the essential boundary nodes is not sufficient to enforce the desired nodal displacements at these nodes [2]. For example, the Moving Least Squares (MLS) [4, 5] shape functions have been used predominantly in EFG based MMs due to the smoothness, continuity and consistency of the approximation field they create. However, the shape function associated with a node generally does not vanish at all the other nodes. MLS shape functions are non-polynomial, non-interpolating rational functions and imposing essential boundary conditions is not as trivial as in the case of FEM.

Many techniques have been developed over the years to overcome the difficulties associated with the imposition of EBC in MMs [1, 6, 7]. These methods can be broadly

categorized into two main groups: methods based on the *modification of the weak form*, and methods based on *modified shape functions*.

The general approach of the first methods group is to change the variational or weighted residual formulation of the problem. Among these approaches, the Lagrange multiplier and the Penalty methods are found to be most widely used. For example, Belytschko et al. [3] uses the Lagrange multiplier method by altering the variational problem formulation to enforce EBC. This method introduces additional unknowns, called Lagrange multipliers, to the Galerkin weak form. As the trial functions do not satisfy the essential boundary conditions, these are imposed using the Lagrange multipliers. Similarly, in case of the Penalty method [8], a penalty formulation is devised in order to account for the EBC in the standard principle of minimum potential energy. The penalty formulation results into a diagonal matrix of penalty factors which are generally scalar parameters and, typically, require large values in order to enforce EBC effectively.

The second group of methods manages essential boundary condition imposition by modifying the shape functions into (almost) interpolating shape functions; therefore, EBC can be imposed in the same way as for FEM. For example, the use of singular weight functions [5] and regularized weight functions [9] to enforce almost interpolating properties of shape functions at nodes have been presented in earlier research. The modification of the shape functions can also be done by blending them with finite element shape functions on the essential boundary [10-13]. In this case, a layer of finite elements along the essential boundaries is created and the shape functions of the finite elements are combined with the shape functions of the meshless method by using ramp functions [1].

However, most of these aforementioned methods have their applicability limited only to implicit solution methods, which require solutions to large systems of algebraic equations [14, 15]. Therefore, due to its low computational cost and straightforward treatment of non-linearities [15], explicit methods have become the preferred choice for many problems which involves large deformations and large strains with geometric and material non-linearities [16-18]. The Meshless Total Lagrangian Explicit Dynamics (MTLED) [19] is one of such methods which has already been established as a robust and efficient method in solving soft tissue large deformations problems [20, 21]. However, traditional methods of imposing EBC are not applicable in MTLED or exhibits serious drawbacks, as explained in the following

section. Therefore, imposing EBC in MMs for explicit time integration solution methods presents itself as a challenge.

This paper proposes a new approach of enforcing the essential boundary conditions in EFG based MTLED framework. We call this new method *Essential Boundary Conditions Imposition in Explicit Meshless* (EBCIEM). The paper is organized as follows: the motivation of EBCIEM in the context of explicit integration for elasticity problems is discussed in the next Section; the methodology of EBCIEM is presented in Section 3; numerical examples are presented in Section 4; followed by discussion and conclusions in Section 5.

## **2. Explicit Time Integration and EBC**

In this section we identify the shortcomings of existing methods of imposing EBC in the context of explicit time integration for elasticity problems.

Explicit time integration is a direct time integration method for the equations of dynamics where nodal accelerations are found directly without any iterations and then integrated to obtain the displacements [14, 15]. There is no need to assemble a global stiffness matrix. An explicit time integration scheme (i.e. central difference method) leads to a system of equations which is decoupled by using a diagonal mass matrix. The methods of imposing EBC by modifying the weak form (the Lagrange multiplier and Penalty methods) introduce additional unknowns in the weak form, leading to a system of equations that cannot be decoupled and, therefore, cannot be solved efficiently using explicit time integration.

The coupled Finite Element and meshless approach is the most efficient and accurate method for imposing essential boundary conditions when using explicit time integration [10-13]. However, this requires the creation of a finite element layer along the essential boundary, which is one of the problems that the use of MMs tries to eliminate.

The EBC can also be directly imposed in MMs by modifying the meshless shape functions to make them almost interpolating at nodes. However, this approach of imposing EBC can produce difficulties in the context of explicit time integration. Explicit time integration is only conditionally stable and a restriction on the time step needs to be enforced in order to obtain stable simulation results; it requires an estimation of the maximum stable

time step that can be used [14]. According to Joldes et al. [22], the stable critical time step for central difference integration, when mass lumping is used, can be estimated as:

$$\Delta t_{crit} = \frac{2}{\omega_{max}} = \frac{2}{\sqrt{\lambda_{max}}} \approx \min_I \left( \frac{2}{\sqrt{\lambda_{Max}^I}} \right) \quad (1)$$

where  $\omega_{max}$  is the maximum frequency of free vibration and  $\lambda_{max}$  is the maximum eigenvalue of the stiffness matrix. The bound for the maximum eigenvalue at an integration point  $I$  is given by:

$$\lambda_{Max}^I \leq \frac{N^I}{\rho^I} (\lambda + 2\mu) \cdot \|\mathbf{B}^I\|_F^2 = N^I c^{I2} \cdot \mathbf{B}_{jI}^I \mathbf{B}_{jI}^I \quad (2)$$

where  $N^I$  is the number of nodes associated with the integration point  $I$ ,  $\rho$  is the material density,  $\lambda$  and  $\mu$  are the Lamé parameters,  $c$  is the dilatational wave speed, and  $\mathbf{B}$  is the matrix of shape function derivatives (SFD).

As can be seen from Eq (1) and (2), the stable critical time step is heavily influenced by the values of SFD. If two discretization nodes are too close to each other, the value of the SFD increases as the slope of the shape function becomes very steep, as shown in Figure 1. This affects the estimated stable critical time-step leading to a very small value, which results in a large number of steps in the simulation and reduces the computational efficiency.

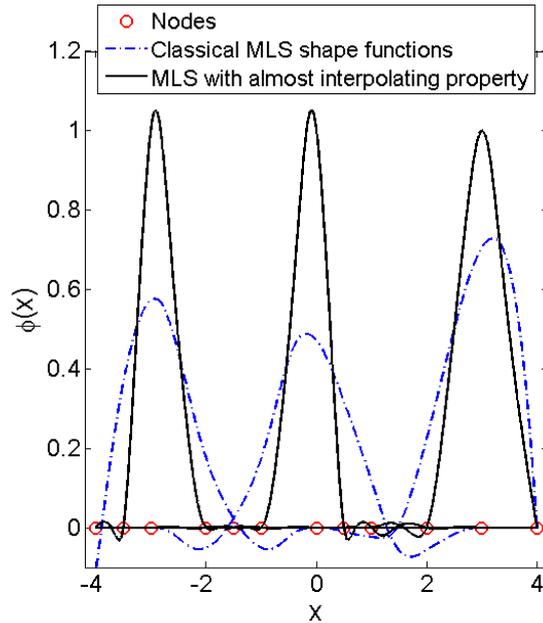


Figure 1: Comparison between classical MLS shape functions and almost interpolating MLS shape functions for 3 nodes in the domain.

### 3. Derivation of the Essential Boundary Conditions Imposition in Explicit Meshless (EBCIEM) Method

We start our derivation from the global system of discretized damped equation of motion [17, 19]:

$$\mathbf{M}^t \ddot{\mathbf{u}} + {}^t\mathbf{F}_{damp} + {}^t\mathbf{F}_{int} = {}^t\mathbf{F}_{ext} \quad (3)$$

where  $\mathbf{u}$  is the vector of nodal displacements,  $\mathbf{M}$  is the mass matrix,  ${}^t\mathbf{F}_{int}$  is the global nodal reaction force vector,  ${}^t\mathbf{F}_{damp}$  is the damping force and  ${}^t\mathbf{F}_{ext}$  is the vector of externally applied force at time  $t$ . For simplicity, the method is derived in 2D. We use the notations from Bathe [14].

We consider that the externally applied force can be subdivided into two parts:

$${}^t\mathbf{F}_{ext} = {}^t\mathbf{F}_{ne} + {}^t\mathbf{F}_e \quad (4)$$

where  ${}^t\mathbf{F}_e$  is the force that is externally applied only on the essential boundary and  ${}^t\mathbf{F}_{ne}$  is the force that is applied on the non-essential boundary. Replacing Eq (4) in Eq (3) and rearranging we obtain:

$$\mathbf{M}^t \ddot{\mathbf{u}} + {}^t\mathbf{F}_{damp} = ({}^t\mathbf{F}_{ne} - {}^t\mathbf{F}_{int}) + {}^t\mathbf{F}_e \quad (5)$$

The damping force is defined as a mass proportional damping to enable the decoupling of equations for explicit time integration and efficient convergence to the steady state solution [23]. This force can be defined as  ${}^t\mathbf{F}_{damp} = c\mathbf{M}^t \dot{\mathbf{u}}$ , where  $c$  is the damping coefficient; Eq (5) therefore becomes:

$$\mathbf{M}^t \ddot{\mathbf{u}} + c\mathbf{M}^t \dot{\mathbf{u}} = ({}^t\mathbf{F}_{ne} - {}^t\mathbf{F}_{int}) + {}^t\mathbf{F}_e \quad (6)$$

In EFG based MMs  ${}^t\mathbf{F}_e$  is calculated as:

$${}^t\mathbf{F}_e = \int_{\Gamma_e} {}^t\Phi(s) {}^t\mathbf{T}(s) d\Gamma_e \quad (7)$$

where  $\Phi(s)$  are the meshless shape functions,  $s$  is the arc-length along the essential boundary and  $\mathbf{T}$  is the distributed force on the essential boundary. We define  ${}^t\mathbf{T}(s)$  as an interpolation of the values of the distributed force at the essential boundary nodes:

$${}^t\mathbf{T}(s) = \sum_{k=1}^{n_e} N^k(s) {}^t\mathbf{T}_e^k \quad (8)$$

where  $N(s)$  are the shape functions used for interpolation and  $n_e$  is the number of essential boundary nodes. The number of elements in the vector  $\mathbf{T}_e$  corresponds to the number of degrees of freedom constrained on the essential boundary, as shown in Figure 2.

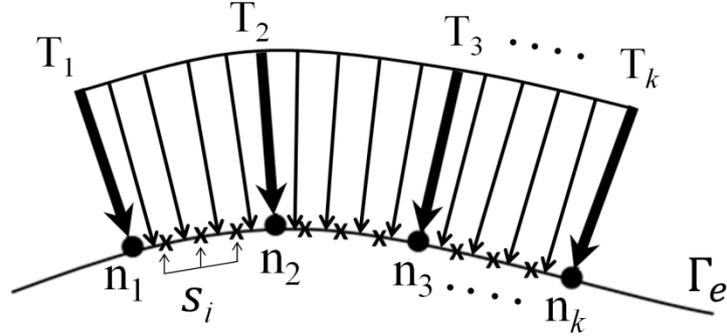


Figure 2: Definition of the distributed force on the essential boundary.

Replacing Eq (8) in Eq (7) we obtain:

$${}^t\mathbf{F}_e = \int_{\Gamma_e} {}^t\Phi(s) \sum_{k=1}^{n_e} N^k(s) {}^t\mathbf{T}_e^k d\Gamma_e \quad (9)$$

Eq (9) needs to be integrated numerically. If we use Gaussian quadrature over the essential boundary, we obtain:

$${}^t\mathbf{F}_e = \sum_{i=1}^{n_g} {}^t\Phi(s_i) \sum_{k=1}^{n_e} N^k(s_i) {}^t\mathbf{T}_e^k w_i \quad (10)$$

where  $s_i$  and  $w_i$  are the Gauss quadrature points and weights, respectively, and  $n_g$  is the total number of integration points along the essential boundary segment. Eq (10) can be written in matrix form as:

$${}^t\mathbf{F}_e = {}^t\mathbf{V} {}^t\mathbf{T}_e \quad (11)$$

with

$${}^t\mathbf{V}_{jk} = \sum_{i=1}^{n_g} {}^t\Phi_j(s_i) N^k(s_i) w_i \quad (12)$$

Different types of shape functions (finite element, meshless) can be used to interpolate the distributed force on the essential boundary. Nevertheless, for meshless shape functions the support domains of the nodes are difficult to define on curved boundaries and numerical integration required above is, in general, more challenging. Therefore, we use finite element shape functions in equation (8).

In order to solve the equation of motion, the following central difference expressions are used for the temporal derivatives:

$${}^t\ddot{\mathbf{u}} = ({}^{t+1/2}\dot{\mathbf{u}} - {}^{t-1/2}\dot{\mathbf{u}})/h \quad (13)$$

$${}^{t-1/2}\dot{\mathbf{u}} = ({}^t\mathbf{u} - {}^{t-1}\mathbf{u})/h \quad (14)$$

and

$${}^{t+1/2}\dot{\mathbf{u}} = ({}^{t+1}\mathbf{u} - {}^t\mathbf{u})/h \quad (15)$$

where  $h$  is a fixed time increment,  $t$  represents the  $t^{th}$  time increment and the superimposed dot represents time derivative. The velocity at time  $t$  is computed as:

$${}^t\dot{\mathbf{u}} = ({}^{t+1/2}\dot{\mathbf{u}} + {}^{t-1/2}\dot{\mathbf{u}})/2 \quad (16)$$

By substituting the above expressions for acceleration and velocity in the equation of motion (6) and rearranging, we obtain:

$${}^{t+1}\mathbf{u} - \alpha\mathbf{M}^{-1} {}^t\mathbf{F}_e = {}^t\mathbf{u} + \beta({}^t\mathbf{u} - {}^{t-1}\mathbf{u}) + \alpha\mathbf{M}^{-1}({}^t\mathbf{F}_{ne} - {}^t\mathbf{F}_{int}) \quad (17)$$

where  $\alpha = 2h^2/(2 + ch)$  and  $\beta = (2 - ch)/(2 + ch)$ . The above system of equations is explicit as long as the mass matrix  $\mathbf{M}$  is diagonal. The right hand side of Eq (17) can be identified as the predicted displacement  $\tilde{\mathbf{u}}$  when the load on the essential boundary is ignored:

$${}^{t+1}\tilde{\mathbf{u}} = {}^t\mathbf{u} + \beta({}^t\mathbf{u} - {}^{t-1}\mathbf{u}) + \alpha\mathbf{M}^{-1}({}^t\mathbf{F}_{ne} - {}^t\mathbf{F}_{int}) \quad (18)$$

We augment the system of Eq (18) with the equations for the imposed displacements  ${}^{t+1}\bar{\mathbf{u}}$ , defined as the interpolation of the displacement field variable at time  $t + 1$ :

$$\Phi {}^{t+1}\mathbf{u} = {}^{t+1}\bar{\mathbf{u}} \quad (19)$$

Substituting Eq (11) into Eq (17) and combining with Eq (18) and Eq (19) we obtain the following system of equations:

$$\begin{bmatrix} 1 & -\alpha\mathbf{M}^{-1} {}^t\mathbf{V} \\ \Phi & 0 \end{bmatrix} \begin{bmatrix} {}^{t+1}\mathbf{u} \\ {}^t\mathbf{T}_e \end{bmatrix} = \begin{bmatrix} {}^{t+1}\tilde{\mathbf{u}} \\ {}^{t+1}\bar{\mathbf{u}} \end{bmatrix} \quad (20)$$

Because the number of displacement degrees of freedom on the essential boundary is equal to the number of values in the vector defining the distributed force on the essential boundary, the number of equations in (20) is equal to the number of unknowns. Applying static condensation to Eq (20) to eliminate  ${}^t\mathbf{T}_e$  yields:

$${}^{t+1}\mathbf{u} = {}^{t+1}\tilde{\mathbf{u}} + \mathbf{M}^{-1} {}^t\mathbf{V} (\Phi \mathbf{M}^{-1} {}^t\mathbf{V})^{-1} ({}^{t+1}\bar{\mathbf{u}} - \Phi {}^{t+1}\tilde{\mathbf{u}}) \quad (21)$$

Eq (21) can be written as

$${}^{t+1}\mathbf{u} = {}^{t+1}\tilde{\mathbf{u}} + {}^{t+1}\mathbf{u}_{correction} \quad (22)$$

where

$${}^{t+1}\mathbf{u}_{correction} = {}^t\mathbf{P} ({}^{t+1}\bar{\mathbf{u}} - \Phi {}^{t+1}\tilde{\mathbf{u}}) \quad (23)$$

and

$${}^t\mathbf{P} = \mathbf{M}^{-1} {}^t\mathbf{V} (\Phi \mathbf{M}^{-1} {}^t\mathbf{V})^{-1} \quad (24)$$

While the derivation of the method is relatively complicated, its implementation is relatively straightforward, requiring just the following steps:

1. Discretization of the essential boundary and assembly of matrix  ${}^t\mathbf{V}$  using Eq (12);
2. Calculation of matrix  ${}^t\mathbf{P}$  using Eq (24);
3. Computation of the displacement correction using Eq (23) and its application using Eq. (22).

The procedure for enforcing EBC described above is especially effective in the context of the Total Lagrangian formulation, as the meshless shape functions, and therefore matrix  ${}^t\mathbf{P}$ , are constant during the time stepping procedure and can be precomputed; therefore steps 1 and 2 described above are only required once. As can be seen from Eq (22), a displacement correction is added at every time step to the displacement field, enforcing the EBS in a prediction-correction fashion. The computation flowchart in the context of MTLED [19] is as follows (the additional steps are marked with a +):

**Pre-computation stage:**

1. Load geometry and boundary conditions;
2. Create integration points;
3. For each integration point compute element compute the shape functions  $\Phi(s)$  and their derivatives;
4. Compute the diagonal mass matrix  $\mathbf{M}$ ;
5. + Discretize the essential boundary and assembly of matrix  ${}^t\mathbf{V}$  using Eq (12);
6. + Calculate matrix  ${}^t\mathbf{P}$  using Eq (24);
7. Initialise nodal displacement  ${}^0\mathbf{u} = \mathbf{0}$ ,  ${}^{-\Delta t}\mathbf{u} = \mathbf{0}$ ;

**Time stepping:**

*In every time step t:*

1. Loop through the integration points:
  - Compute deformation gradient using nodal displacements from the previous time step
  - Compute the stress using the material law;
  - Compute local reaction forces;
2. Assemble global nodal reaction forces  ${}^t\mathbf{F}_{int}$  using Gaussian quadrature;
3. Compute predicted displacements using central difference formula (18);
4. + Compute the displacement correction using Eq (23) and apply it using Eq. (22).

### 3.1. A Simplified Essential Boundary Conditions Imposition in Explicit Meshless (SEBCIEM)

The presented EBCIEM method involves the computation of integrals on the essential boundary, and therefore requires a discretization of the boundary for numerical integration. We can obtain a simplified version of the method if we lump the distributed force on the essential boundary at the essential boundary nodes. In such a case, the essential boundary force is considered to be acting only at the nodes, and can be simplified to:

$${}^t\mathbf{T}(s) = \sum_{k=1}^{n_e} \delta(s_k) \mathbf{T}_e^k \quad (25)$$

where  $\delta$  is the Dirac Delta function [24]. Replacing Eq (25) in Eq (7) we obtain the force on the essential boundary as:

$${}^t\mathbf{F}_e = \int_{\Gamma_e} {}^t\Phi \sum_{k=1}^{n_e} \delta(s_k) \mathbf{T}_e^k d\Gamma_e \quad (26)$$

As, by definition [24],

$$\int_{\Gamma_e} f(s) \delta(s_k) d\Gamma_e = f(s_k) \quad (27)$$

we have:

$${}^t\mathbf{F}_e = \sum_{k=1}^{n_e} {}^t\Phi(s_k) \cdot \mathbf{T}_e^k \quad (28)$$

Eq (28) can be written in matrix form as:

$${}^t\mathbf{F}_e = {}^t\mathbf{V} \mathbf{T}_e \quad (29)$$

with

$${}^t\mathbf{V}_{jk} = {}^t\Phi_j(s_k) \quad (30)$$

As can be seen from Eq (28), it is no longer necessary to define additional quadrature points along the essential boundary to evaluate  ${}^t\mathbf{F}_e$ . The procedure to obtain the matrix  $\mathbf{P}$  and displacement correction  $\mathbf{u}_{correction}$  is the same as presented in the previous section.

#### 4. Numerical Examples

We present four cases of biomechanics applications to test the performance of EBCIEM and SEBCIEM. The simulation examples presented in the following sub-sections are using the Meshless Total Lagrangian Explicit Dynamics (MTLED) solution method [17, 19]. The adoption of the Total Lagrangian (TL) formulation allows pre-computation of all derivatives with respect to spatial coordinates and the explicit time integration based on the central difference method eliminates the necessity for iterations during each time-step when geometric or material nonlinearities are present. MTLED is based on an EFG discretisation method and uses a background grid for numerical integration.

If the shape functions are considered to be almost interpolating, EBC can be imposed just as in FEM, by allocating the prescribed values to the displacement field variables (we will call this the *correction method*). In Sections 4.1 - 4.3, simulation results obtained by using the proposed methods for imposing EBC are compared with the results obtained using the correction method; this allows us to determine how large are the errors in imposing EBC in the case of using the simple correction method and to justify the need for the proposed methods. In Section 4.4, the EBCIEM and SEBCIEM results are compared with the results obtained by the commercial finite element software ABAQUS. In order to ensure fast convergence of the explicit time integration method towards the steady state solution, we use dynamic relaxation [23, 25].

Numerical integration is required in EBCIEM on the essential boundary to compute the forces, as shown in Eq (10), with Gaussian integration the method of choice. As MLS shape functions are not polynomials, computing integrals which involve such functions is very challenging. Increasing the number of integration points does not always reduce the

integration error; more complex schemes, such as adaptive integration based on domain subdivision [26], can reduce the integration error to any desired level. Nevertheless, a trade-off must be found between the number of Gauss points used and the computational cost of integration. We used 3 integration points for each discretizing segment in 2D and 3 integration points over each discretizing triangle in 3D. While this ensures exact integration for polynomials of degree 5 in 2D and of degree 2 in 3D, the computation of the integrals on the essential boundary is not exact. While this does not affect the imposition of displacements on the essential boundaries, which are always exactly imposed, it does affect the solution on the rest of the domain. Therefore, the solution errors are computed over the entire domain in some of the following examples, in order to capture the effect of the integration errors.

#### ***4.1. Extension of a Soft Tissue Sample in 2D***

We use this simple example to demonstrate the composition of matrices  ${}^t\mathbf{V}$  and  ${}^t\mathbf{P}$ , as well as to show that the proposed method (EBCIEM) imposes EBC exactly.

For the meshless computation, a 2D geometry (10cm x 4cm) is created and the domain and boundary are discretised using 57 nodes. Displacement loading (3cm) is applied to the nodes on the right hand side boundary (5 nodes) and the left boundary nodes are fixed (5 nodes). In order to evaluate Eq (10), 1-D interpolating finite element shape functions are used along the essential boundary and a 3-point Gauss quadrature is used on each essential boundary segment to integrate the applied traction force, as shown in Figure 3. The composition of the sparse constant matrix  ${}^t\mathbf{P}$  described by Eq (24) is shown in Figure 4.b. Multiplications of this constant matrix with the difference between the imposed and predicted displacements are the only additional computations needed at each time-step.

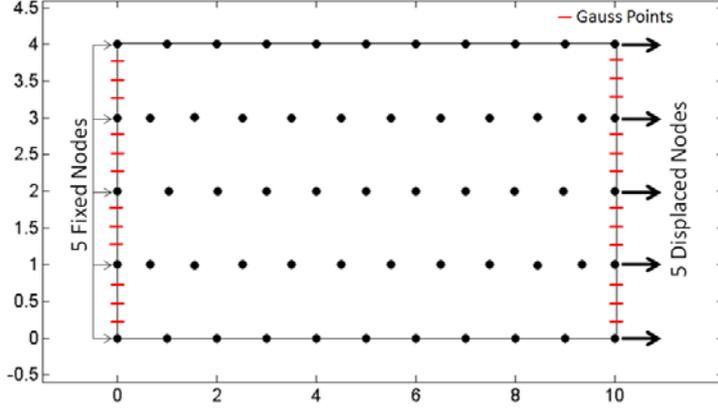


Figure 3: 2D soft tissue geometry. Dimensions are in centimetres. Total nodes:  $n = 57$ ; Essential boundary nodes,  $n_e=10$  (5 fixed nodes and 5 displaced nodes). 3-point Gauss quadrature was used on each essential boundary segment in the evaluation of Eq (10).

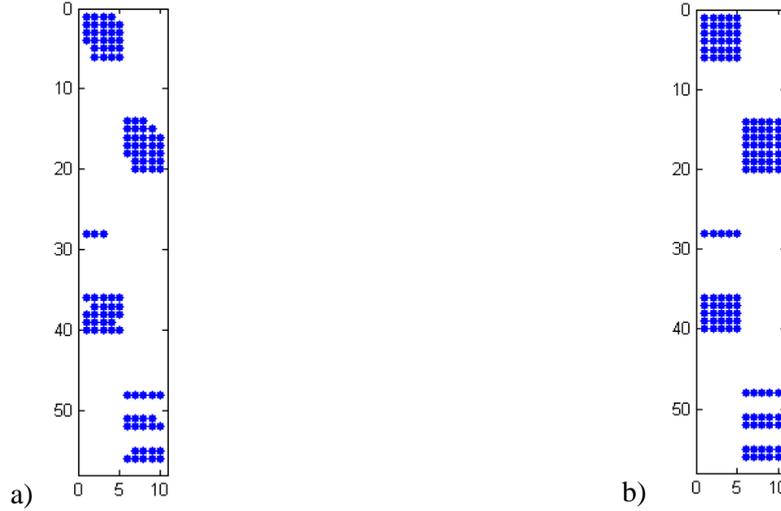


Figure 4: Non-zero elements of a) Assembled  ${}^t\mathbf{V}$  matrix; b) Precomputed constant matrix  ${}^t\mathbf{P}$ .

To ensure integration accuracy, a regular background grid is used consisting of 1000 integration cells with one integration point per cell. The Moving Least Squares (MLS) with quartic spline weight function is used to approximate the unknown displacement field. For each node, the radius of the influence domain is configured as constant,  $R = 2.2$ . An hyper-elastic Neo-Hookean material model is chosen to capture the behaviour of soft tissues undergoing large deformation [23, 27], with the strain energy potential given by:

$$\mathbf{U} = \frac{\mu_0}{2} (\bar{I}_1 - 1) + \frac{k_0}{2} (J - 1)^2 \quad (31)$$

where  $\mu_0$  is the initial shear modulus,  $k_0$  is the bulk modulus and  $\bar{I}_1$  and  $J$  are the first and the third strain invariants.

The simulation results and numerical details are presented in Figure 5 and Table 1.

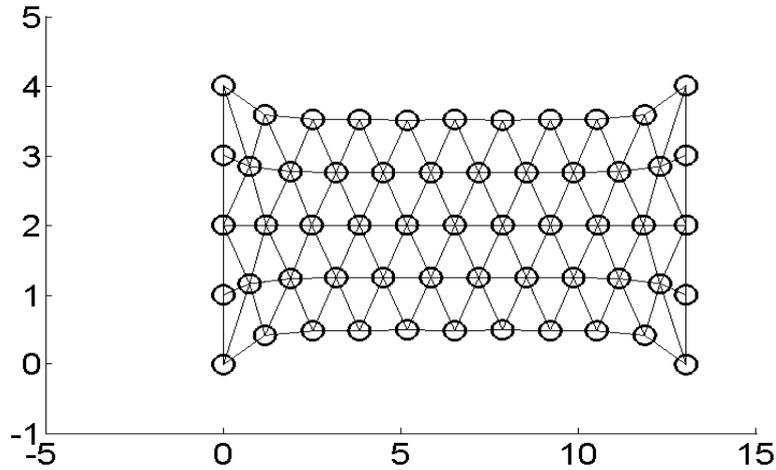


Figure 5 : Extension of soft tissue sample in 2D by using EBCIEM to impose EBC: deformed configuration. Dimensions are in centimetres.

**Table 1.** Positions of essential boundary nodes for the soft-tissue extension in 2D. The numbers reported in the table are shown for 4<sup>th</sup> significant digit, but the EBC are imposed by EBCIEM exactly to machine precision ( $\sim 10^{-15}$ ).

Essential Boundary Nodes	Node ID	Final Nodal Displacements (EBCIEM) (cm)		Final Nodal Displacements (Correction Method) (cm)		Average Difference (cm)	Maximum Difference (cm)
		x	Y	x	y		
Fixed	1	0.0000	4.0000	0.0003	3.9769	0.01394	0.02315
	2	0.0000	3.0000	-0.0089	2.9960		
	3	0.0000	2.0000	-0.0039	2.0000		
	4	0.0000	1.0000	-0.0089	1.0040		
	5	0.0000	0.0000	0.0003	0.0231		
Displaced	15	13.0000	0.0000	12.9997	0.0231		
	16	13.0000	1.0000	13.0089	1.0040		
	17	13.0000	2.0000	13.0039	2.0000		
	18	13.0000	3.0000	13.0089	2.9960		
	19	13.0000	4.0000	12.9997	3.9769		

As can be seen from Table 1, EBCIEM imposes the essential boundary conditions exactly as prescribed; this is expected, because the equations defining the EBC values are added to the system of equations describing the time integration. Because the only additional cost in imposing the essential boundary conditions consists in computing the displacement

correction using Eq. (23), the increase in computation time due to the use of EBCIEM is less than 5% for this problem.

#### ***4.2. Compression of a Soft Tissue Sample in 3D***

In this example we demonstrate the application of EBCIEM in 3D, and evaluate the effects of the simplification introduced in SEBCIEM by lumping the distributed surface forces at the nodes.

We simulate the deformation of a cylinder having the material properties of a soft tissue. The cylinder used in the experiments, shown in Figure 6.a, has a height of 0.1m and radius of 0.05m. One face ( $z = 0\text{m}$ ) of the cylinder is rigidly constrained while the opposite face ( $z = 0.1\text{m}$ ) is displaced. A maximum compressive displacement loading of 0.02m is applied smoothly. An hyper-elastic Neo-Hookean material model with Young's modulus of 3000 Pa, Poisson's ratio of 0.49 and density of  $1000 \text{ kg/m}^3$  are chosen to capture the behaviour of soft tissues as nearly incompressible continuum. The cylinder is discretised with 1252 nodes and 23956 integration points are created from 5989 tetrahedral background integration cell with four integration points per tetrahedron. The Modified Moving Least Squares (MMLS) with quadratic basis and quartic spline weight function is used to approximate the unknown field functions [28, 29]. The same weights are used for all the additional MMLS constraints ( $\mu = 10^{-7}$ ). The influence radius of each node is computed as the average distance between the node and the other nodes of the surrounding 4-node tetrahedron background integration cells multiplied with a constant dilatation parameter ( $D = 2$ ).

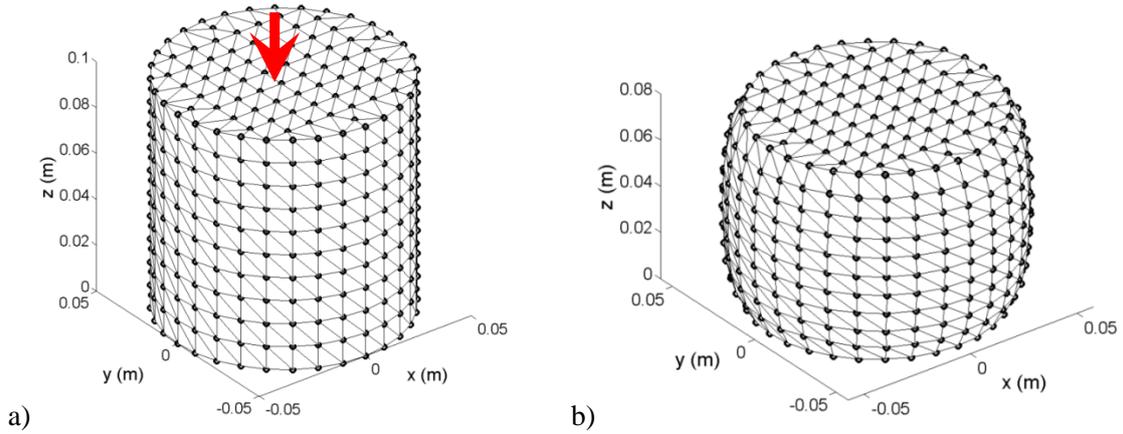


Figure 6: Compression test of a cylinder with the material properties of soft-tissue: a) undeformed configuration; b) deformed configuration.

In case of EBCIEM, the fixed (96 nodes) and displaced (96 nodes) surfaces are triangulated, as shown in Figure 7. 2D finite element shape function and 3-point Gauss quadrature is used in each triangle to integrate the externally applied force on the essential boundary ( ${}^t\mathbf{F}_e$ ). Figure 8 shows the organisation of the assembled  ${}^t\mathbf{V}$  and  ${}^t\mathbf{P}$  matrices as defined by Eq (24). Figure 6.b shows the deformed configuration of the test cylinder and Figure 9 shows the differences in the computed deformation field for the entire domain between EBCIEM and correction method.

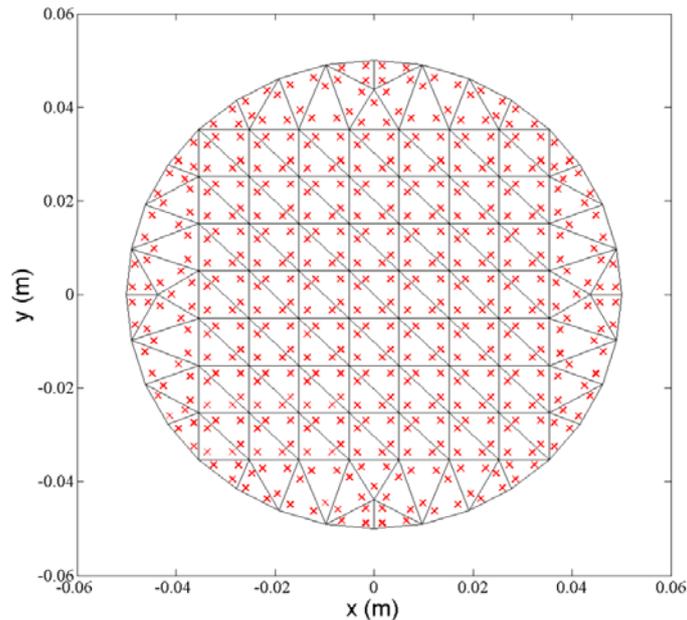


Figure 7: Triangulated essential boundary surface (shown for fixed nodes only). 3-point Gauss quadrature used for each triangle.

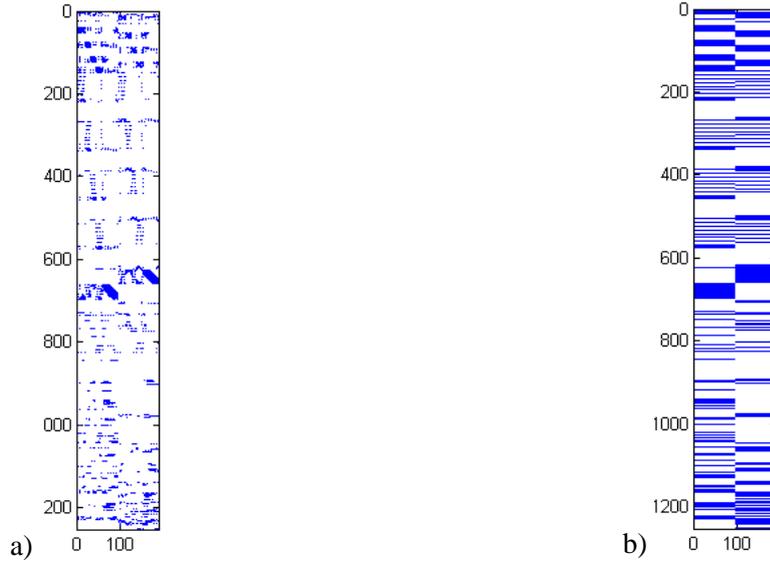


Figure 8: Non-zero elements of a) Assembled  ${}^t\mathbf{V}$  matrix; b) Precomputed constant matrix  ${}^t\mathbf{P}$ .

In case of SEBCIEM, we evaluate the externally applied force on essential boundary  ${}^t\mathbf{F}_e$  using Eq (28). No surface triangulation is necessary in this case. Figure 10 and Figure 11 show the differences of the computed deformation field for the entire domain between SEBCIEM and correction method and between EBCIEM and SEBCIEM respectively. Table 2 shows the average and maximum differences of the computed final positions of the essential boundary nodes.

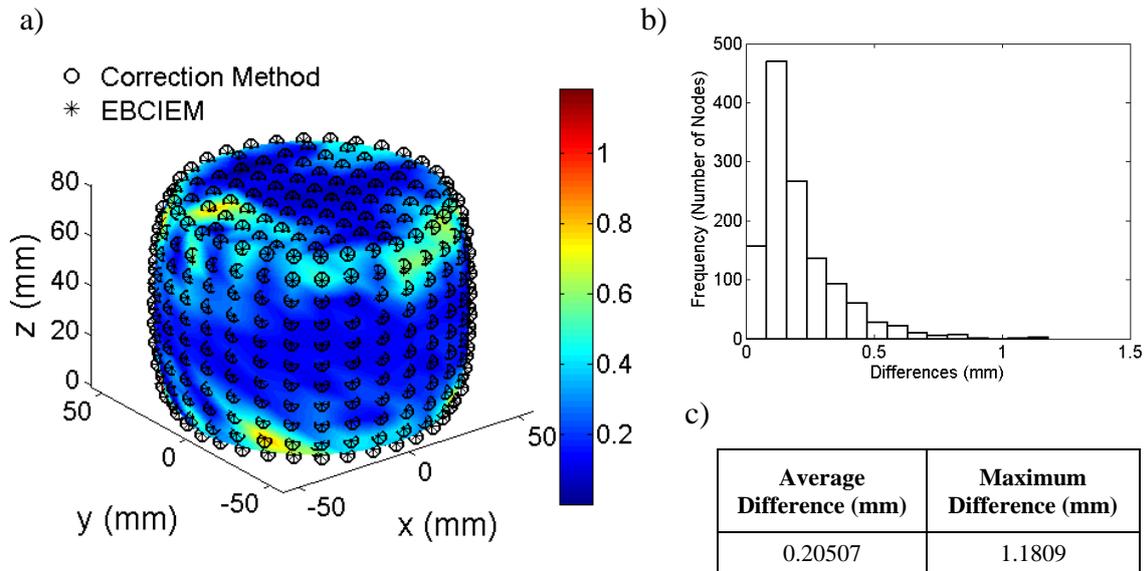


Figure 9: Compression of a Soft Tissue Sample in 3D: a) Differences of the computed deformation field between EBCIEM and correction method; b) histogram of differences; c) average and maximum differences. All dimensions are in millimetres.

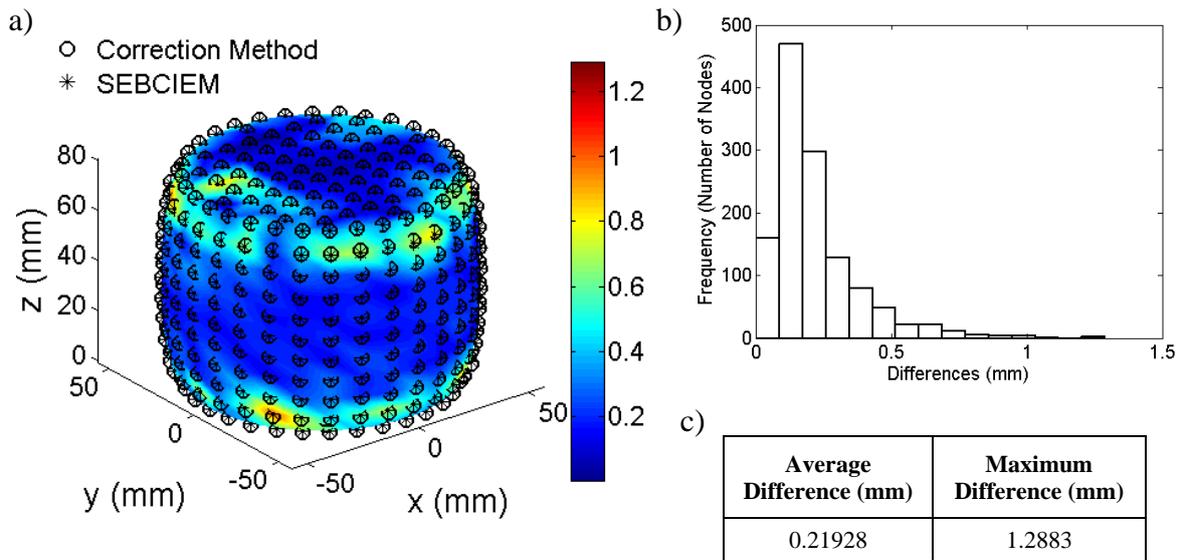


Figure 10: Compression of a Soft Tissue Sample in 3D: a) Differences of the computed deformation field between SEBCIEM and correction method; b) histogram of differences; c) average and maximum differences. All dimensions are in millimetres.

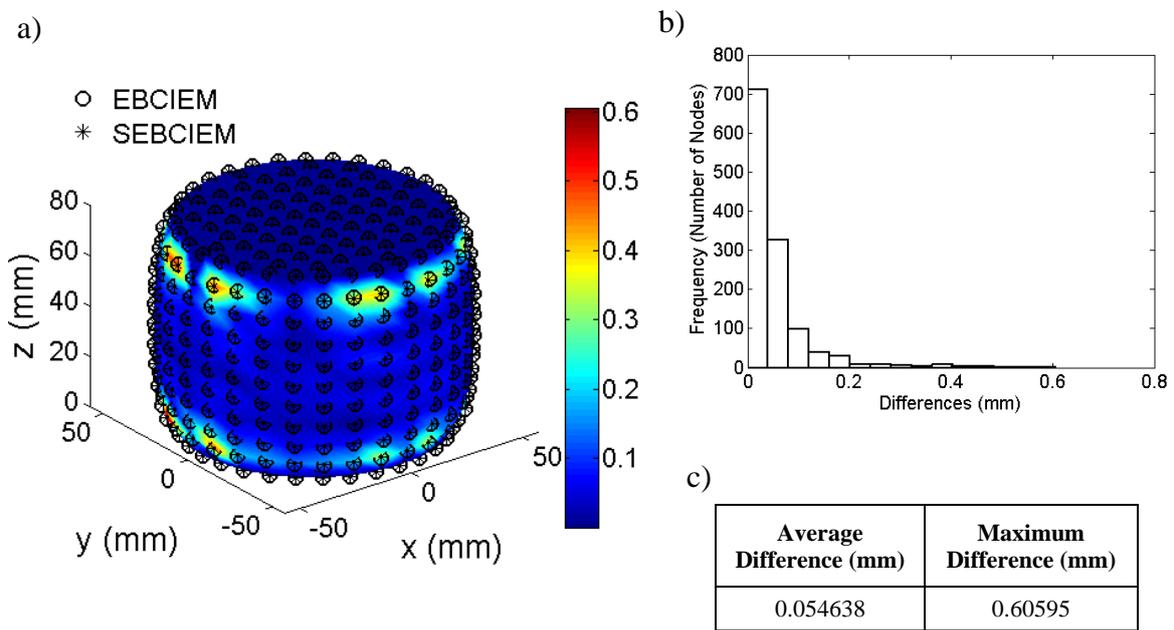


Figure 11: Compression of a Soft Tissue Sample in 3D: a) Differences of the computed deformation field between EBCIEM and SEBCIEM; b) histogram of differences; c) average and maximum differences. All dimensions are in millimetres.

**Table 2.** Compression of a Soft Tissue Sample in 3D: Comparison of average and maximum differences of the final positions of the essential boundary nodes obtained using EBCIEM, SEBCIEM and the correction method.

<b>EB Nodes</b>	<b>Methods</b>	<b>Average Difference (mm)</b>	<b>Maximum Difference (mm)</b>
<i>Fixed Nodes</i>	EBCIEM and Correction Method	0.27056	0.74101
	SEBCIEM and Correction Method	0.27056	0.74101
	EBCIEM and SEBCIEM	$1.5138 \times 10^{-14}$	$6.8982 \times 10^{-14}$
<i>Displaced Nodes</i>	EBCIEM and Correction Method	0.27607	0.81965
	SEBCIEM and Correction Method	0.27607	0.81965
	EBCIEM and SEBCIEM	$8.4023 \times 10^{-14}$	$2.3972 \times 10^{-13}$

Both EBCIEM and SEBCIEM impose the essential boundary conditions exactly to machine precision, as in the previous 2D case study. The simplification in describing the forces on the essential boundary leads to small differences in the computed deformation field within the problem domain which can be acceptable in many applications.

### 4.3. Extension of a Soft Tissue Sample in 3D

The purpose of this example is to demonstrate how SEBCIEM can be used to impose EBC on a set of boundary nodes that do not form a surface. In this case, we reconsider the test case presented in Section 4.2; however, we apply extension displacement loading of 0.02 m only to the edge line of the top surface ( $z = 0.1$  m), as shown in Figure 12.a. The material model, shape functions and numerical integration scheme are the same as described in Section 4.2.

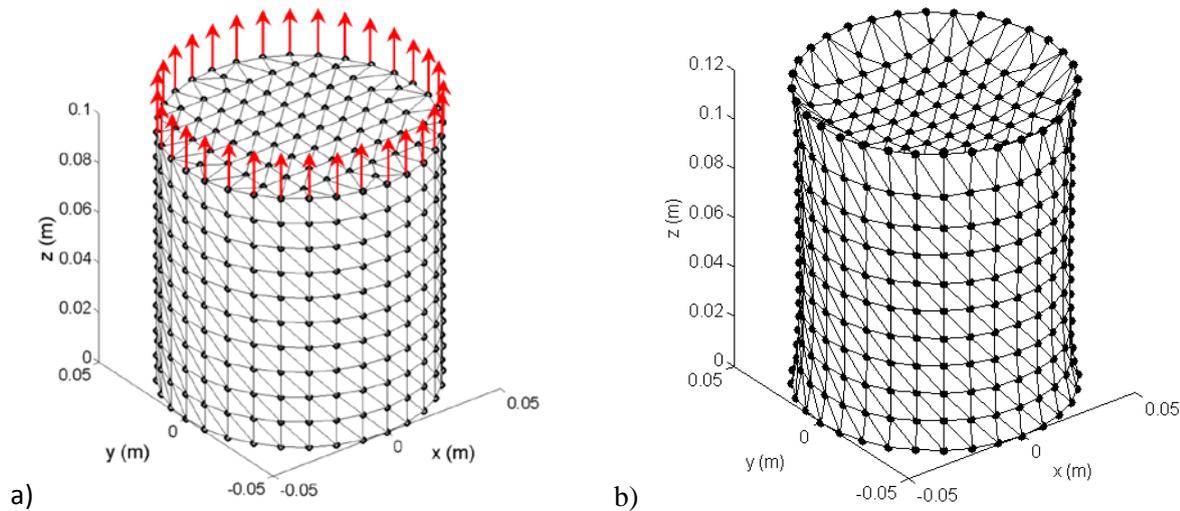


Figure 12: Extension of a cylinder: a) undeformed configuration; b) deformed configuration.

Figure 13 shows the differences of computed deformation field for the entire domain between SEBCIEM and correction method. The histogram plots show the number of nodes that fall into specific value ranges. Furthermore, Table 3 shows the average and maximum differences of the final positions of the EB nodes obtained with the two methods.

**Table 3.** Extension of a Soft Tissue Sample in 3D: Comparison of average and maximum differences of the final positions of the EB nodes obtained using SEBCIEM and Correction method.

EB Nodes	Average Difference (mm)	Maximum Difference (mm)
Fixed Nodes	0.0886	0.3004
Displaced Nodes	3.0385	5.8417

The EBC are imposed exactly by SEBCIEM. The results demonstrate the effectiveness of SEBCIEM in imposing EBC in applications where discretization along the

essential boundary is difficult to achieve. They also demonstrate the importance of proper handling of EBC, as the use of the simple correction method leads to large errors on the boundary, as shown in Table 3.

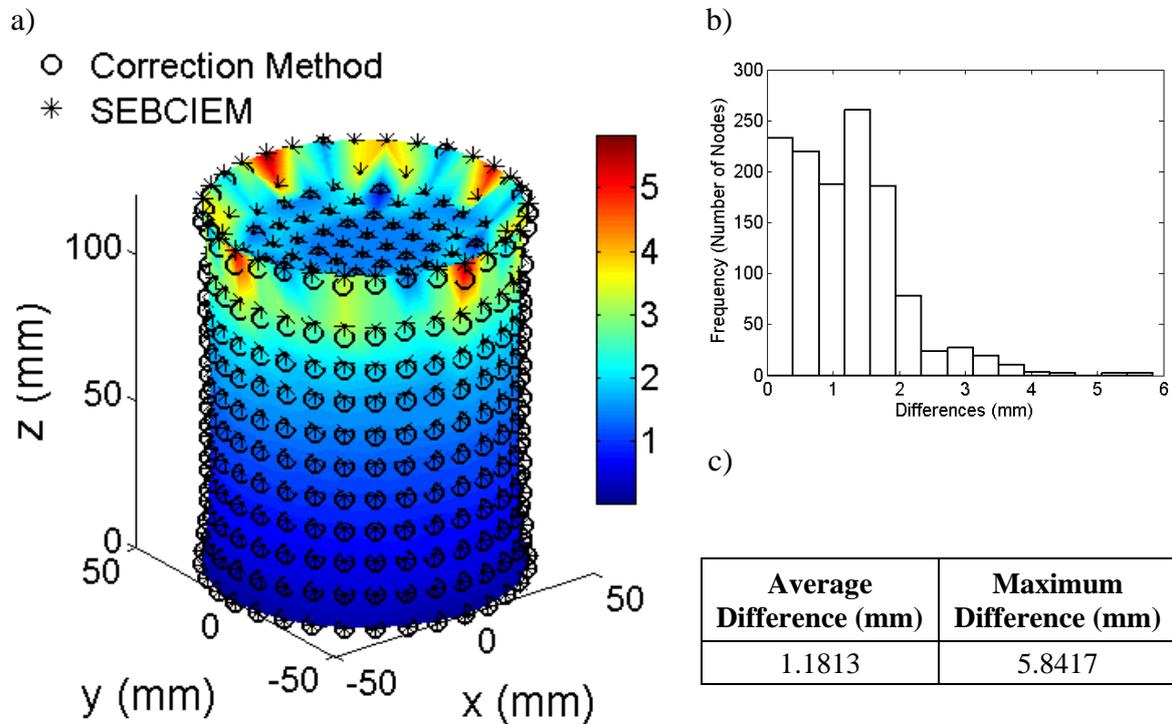


Figure 13: Extension of a Soft Tissue Sample in 3D: a) Differences of computed deformation field for the entire domain between SEBCIEM and Correction method; b) Histogram of differences; c) Average and maximum differences. All dimensions are in millimetres.

#### 4.4. Unconstrained compression of a cube

In this example we simulate the unconstrained compression of a cube with an edge length of 0.1 m. For the meshless simulations, the cube is discretised using 260 nodes and 4056 integration points are created using a tetrahedral background mesh with four integration points per tetrahedron. MMLS with quartic spline weight function is used to approximate the unknown field functions [28, 29]. A hyper-elastic Neo-Hookean material model with Young's modulus of 3000 Pa and Poisson's ratio of 0.49 is used and a maximum displacement of 0.02 m is applied to the top surface. The meshless discretization of the problem domain and the boundary conditions are shown in Figure 14.

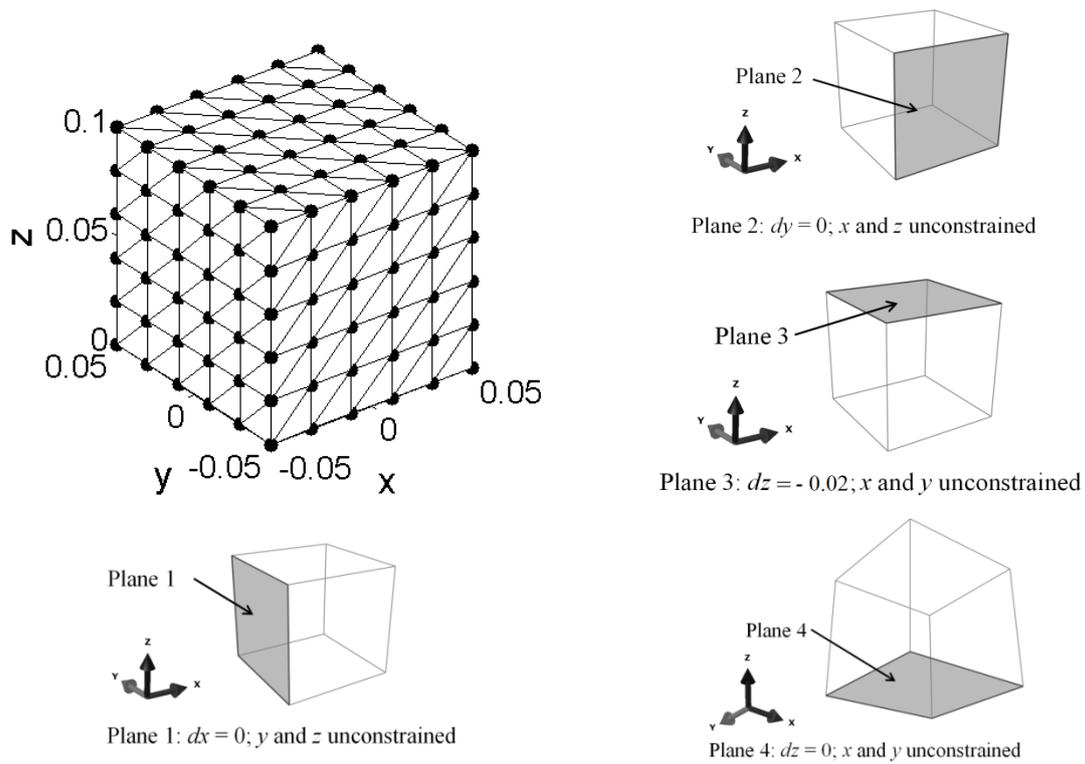


Figure 14: Unconstrained Compression of a Cube: The discretization (260 nodes, 1014 integration cells) and the boundary conditions.

We obtain meshless simulation results using both EBCIEM and SEBCIEM to impose EBC and compare them with the Finite Element (FE) solution obtained using the ABAQUS static solver and the same discretization. The chosen deformation mode and material model result in uniform strains in the cube, and the finite element solution obtained using ABAQUS should be exact and independent of discretisation. Like any FE approach, ABAQUS imposes the EBC exactly as prescribed due to the inherent interpolating property of the finite element shape functions.

Figure 15 shows the differences between the nodal displacements obtained using the meshless methods (EBCIEM and SEBCIEM) and the reference solution obtained using ABAQUS. Table 4 shows the differences of the nodal displacements obtained using EBCIEM, SEBCIEM and ABAQUS for the entire domain.

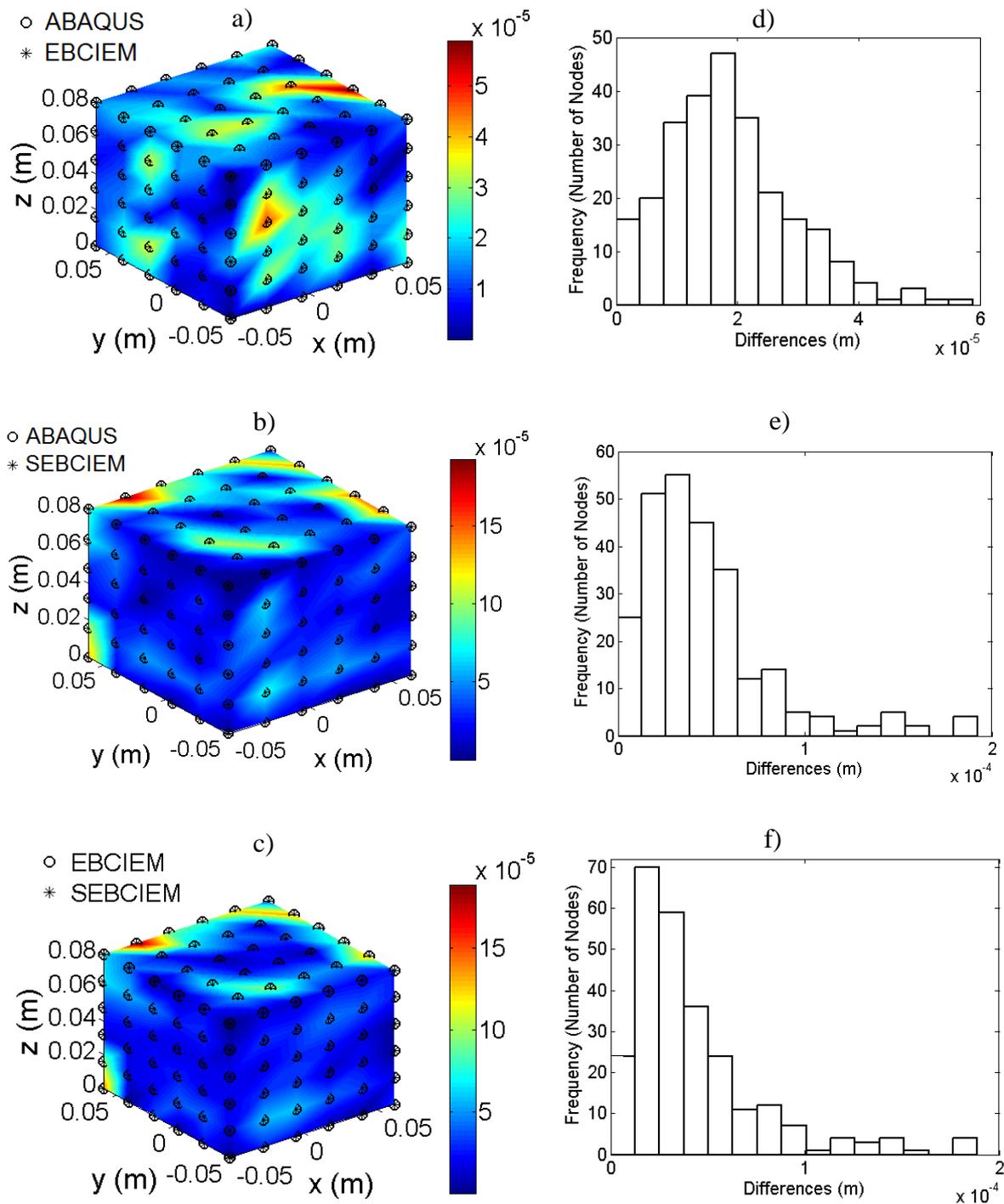


Figure 15: Unconstrained Compression of a Cube: Differences of computed deformation field between: a) EBCIEM and ABAQUS; b) SEBCIEM and ABAQUS, and c) EBCIEM and SEBCIEM. Histogram of differences between: d) EBCIEM and ABAQUS; e) SEBCIEM and ABAQUS; and f) EBCIEM and SEBCIEM.

**Table 4.** Unconstrained Compression of a Cube: Comparison of average and maximum differences of the nodal displacements obtained using EBCIEM, SEBCIEM and ABAQUS.

Methods to Compare	Average Difference (m)	Maximum Difference (m)
EBCIEM and ABAQUS	$1.8691 \times 10^{-5}$	$5.8853 \times 10^{-5}$
SEBCIEM and ABAQUS	$4.6187 \times 10^{-5}$	$1.9240 \times 10^{-4}$
EBCIEM and SEBCIEM	$4.2372 \times 10^{-5}$	$1.8890 \times 10^{-4}$

The results demonstrate that the differences between the meshless (EBCIEM and SEBCIEM) and FE solution are very small for the entire domain, demonstrating the correctness of our meshless simulations.

## 5. Conclusions

In this research, we have introduced two new ways of imposing essential boundary condition in meshless methods based on Element Free Galerkin principle. We have explained the motivations behind developing the new methods, called *Essential Boundary Conditions Imposition in Explicit Meshless* (EBCIEM) and *Simplified Essential Boundary Conditions Imposition in Explicit Meshless* (SEBCIEM), in the context of explicit time integration. We have derived the mathematical formulation for the two methods.

The new methods consider the imposed essential boundary conditions are the result of externally applied forces distributed over the essential boundary and we describe two approaches to evaluate the externally applied force on the essential boundary. In the first approach, EBCIEM, the values of the distributed force are interpolated at the essential boundary nodes. A discretization along the essential boundary is necessary in this case to numerically integrate the externally applied force on the essential boundary. In the second approach, SEBCIEM, the distributed force on the essential boundary are lumped at the essential boundary nodes. The advantage of SEBCIEM is that it does not require any discretization along the essential boundary to evaluate the externally applied force. In applications where discretization along the essential boundary is difficult to achieve, the simplified version of EBCIEM could be useful. Both methods obtain displacement corrections which are added to the displacement field during time stepping to impose the displacements on the essential boundary.

We established the applicability of the new methods in a meshless Total Lagrangian framework when using explicit time integration. However, the methods can be used in the Updated Lagrangian formulation, if the decreased computational performance is acceptable. We tested the proposed methods using four applications, including both 2D and 3D examples. Because of the way they are constructed, both EBCIEM and SEBCIEM impose exactly the essential boundary conditions to machine precision.

**Acknowledgments:** The second author is a recipient of the SIRF scholarship and acknowledges the financial support of the University of Western Australia. The financial support of Australian Research Council (Discovery Grant No. DP160100714) is gratefully acknowledged. We wish to acknowledge the Raine Medical Research Foundation for funding G. R. Joldes through a Raine Priming Grant.

#### **References:**

- [1] S. Li, W.K. Liu, Meshfree Particle Methods, Springer2004.
- [2] G.R. Liu, Meshfree methods: moving beyond the finite element method, CRC press2009.
- [3] T. Belytschko, Y.Y. Lu, L. Gu, Element-Free Galerkin Methods, Int J Numer Meth Eng, 37 (1994) 229-256.
- [4] T.W. Parks, C.S. Burrus, Digital Filter Design, John Wiley & Sons, Inc., New York, 1987.
- [5] P. Lancaster, K. Salkauskas, Surfaces Generated by Moving Least-Squares Methods, Math Comput, 37 (1981) 141-158.
- [6] S. Fernández-Méndez, A. Huerta, Imposing essential boundary conditions in mesh-free methods, Computer Methods in Applied Mechanics and Engineering, 193 (2004) 1257-1275.
- [7] G.-R. Liu, Y.-T. Gu, An introduction to meshfree methods and their programming, Springer2005.
- [8] T. Zhu, N.S. Atluri, A modified collocation method and a penalty formulation for enforcing the essential boundary conditions in the element free Galerkin method, Computational Mechanics, 21 (1998) 211-222.
- [9] T. Most, C. Bucher, A Moving Least Squares weighting function for the Element-free Galerkin Method which almost fulfills essential boundary conditions, Struct Eng Mech, 21 (2005) 315-332.

- [10] Y. Krongauz, T. Belytschko, Enforcement of essential boundary conditions in meshless approximations using finite elements, *Computer Methods in Applied Mechanics and Engineering*, 131 (1996) 133-145.
- [11] T. Belytschko, D. Organ, Y. Krongauz, A coupled finite element - Element-free Galerkin method, *Computational Mechanics*, 17 (1995) 186-195.
- [12] J. Berger, A. Horton, G. Joldes, A. Wittek, K. Miller, Coupling finite element and mesh-free methods for modelling brain deformations in response to tumour growth, *Computational Biomechanics for Medicine III MICCAI-Associated Workshop*, (2008).
- [13] G.Y. Zhang, A. Wittek, G.R. Joldes, X. Jin, K. Miller, A three-dimensional nonlinear meshfree algorithm for simulating mechanical responses of soft tissue, *Engineering Analysis with Boundary Elements*, (2013).
- [14] K.J. Bathe, *Finite Element Procedures*, Prentice Hall 1996.
- [15] T. Belytschko., W.K. Liu., B. Moran., *Nonlinear Finite Elements for Continua and Structures*, John Wiley & Sons Ltd. 2000.
- [16] K. Miller, *Biomechanics of the Brain*, Springer 2011.
- [17] K. Miller, G.R. Joldes, D. Lance, A. Wittek, Total Lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation, *Communications in Numerical Methods in Engineering*, 23 (2006) 121-134.
- [18] A. Wittek, P.M.F. Nielsen, K. Miller, *Computational Biomechanics for Medicine : Soft Tissues and the Musculoskeletal System*, Springer 2011.
- [19] A. Horton, A. Wittek, G.R. Joldes, K. Miller, A meshless Total Lagrangian explicit dynamics algorithm for surgical simulation, *International Journal for Numerical Methods in Biomedical Engineering*, 26 (2010) 977-998.
- [20] M. Li, K. Miller, G.R. Joldes, R. Kikinis, A. Wittek, Biomechanical model for computing deformations for whole-body image registration: A meshless approach, *International Journal for Numerical Methods in Biomedical Engineering*, (2016) n/a-n/a.
- [21] K. Miller, A. Horton, G.R. Joldes, A. Wittek, Beyond finite elements: a comprehensive, patient-specific neurosurgical simulation utilizing a meshless method, *Journal of biomechanics*, 45 (2012) 2698-2701.
- [22] G.R. Joldes, A. Wittek, K. Miller, Stable time step estimates for mesh-free particle methods, *Int J Numer Meth Eng*, 91 (2012) 450-456.
- [23] G.R. Joldes, A. Wittek, K. Miller, Computation of intra-operative brain shift using dynamic relaxation, *Computer Methods in Applied Mechanics and Engineering*, 198 (2009) 3313-3320.
- [24] M.D. Greenberg, *Advanced Engineering Mathematics*, Prentice Hall 1998.

- [25] G.R. Joldes, A. Wittek, K. Miller, An adaptive Dynamic Relaxation method for solving nonlinear finite element problems. Application to brain shift estimation., *Int. J. Numer. Method. Biomed. Eng.*, 27 (2011) 173-185.
- [26] G.R. Joldes, A. Wittek, K. Miller, Adaptive Numerical Integration in Element-Free Galerkin Methods for Elliptic Boundary Value Problems., *Eng. Anal. Bound. Elem.*, 51 (2015) 52-63.
- [27] G.R. Joldes, A. Wittek, K. Miller, Suite of finite element algorithms for accurate computation of soft tissue deformation for surgical simulation, *Med. Image Anal.*, 13 (2009) 912-919.
- [28] H. Chowdhury, G. Joldes, A. Wittek, B. Doyle, E. Pasternak, K. Miller, Implementation of a Modified Moving Least Squares Approximation for Predicting Soft Tissue Deformation Using a Meshless Method, in: B. Doyle, K. Miller, A. Wittek, P.M.F. Nielsen (Eds.) *Computational Biomechanics for Medicine*, Springer International Publishing, 2015, pp. 59-71.
- [29] G.R. Joldes, H.A. Chowdhury, A. Wittek, B. Doyle, K. Miller, Modified moving least squares with polynomial bases for scattered data approximation, *Applied Mathematics and Computation*, 266 (2015) 893-902.