An Element Free Galerkin Method for the Prediction of Soft Tissue Deformation in Surgical Simulation

Habibullah Amin Chowdhury
B.Sc.Eng, M.Eng

This thesis is presented for the degree of Doctor of Philosophy of
The University of Western Australia

Intelligent Systems for Medicine Laboratory
School of Mechanical & Chemical Engineering

2017
I, Habibullah Amin Chowdhury, certify that:

This thesis has been substantially accomplished during enrolment in the degree.

This thesis does not contain material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution.

No part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of The University of Western Australia and where applicable, any partner institution responsible for the joint-award of this degree.

This thesis does not contain any material previously published or written by another person, except where due reference has been made in the text.

The work(s) are not in any way a violation or infringement of any copyright, trademark, patent, or other rights whatsoever of any person.

This thesis contains published work and/or work prepared for publication, some of which has been co-authored.

Signature: [Signature Section]

Date: 20/03/2017
ABSTRACT

In applications such as neurosurgical simulation or neuroimage registration, some of the key objectives of computational biomechanics are to enable a surgeon to simulate surgery within the operating theatre using cost-effective, readily-available computing facilities and to visualize the results immediately with high accuracy. For this purpose, the creation of an easy to manipulate computational grid, as well as a robust, accurate method for solving the fundamental equations describing the biomechanical behaviour of the subjects are some of the essential requirements.

For the last four decades, Finite Element Analysis (FEA) has been the method of choice in computational biomechanics. However, even in the most efficient finite element framework there are limitations. Despite the undeniable progress achieved using FEA, its limitations have been highlighted in the literature in the context of patient-specific applications, where compatibility with a clinical workflow is essential, and include: a) time-consuming generation of patient-specific computational grids (finite element meshes) from medical images — a process that requires image segmentation, creation of water-tight surfaces from the segmentation and discretization of the complex geometries of body organs defined by these surfaces into interconnected meshes of high-quality elements; and b) deterioration of the solution accuracy when elements undergo distortion under large deformations induced by interactions between the tissues and surgical tools.

In this context, Meshless Methods (MMs) have been suggested as possible alternatives to FEA. The complex finite element grid generation and element distortion problem are eliminated, as no mesh is required; only a set of nodes scattered within the domain and on the boundary of the domain is needed to discretize the problem space. Among the many types of MMs, the Element Free Galerkin (EFG) method is applied in biomechanics applications and increasingly becoming one of the most commonly used meshless methods in this field. EFG was initially designed and commonly used for crack propagation and has been applied to diverse linear and non-linear problems. Despite its widespread use, meshless methods based on EFG formulation exhibit three major limitations: a) meshless shape functions using higher order basis cannot always be computed for arbitrarily distributed nodes (irregular node placement is crucial for facilitating automated discretization of complex geometries); b) challenges in imposing the Essential Boundary Conditions (EBC), and, c) inaccurate numerical integration in space, as meshless shape functions are not polynomials. The main purpose of this thesis
is to address all these crucial drawbacks of the EFG based MMs discussed above and provide effective solutions for these problems.

Due to its continuity and smoothness properties, the Moving Least Squares (MLS) approximation has been the preferred choice in EFG. However, in order to compute the shape functions, the classical MLS places strict requirements on the nodal distributions inside each support domain. Therefore, the practical use of higher order polynomial basis, which can generate more accurate approximations of complex deformation fields, is not trivial in classical MLS for randomly distributed nodes. In this context, this thesis presents a Modified Moving Least Squares (MMLS) method for interpolating scattered data both for visualization and for performing numerical computations using MMs. The error functional used in the derivation of the classical MLS approximation is augmented with additional terms based on the coefficients of the polynomial base functions. This allows quadratic polynomial base functions to be used with the same size of the support domain as linear base functions, resulting in better approximation capability. The thesis also demonstrates that the MMLS possess the necessary properties which allow its use as an Element Free Galerkin (EFG) meshless approximation method. The properties such as acceptable nodal distribution, continuity, consistency and invariance for the MMLS are proven. The increased robustness of the MMLS method to irregular nodal distributions makes it suitable for use in EFG based MMs. The interpolation capabilities of MMLS are assessed by several univariate and bivariate examples.

In this thesis, the MMLS approximation is implemented in the EFG based Meshless Total Lagrangian Explicit Dynamics (MTLED) algorithm. The performance of the EFG method based on MMLS, which uses quadratic base functions, is compared to that of the EFG method, which uses classical MLS with linear base functions, using both 2D and 3D examples. The simulation results demonstrate the superior performance of the MMLS over classical MLS in terms of solution accuracy, while shape functions can be computed using the same nodal distribution and support domain size for both methods.

The next limitation of MMs addressed in this thesis is the exact imposition of EBC, which is crucial for accurate prediction of organ deformations due to surgery. For some classes of problems, driving deformation through EBC imposition can achieve accurate solutions without patient-specific information about tissue constitutive properties; therefore, imposing EBC is crucial for patient-specific applications. However, the MLS
shape functions in MMs are generally non-polynomial and non-interpolating rational functions. Thus, imposing EBC in MMs is not as trivial as in the FEA, as the shape functions do not possess the Kronecker delta property. Furthermore, most methods proposed for imposing EBC in MMs are not applicable to explicit time integration, which enables real-time computations for surgery simulation on commodity hardware (off-the-shelf Graphics Processing Units). For this purpose, this thesis presents a novel way of imposing EBC in EFG based MMs suitable for explicit time integration, named Essential Boundary Conditions Imposition for Explicit Meshless (EBCIEM). The effectiveness of the proposed method is demonstrated using both 2D and 3D numerical examples.

The other major limitation of MMs addressed by this thesis is the numerical integration in space where Gaussian quadrature over a background mesh is the preferred method. However, the non-polynomial nature of MLS shape functions also affects the accuracy of the numerical integration in EFG methods. Furthermore, the shape functions usually have a much larger support domain which may not align with the integration cells. Therefore, there are sources of errors associated with the integration schemes used in EFG based MMs. In this context, the thesis presents an implementation of an adaptive integration procedure in the MTLED algorithm. The new algorithm creates a distribution of integration points within the problem domain and allows the computation of integrals with controlled accuracy. The method introduces new integration points only in the areas where the integration accuracy is not sufficient. The method imposes no constraints on the type of support domains that can be used. Numerical examples in 2D and 3D are presented to demonstrate this new adaptive numerical integration procedure.
# TABLE OF CONTENTS

**ABSTRACT** .......................................................................................................................... iii

**Chapter 1: Introduction** .......................................................................................................... 1

1.1 Background and Motivation .............................................................................................. 2

1.1.1 Meshless Methods and Surgical Simulation: A Brief Review .................................. 4

1.2 Element Free Galerkin Method ....................................................................................... 7

1.3 Scope and Significance of This Study ............................................................................. 8

1.4 Thesis Overview ............................................................................................................. 10

**Chapter 2: Modified Moving Least Squares Approximation** ............................................. 13

2.1 Introduction .................................................................................................................... 15

2.2 Classical Moving Least Squares Approximation .......................................................... 16

2.3 Modified Moving Least Squares Approximation .......................................................... 19

2.4 Properties of the Modified Moving Least Squares Approximation ............................. 21

2.4.1 Acceptable nodal distribution ............................................................................... 22

2.4.2 Continuity .............................................................................................................. 24

2.4.3 Consistency ........................................................................................................... 24

2.4.4 Invariance ............................................................................................................. 26

2.5 Numerical Examples .................................................................................................... 31

2.5.1 MLS and MMLS shape functions in 1D ............................................................... 32

2.5.2 Approximation capability in 1D ............................................................................ 33

2.5.3 Approximation capability in 2D ............................................................................ 34

2.6 Concluding Remarks .................................................................................................... 39

**Chapter 3: An Element Free Galerkin Method Based on the Modified Moving Least Squares Approximation** ................................................................................................................... 40

3.1 Introduction .................................................................................................................... 42

3.2 Meshless Total Lagrangian Explicit Dynamics Algorithm ........................................... 43

3.2.1 Total Lagrangian Explicit Dynamics .................................................................... 43

3.2.2 Dynamic Relaxation .............................................................................................. 44

3.2.3 Material formulation ............................................................................................. 46

3.3 Modified Moving Least Squares in 3D ......................................................................... 47

3.4 Numerical Examples .................................................................................................... 49

3.4.1 Extension of a Soft Tissue Sample in 2D .............................................................. 50

3.4.2 Craniotomy Induced Brain Shift in 2D .................................................................. 52

3.4.3 Unconstrained compression of a Cube .................................................................. 55

3.5 Concluding Remarks .................................................................................................... 58
ACKNOWLEDGEMENTS

"what we cannot say, we must pass over in silence”... – Ludwig Wittgenstein
“what cannot be said above all must not be silenced but written!”... – Jacques Derrida

I would like to express my sincere gratitude to my supervisors – W/Prof. Karol Miller, Assoc/Prof. Grand Joldes, Prof. Adam Wittek and Prof. Elena Pasternak.

I earnestly thank Grand for providing support and scope of my research, and also, for being tolerant when I was lagging behind, even when we were Rogaining.

I sincerely thank Karol for his ever inspiring erudite advices and suggestions regarding every aspect of mortal life and scientific endeavours.

I am grateful to Adam for his insightful comments and encouragement, and reminding me every step of way if ‘the paper is ready?’, or ‘ahh everything is wrong!’.

I am thankful to Elena for the suggestions, motivations and conversations.

I acknowledge the financial support of the University of Western Australia for providing the Scholarship for International Research Fees (SIRF). The financial support of Australian Research Council (Discovery Grant No. DP120100402 & DP160100714) is gratefully acknowledged.
AUTHORSHIP DECLARATION: CO-AUTHORED PUBLICATIONS

This thesis contains work that has been published and prepared for publication.

Details of the work:


Location in thesis:
Chapter 2

Student contribution to work:
60%

Co-author signatures and dates:

G.R. Joldes, 27/03/2017
A. Wittek, 23/03/2017
B. Doyle, 24/03/2017
E. Pasternak, 27/03/2017
K. Miller, 01/11/2017

Details of the work:


Location in thesis:
Chapter 2

Student contribution to work:
40%

Co-author signatures and dates:

G.R. Joldes, 27/03/2017
A. Wittek, 23/03/2017
B. Doyle, 24/03/2017
K. Miller, 01/11/2017
<table>
<thead>
<tr>
<th>Details of the work:</th>
</tr>
</thead>
</table>

**Location in thesis:**
Chapter 2

**Student contribution to work:**
60%

**Co-author signatures and dates:**
- G.R. Joldes, 27/03/2017
- A. Wittek, 23/03/2017
- B. Doyle, 24/03/2017
- E. Pasternak, 27/03/2017
- K. Miller, 01/11/2017

<table>
<thead>
<tr>
<th>Details of the work:</th>
</tr>
</thead>
</table>

**Location in thesis:**
Chapter 3

**Student contribution to work:**
60%

**Co-author signatures and dates:**
- A. Wittek, 23/03/2017
- K. Miller, 01/11/2017
- G.R. Joldes, 27/03/2017

<table>
<thead>
<tr>
<th>Details of the work:</th>
</tr>
</thead>
</table>

**Location in thesis:**
Chapter 4

**Student contribution to work:**
60%

**Co-author signatures and dates:**
- G.R. Joldes, 27/03/2017
- A. Wittek, 23/03/2017
- K. Miller, 01/11/2017
<table>
<thead>
<tr>
<th>Details of the work:</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Location in thesis:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Student contribution to work:</th>
</tr>
</thead>
<tbody>
<tr>
<td>60%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Co-author signatures and dates:</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.R. Joldes, 27/03/2017</td>
</tr>
<tr>
<td>A. Wittek, 23/03/2017</td>
</tr>
<tr>
<td>K. Miller, 01/11/2017</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Student signature:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date: 20/03/2017</td>
</tr>
</tbody>
</table>

I, W/Prof. Karol Miller, certify that the student statements regarding their contribution to each of the works listed above are correct.

<table>
<thead>
<tr>
<th>Coordinating supervisor signature:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date: 01/11/2017</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction
1.1 Background and Motivation

In applications such as neurosurgical simulation or neuroimage registration, some of the key objectives of computational biomechanics are to enable a surgeon to simulate surgery within the operating theatre, using cost-effective and readily-available computing facilities and to visualize the results immediately with high accuracy. In this way, a surgeon, without requiring any professional knowledge in numerical computation, would be able to evaluate the implications of each stage of a surgical procedure and explore potential alternative solutions. For this purpose, the creation of an easily-manipulated computational grid, as well as a robust, accurate method for solving the fundamental equations describing the biomechanical behaviour of the subject are the essential requirements.

For the last four decades, Finite Element Analysis (FEA) has been the method of choice in computational biomechanics. Nevertheless, the conventional approach to compute soft tissue deformation was depended on linear finite element algorithms that assume infinitesimal deformations (Cotin et al., 2000, Warfield et al., 2002). However, modeling of soft tissue organs for applications such as neurosurgical simulation and neuroimage registration for image-guided surgery is a non-linear problem of continuum mechanics that involves large deformations and large strains with geometric and material non-linearities (Miller, 2011). Therefore the infinitesimal assumption is not satisfied.

In this context, Miller et al. (2006) have developed an efficient finite element algorithm using total Lagrangian (TL) formulation and explicit time integration for computing soft tissue deformation. The algorithm is capable of handling both geometric and material non-linearities. The adoption of 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 iteration during each time-step (Bathe, 1996). Several applications have already been demonstrated in surgical simulation and image registration based on this framework (Wittek et al.,
This is to note that for image-guided surgery, accurate warping of high quality pre-operative Magnetic Resonance (MR) images to the deformed intra-operative organ configuration is the process known as non-rigid registration. In this case, only the final state of soft-tissue organ deformation during surgery needs to be predicted which requires algorithms for determining steady-state solution (Wittek et al., 2011, Miller et al., 2011). Although the neuroimage registration is a non-linear problem as it involves large deformations, non-linear material properties and non-linear boundary conditions, it is a less demanding problem than neurosurgery simulation. In neurosurgery simulation, the time history of forces and deformations has to be obtained, whereas in neuroimage registration, only the steady state solution for deformations is of interest (Miller, 2011). In the finite element framework, adopting the TL formulation, explicit time integration and dynamic relaxation (Joldes et al., 2011) to reach steady state solution provided an efficient numerical method for solving soft tissue deformation with geometric and material non-linearities.

However, even in the most efficient finite element framework there are limitations. Despite the undeniable progress achieved using FEA, its limitations have been highlighted in the literature in the context of patient-specific applications, where compatibility with a clinical workflow is essential, including: a) time-consuming generation of patient-specific computational grids (finite element meshes) from medical images — a process that requires image segmentation, creation of water-tight surfaces from the segmentation and discretization of the complex geometries of body organs defined by these surfaces into interconnected meshes of high-quality elements; and b) deterioration of the solution accuracy when elements undergo distortion under large deformations induced by interactions between the tissues and surgical tools.

In this context, Meshless Methods (MMs) (Li and Liu, 2004, Liu, 2010) have been suggested as possible alternatives to FEA. The complex finite element grid generation and element distortion problem are eliminated, as no mesh is required; only a set of nodes scattered within the domain and on the boundary of the domain is needed to discretize the problem space. Placement of
these nodes can be done automatically since their arrangement is almost arbitrary. Several types of MMs have been developed over the last few decades. An elaborate review of different types of MMs can be found in (Chen et al., 2006, Fasshauer, 2007, Li and Liu, 2004, Liu, 2010). In the next subsection, some of the existing uses of MMs in the context of surgical simulation are briefly discussed.

1.1.1 Meshless Methods and Surgical Simulation: A Brief Review

The first ‘mesh-less’ method dates back from 1977 when Gingold and Monaghan (1977) and Lucy (1977) developed a Lagrangian method called Smoothened Particle Hydrodynamics (SPH) to model astrophysics problems. They treated the discrete astrophysics problems as continuous problems and their method discretized the fluid by a set of moving particles (Liu, 2010). Although SPH was successful in modelling astrophysics problems back in 70’s, it was only recently that SPH has been applied to other classes of problems including soft tissue deformation for surgical simulation. A combined particle/continuum approach in modelling soft tissue damage and failure has been demonstrated by Rausch et al. (2016) using SPH. The authors implemented the anisotropic hyperelastic constitutive model, which is commonly employed for modeling soft tissues within the SPH framework. They also used a normalized kernel function which helped in resolving the unnatural phenomenon often produced by the first derivative of the traditional spline kernel functions used in SPH. However, lack of interpolation consistency and difficulty in imposing essential boundary conditions are still some open issues in SPH.

The Reproducing Kernel Particle (RKP) method (Liu et al., 1995) was used by Smolen and Patriciu (2009) for predicting soft tissue deformation. The authors showed that their model with coarse particle grid can produce good results in deforming soft tissues for robotic manipulators. An Adaptive Reproducing Kernel method was used by Li and Lee (2007) to predict large deformation
in soft tissues during needle insertion. In this case, numerical integration was performed across the Voronoi cells on each node.

The Finite Spheres (FS) method was used by De et al. (2003) to simulate the palpation of soft tissues through a tool tip. The work includes simulating 2D and 3D geometries with linear geometrical and material formulations. De et al. have also presented a specialized version of FS method using point collocation for fast computation of surgery simulation in a virtual environment. In this case, numerical integration is avoided as the governing differential equations are directly applied to the nodal points. The authors used Moving Least Squares (MLS) with singular weight functions as interpolation functions. Further developments of the method are produced by De et al. (2005) and Lim et al. (2007) for three dimensional cases with geometrical nonlinearity. The point collocation based finite sphere method is used by Lim and De (2004) to simulate the cutting of realistic organ models.

Natural Element (NE) method (Sukumar et al., 1998, Traversoni, 1994, Braun and Sambridge, 1995), or Natural Neighbour Galerkin method is used by Doblare et al. (2005) to simulate hyperelastic modelling of ligaments. The NE method is interpolant by nature and well suited to simulate piecewise homogeneous domains with high accuracy. Furthermore, essential boundary conditions can be imposed easily. The NE method can handle large deformation without any special numerical treatment.

Material Point (MP) method is used by Ionescu et al. (2006) to simulate soft tissue failure. The authors implemented a hyperelastic constitutive model with a strain-based failure criterion using the MP method. Simulations of penetrating trauma to soft tissue samples were performed. The main advantages of the MP framework for failure simulations are the easy representation of complicated geometries, the ability to implement arbitrary constitutive models and the ability to avoid mesh entanglement and inversion problems. The main idea of MP method is to trace the motions of a set of material points, which carry the information of all the state variables in a
Lagrangian manner. On the other hand, spatial discretization and displacement interpolation is made with respect to spatial coordinate detached from the material body as an Eulerian description (Li and Liu, 2004).

The Multiquadric (MQ) basis proposed by Hardy (1990) became one of the widely used methods in surface fitting and in approximating solutions for partial differential equations (Kansa, 1990). The use of radial basis functions can avoid singular moment matrices during the shape function construction compared to using polynomial basis functions (Liu, 2010). The Multiquadric Radial Basis Functions is used by Hon et al. (2002) to simulate a two-dimensional triphasic model of charged and hydrated soft tissue. The authors implemented nonlinear material properties with incompressibility, and described the prospect of the method to solve for complicated geometries. Classical domain decomposition technique is also combined successfully with the proposed method for solving large scale problems.

The Element Free Galerkin (EFG) method is applied in biomechanics applications (Jin et al., 2012, Li et al., 2016, Liu and Shi, 2003) and continuously becoming one of the most commonly used meshless methods in this field. EFG was initially designed and commonly used for crack propagation (Belytschko et al., 1995a). Over the years, EFG methods have gained popularity and have been applied to diverse problems such as elasticity (Dolbow and Belytschko, 1998, Lu et al., 1994, Bahmyari and Khedmati, 2017), linear and non-linear problems in 3D (Barry and Saigal, 1999, Belytschko et al., 1997, Zhang et al., 2016), static and dynamic fractures (Belytschko et al., 1995b, Khazal et al., 2016), vibration (Chen et al., 2003, Liu et al., 2002, Wu et al., 2016, Bahmyari et al., 2013), electromagnetics (Clingoski et al., 1998, Yu and Jia, 2008, Yang et al., 2011), electromechanical coupled analysis (Li et al., 2017b), heat transfer (Singh et al., 2007, Yang and Liu, 2005, Li et al., 2017a, Zhang et al., 2017), structural problems (Joldes et al., 2017) etc. Despite its widespread use, some aspects of the EFG method still require further research. In the next sections, the evolution and some of the major problems associated with the EFG method will be discussed.
1.2 Element Free Galerkin Method

The origin of EFG method was motivated by the Diffuse Element Method (DEM), which is the first ‘mesh-less’ method based on the Galerkin technique, introduced by Nayroles and Touzot (1992). The idea of DEM was to replace the finite element interpolation technique within an element by the Moving Least Squares (MLS) approximation. The MLS method was first introduced by Shepard (1968) to construct smooth approximations for fitting a cloud of points (Nguyen et al., 2008). In 1981, Lancaster and Salkauskas (1981) extended this method for general surface generation problems. Belytschko (1994) introduced an extended version of the DEM and named it the ‘Element Free Galerkin’ method. In EFG method, the problem domain discretisation is achieved using nodes arbitrarily distributed within and on the boundary of the problem domain, a Galerkin weak form is employed to develop the discretised system equations and background cells are used for numerical integration. The EFG method introduced some improvements over the DEM formulation. In EFG, the full form of the derivatives of the approximation function is used, whereas in DEM, the derivatives of the approximation function are obtained by considering the coefficients of the polynomial basis as constants. According to Belytschko (1994), EFG methods produce more accurate results compared with DEM by not neglecting the derivatives of the coefficients.

The MLS approximation procedure is used in EFG method to approximate the displacement field at a point of interest using the nodal parameters of the nodes that fall into the support domain of that point. The MLS shape functions used in the EFG method are ‘mesh-less’ as they are constructed entirely based on the nodes. In FEM, the supports of the shape functions are also used as integration cells. However, in EFG based MMs the background integration cells are constructed independently of the shape functions. Although the simplest way to define the background integration cells is by creating a mesh, such mesh does not need to be defined based on the discretising nodes, and each integration cell can be handled independently during integration without having to ensure mesh compatibility.
The EFG method is used for developing a suite of nonlinear meshless algorithms in my research group Intelligent Systems for Medicine Laboratory (ISML) for computing large deformation in soft tissues, known as the Meshless Total Lagrangian Explicit Dynamics (MTLED). The algorithm is motivated by the earlier work of Miller et al. (2006) and developed by Horton et al. (2010) and Joldes et al. (2009a, 2012). The MTLED is used in this thesis for computing soft tissue deformation. A more detailed discussion of the algorithm is available in Section 3.2.

1.3 Scope and Significance of This Study

As stated in the previous section, meshless methods using EFG formulation, similar to MTLED algorithm, have been successfully used in predicting soft tissue deformation during surgery. However, in their current formulation, they require an expert in computational mechanics to construct computational grids and perform simulations making these algorithms incompatible with a clinical workflow. This is because of three major limitations exhibited by meshless methods using EFG formulations, such as the MTLED algorithm:

a) Meshless shape functions using higher order basis cannot always be computed for arbitrarily distributed nodes (irregular node placement is crucial for facilitating automated discretization of complex geometries).

b) Challenges in imposing the Essential Boundary Conditions (EBC).

c) Inaccurate numerical integration in space, as meshless shape functions are not polynomials.

Due to its continuity and smoothness properties, the Moving Least Squares (MLS) (Shepard, 1968, Lancaster and Salkauskas, 1981) approximation has been the preferred choice in EFG. However, in order to compute the shape functions, the classical MLS places strict requirements on the nodal distributions inside each support domain. Therefore, the practical use of higher order
polynomial basis, which can generate more accurate approximations of complex deformation fields, is not trivial in classical MLS for randomly distributed nodes.

The next limitation is the exact imposition of EBC which is crucial for accurate prediction of organ deformations due to surgery. For some classes of problems, driving deformation through EBC imposition can achieve accurate solutions without patient-specific information about tissue constitutive properties (Wittek et al., 2009); therefore, imposing EBC is crucial for patient-specific applications. However, the MLS derived shape functions in MMs are generally non-polynomial and non-interpolating rational functions. Thus, imposing EBC in MMs is not as trivial as in the FEA, as the shape functions do not possess the Kronecker delta property (Liu, 2010). Furthermore, most methods proposed for imposing EBC in MMs are not applicable to explicit time integration, which enables real-time computations for surgery simulation on commodity hardware (off-the-shelf Graphics Processing Units) (Joldes et al., 2010).

The other major limitation is the numerical integration in space in the Galerkin-type MMs where Gaussian quadrature over a background mesh is the preferred method. However, the non-polynomial nature of MLS shape functions also affects the accuracy of the numerical integration in EFG methods. Furthermore, the shape functions usually have a much larger support domain which may not align with the integration cells. Therefore, there are sources of errors associated with the integration schemes used in EFG based MMs.

This thesis addresses all of the above three limitations of explicit EFG method and finds effective solutions for the problems mentioned. The thesis presents a Modified Moving Least Squares (MMLS) method for interpolating scattered data both for visualization and for performing numerical computations, a novel way of imposing EBC for explicit time integration, named Essential Boundary Conditions Imposition for Explicit Meshless (EBCIEM), and implements an adaptive numerical integration procedure within the Meshless Total Lagrangian Explicit Dynamics algorithm. An overview of the thesis is presented in the next section.
1.4 Thesis Overview

The thesis is composed of six chapters. Chapters (2-4) contain already published articles.

• Chapter 1 – Introduction:
  o This chapter presents a general introduction and discussions about the motivation of using meshless method in the context of surgical simulation. The scope and significance of the thesis is also discussed.

• Chapter 2 - Modified Moving Least Squares Approximation:
  o This chapter presents a Modified Moving Least Squares (MMLS) method for interpolating scattered data both for visualization and for performing numerical computations using Meshless Method. The error functional used in the derivation of the classical MLS approximation is augmented with additional terms based on the coefficients of the polynomial base functions. This allows quadratic polynomial base functions to be used with the same size of the support domain as linear base functions, resulting in better approximation capability. The chapter also demonstrates that the MMLS possess the necessary properties which allow its use as an Element Free Galerkin (EFG) meshless approximation method. The properties such as acceptable nodal distribution, continuity, consistency and invariance for the MMLS are proven. The chapter demonstrates that MMLS shape functions form a partition of unity and the MMLS approximation satisfies the patch test. The invariance properties are important for the accurate computation of the shape functions by using translation and scaling to a canonical domain. The increased robustness of the MMLS method to irregular nodal distributions makes it suitable for using it EFG based MMs, such as the Meshless Total Lagrangian Explicit Dynamics algorithm. The interpolation capabilities of MMLS are assessed by several univariate and bivariate examples.
• Chapter 3 - *An Element Free Galerkin Method Based on the Modified Moving Least Squares Approximation:*

  o This chapter implements the MMLS approximation in the EFG based MTLED algorithm. An overview of the MTLED algorithm is presented. The performance of the EFG method based on MMLS, which uses quadratic base functions, is compared to that of the EFG method which uses classical MLS with linear base functions, using both 2D and 3D examples. Two examples are solved in 2D: an extension of a soft tissue sample and a craniotomy induced brain shift. An elasticity problem is solved in 3D (unconstrained compression of a cube) which has an exact finite element solution. The simulation results demonstrate the superior performance of the MMLS over classical MLS in terms of solution accuracy, while shape functions can be computed using the same nodal distribution and support domain size for both methods.

• Chapter 4 - *Essential Boundary Conditions Imposition for Explicit Meshless Method:*

  o This chapter presents a new technique in applying Essential Boundary Conditions (EBC) in Meshless Methods (MM) based on Element Free Galerkin (EFG) principles. Imposing EBC is a fundamental issue in MMs. The imposition of prescribed displacement values on the boundary in MM 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. A novel way of imposing EBC in EFG based MM suitable for explicit time integration, named Essential Boundary Conditions Imposition for Explicit Meshless (EBCIEM) is presented. The effectiveness of the proposed method is demonstrated using both 2D and 3D numerical examples.
• Chapter 5 - Adaptive Numerical Integration in Meshless Total Lagrangian Explicit Dynamics:
  o This chapter implements the new adaptive integration procedure (Joldes et al., 2015b) in the Meshless Total Lagrangian Explicit dynamics algorithm. The new algorithm creates a distribution of integration points within the problem domain and allows the computation of integrals with controlled accuracy. The method introduces new integration points only in the areas where the integration accuracy is not sufficient. The method imposes no constraints on the type of support domains that can be used. Numerical examples include a soft tissue extension in 2D and unconstrained compression of a cube and a swine brain indentation in 3D.

• Chapter 6 - Conclusions and Discussions:
  o This chapter presents the conclusions and recommendations for future work.
Chapter 2

Modified Moving Least Squares Approximation
Most of the contents of this chapter have been published on:


Additionally, the results of this chapter led to:

2.1 Introduction

In surgical simulation, the accurate prediction of soft tissue deformation is a fundamental prerequisite. A reliable method for interpolating scattered data is required both for visualization and for performing numerical computations, such as when using Meshless Method (MM) (Li and Liu, 2004, Liu, 2003, Jin et al., 2014, Zhang et al., 2014, Miller et al., 2012, Liu, 2010). MMs have evolved over the years as an alternative approach which addresses some of the shortcomings of the Finite Element Method (FEM), as discussed in Chapter 1. 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.

Meshless shape functions are necessary in interpolating scattered data for visualization and approximating the unknown field functions to find the approximate solution to a problem governed by PDEs and boundary conditions using arbitrarily distributed field nodes (Liu, 2010). The Moving Least Squares (MLS) shape functions have been preferred predominantly in MMs due to the smoothness, continuity and consistency of the approximation field they create (Nayroles et al., 1992, Belytschko et al., 1994, Liu, 2010).

The MLS method was first introduced by Shepard (1968) to construct smooth approximations for fitting a cloud of points (Nguyen et al., 2008). In 1981, Lancaster and Salkauskas (1981) extended this method for general surface generation problems. In generating meshless shape functions, higher order polynomial basis functions are useful in approximating complex data distributions. They also have the potential to increase the accuracy of the simulation results compared with linear basis function. Nevertheless, as the degree of the polynomial basis function increases, it becomes more difficult to ensure the independence of the shape functions, requiring increased size of the support domains so that more data points are included in the
influence domain for each evaluation point. Furthermore, not all node distributions can be used in numerical computations, as for some distributions, the shape functions cannot be computed over the entire problem domain. A valid node distribution is referred to as an “admissible node distribution” (Li and Liu, 2004). The number of admissible node distributions can be increased by increasing the size of the support domains (the dilatation parameters), but this leads to an increased number of data points inside influence domains, an increased number of shape functions covering a local area, more linearly-dependent shape functions in the local area and increased computational cost.

This chapter presents a modified MLS (MMLS) approximation which allows higher order polynomial base functions to be used under the same conditions as lower degree base functions. This is achieved by augmenting the error functional used in the derivation of the MLS shape functions with additional terms based on the coefficients of the polynomial base functions, therefore introducing additional constraints.

This chapter is organized as follows: the classical MLS approximation is presented in the next Section; the modified MLS approximation with second order polynomial bases is introduced in Section 2.3, some properties of the proposed approximation are discussed in Section 2.4, several univariate and bivariate examples are presented in Section 2.5, followed by discussion and conclusions in Section 2.6.

2.2 Classical Moving Least Squares Approximation

The procedure for constructing classical MLS shape function is presented here using the notations and derivation procedure from (Liu, 2003). The derivation starts with the approximation of a function \( u(x) \), denoted by \( u^h(x) \), which is defined by a combination of \( m \) monomials (also called basis functions):
\[ u^h(x) = \sum_{i=1}^{m} p_i(x) a_i(x) = \mathbf{p}^T(x) \mathbf{a}(x) \]  

(2.1)

where \( m \) is the number of terms in the basis \( \mathbf{p}(x) \), and \( a_i(x) \) are coefficients that depend on the spatial coordinates \( x \). For example, commonly used bases and the corresponding coefficients in 2D are:

- linear bases:

\[ \mathbf{p}^T(x) = [1, x, y], \quad \mathbf{a}^T(x) = [a_1, a_x, a_y] \]  

(2.2)

- quadratic bases:

\[ \mathbf{p}^T(x) = [1, x, y, x^2, xy, y^2], \quad \mathbf{a}^T(x) = [a_1, a_x, a_y, a_{x^2}, a_{xy}, a_{y^2}] \]  

(2.3)

These coefficients are computed by minimizing an error functional defined based on the weighted least squares errors:

\[ J(x) = \sum_{j=1}^{n} \left[ (u^h(x_j) - u_j)^2 w(\|x - x_j\|) \right] \]  

(2.4)

where \( n \) is the number of nodes in the support domain of \( x \). Rewriting in matrix form yields:

\[ J = (\mathbf{P} \mathbf{a} - \mathbf{u})^T \mathbf{W} (\mathbf{P} \mathbf{a} - \mathbf{u}) \]  

(2.5)

Where

\[ \mathbf{u}^T = [u_1, u_2, ... u_n] \]  

(2.6)

\[ \mathbf{P} = \begin{bmatrix} p_1(x_1) & p_2(x_1) & \cdots & p_m(x_1) \\ p_1(x_2) & p_2(x_2) & \cdots & p_m(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(x_n) & p_2(x_n) & \cdots & p_m(x_n) \end{bmatrix} = \begin{bmatrix} \mathbf{p}(x_1)^T \\ \vdots \\ \mathbf{p}(x_n)^T \end{bmatrix} \]  

(2.7)
Minimization is done by setting the partial derivatives of the error functional \( J \) to zero:

\[
\frac{\partial J}{\partial a} = P^T WP a(x) - P^T W u = 0
\]

(2.9)

By solving the resulting system of equations, the values of the coefficients at the evaluation point are obtained as:

\[
a(x) = (P^T WP)^{-1} P^T W u
\]

(2.10)

the classical MLS approximation function is then obtained as:

\[
u^h(x) = P^T (P^T WP)^{-1} P^T W u = \sum_{j=1}^{n} \varphi_j(x) u_j = \Phi^T(x) u
\]

(2.11)

where the shape functions are defined as:

\[
\Phi(x) = \begin{bmatrix} \varphi_1(x) & \ldots & \varphi_n(x) \end{bmatrix} = P^T (P^T WP)^{-1} P^T W
\]

(2.12)

Here, \( u \) is the nodal vector parameters of all the nodal field variables in the local support domain, \( \Phi(x) \) is the vector of classical MLS shape functions and the square matrix \( M = P^T WP \) is known as the moment matrix. Generally, linear or quadratic basis functions and cubic or quartic weight functions are used to create the MLS approximation. It should be noted that the approximation in Eq(2.11) and the shape functions are not polynomials even if the bases \( p(x) \) are polynomials.
The weight function plays an important role in the formulation of the MLS approximation: it provides weightings for the residuals at different nodes within the (compact) support domain and it ensures that nodes enter and leave the influence domain smoothly so that the shape functions satisfy the compatibility condition and the approximation is globally continuous.

As can be seen from Eq(2.10) and Eq(2.11), the shape functions construction depends on the non-singularity of the moment matrix. The necessary conditions for the moment matrix to be non-singular depend on the types of basis functions used. For example, in a two-dimensional case, the moment matrix is non-singular as long as there are at least 3 non-collinear nodes in the support domain for linear basis functions; whereas for a quadratic basis, at least 6 nodes are needed in the support domain. The support domain of a point \( x \) determines the number of nodes used to compute the approximation value at \( x \). However, some nodal distributions can still lead to singular moment matrices even if enough nodes are included in the support domain. This type of scenario can occur, for example, if the nodes are distributed on two parallel lines in 2D. To overcome this problem, the traditional remedy was to enlarge the support domains in order to include more nodes. This, however, leads to higher approximation error and increased computational cost. Consequently, these limitations prevent the practical use of higher order polynomial basis for randomly distributed nodes despite their potential capability for better approximation of complex deformation fields and better convergence properties.

### 2.3 Modified Moving Least Squares Approximation

A Modified Moving Least Squares Method (MMLS) (Joldes et al., 2015a) is presented in this section that will avoid singular moment matrices for higher order polynomial basis functions. The development of MMLS is based on the observation that a singular moment matrix mainly means that Eq(2.9) used to compute the coefficients \( a(x) \) has multiple solutions, and therefore the functional in Eq(2.4) does not include sufficient constraints to guarantee a unique solution for the
given nodal distribution. Based on this observation, for 2D, additional constraints are included in the functional as:

$$J(x) = \sum_{j=1}^{n} \left[ (u^h(x_j) - u_j)^2 w(||x - x_j||) \right] + \mu_x a_x^2 + \mu_y a_y^2 + \mu_{xy} a_{xy}^2 + \mu_{y^2} a_y^2$$  \hspace{1cm} (2.13)$$

where

$$\mu = \begin{bmatrix} \mu_x^2 & \mu_{xy} & \mu_y^2 \end{bmatrix}$$  \hspace{1cm} (2.14)$$

is defined as vector of positive weights for the additional constraints.

When the classical MLS moment matrix is singular, the minimization problem that needs to be solved to compute the coefficients has multiple solutions. The additional constraints ensure that, for such cases, a unique solution can be obtained, which corresponds to the coefficients of the second degree monomials in the basis being close or equal to zero. Furthermore, choosing the weights for the additional constraints as small positive numbers ensures that the classical MLS shape functions are altered only very slightly when the moment matrix in not singular.

The new functional can be rewritten in matrix form as:

$$\bar{J} = (Pa - u)^T W(Pa - u) + a^T Ha$$  \hspace{1cm} (2.15)$$

where \(H\) is a matrix with all elements zeros except the last three diagonal entries, which are equal to \(\mu\):

$$H = \begin{bmatrix} O_{33} & 0_{33} \\ 0_{33} & \text{diag}(\mu) \end{bmatrix}$$  \hspace{1cm} (2.16)$$

Following the same minimization procedure, the modified functional is obtained as:
\[
\frac{\partial \textbf{J}}{\partial \textbf{a}} = [\textbf{P}^T \textbf{W} \textbf{P} + \textbf{H}] \textbf{a}(\text{x}) - \textbf{P}^T \textbf{W} \textbf{u} = 0
\]  
(2.17)

The new coefficients can be computed as:

\[
\textbf{a}(\text{x}) = (\textbf{P}^T \textbf{W} \textbf{P} + \textbf{H})^{-1} \textbf{P}^T \textbf{W} \textbf{u}
\]  
(2.18)

as long as the modified moment matrix

\[
\textbf{M} = \textbf{P}^T \textbf{W} \textbf{P} + \textbf{H} = \textbf{M} + \textbf{H}
\]  
(2.19)

is non-singular. The modified approximant is finally obtained as:

\[
\overline{\textbf{u}}^h(\text{x}) = \textbf{P}^T (\textbf{P}^T \textbf{W} \textbf{P} + \textbf{H})^{-1} \textbf{P}^T \textbf{W} \textbf{u} = \sum_{j=1}^{n} \overline{\varphi}_j(\text{x}) u_j
\]  
(2.20)

with the new MMLS shape functions:

\[
\overline{\Phi}(\text{x}) = [\overline{\varphi}_1(\text{x}) \ldots \overline{\varphi}_n(\text{x})] = \textbf{P}^T (\textbf{P}^T \textbf{W} \textbf{P} + \textbf{H})^{-1} \textbf{P}^T \textbf{W}
\]  
(2.21)

The small alteration of the moment matrix presented in the above equation is the only difference between the MMLS and the classic MLS with quadratic basis functions.

2.4 Properties of the Modified Moving Least Squares Approximation

In this section, some useful and important properties of the modified MLS (MMLS) are discussed. These properties are: a) acceptable nodal distribution, b) continuity, c) consistency, and, d) invariance.
2.4.1 Acceptable nodal distribution

A nodal distribution is considered which is as acceptable only if the moment matrix is non-singular, allowing the computation of shape functions at any point in the domain.

**Lemma 1**: A nodal distribution which is acceptable for the classical MLS method with linear bases is also acceptable for the new modified MLS method with quadratic bases.

**Proof**: The moment matrix given by Eq(2.19) can be rewritten as:

\[
\bar{M} = \bar{P}^T \bar{W} \bar{P}
\]  

(2.22)

with

\[
P = \begin{bmatrix}
1 & x_1 & y_1 & y_1^2 & x_1 y_1 & y_1^2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
1 & x_n & y_n & y_n^2 & x_n y_n & y_n^2 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]  

(2.23)

and

\[
\bar{W} = \begin{bmatrix}
W & O_{n3} \\
O_{3n} & \text{diag}(\mu)
\end{bmatrix}
\]  

(2.24)

Eq(2.22) can be further transformed into:

\[
\bar{M} = \bar{P}^T \bar{Y}^T \bar{Y} \bar{P} = \bar{R}^T \bar{R}
\]  

(2.25)

where

\[
\bar{Y} = \text{sqrt}(W), \quad \bar{R} = \bar{Y} \bar{P}
\]  

(2.26)
Based on the matrix rank properties, because $Y$ is a diagonal matrix with non-zero (positive) diagonal elements (based on (2.24) and (2.26), the positive $W$ and $\mu$):

$$\text{rank}(\mathbf{R}) = \text{rank}(\mathbf{P})$$  \hspace{1cm} (2.27)

From Eq(2.25):

$$\text{rank}(\mathbf{M}) = \text{rank}(\mathbf{R}^T \mathbf{R}) = \text{rank}(\mathbf{R})$$ \hspace{1cm} (2.28)

and therefore, by combining Eq(2.27) and Eq(2.28):

$$\text{rank}(\mathbf{M}) = \text{rank}(\mathbf{P})$$ \hspace{1cm} (2.29)

Eq(2.29) shows that in order for the modified moment matrix to be non-singular, matrix $\mathbf{P}$ needs to have full rank (six in our case). Based on its definition in Eq(2.23), matrix $\mathbf{P}$ has full rank only if matrix

$$\mathbf{P}_1 = \begin{bmatrix} 1 & x_1 & y_1 \\ \vdots & \vdots & \vdots \\ 1 & x_n & y_n \end{bmatrix}$$ \hspace{1cm} (2.30)

has full rank. This condition is the same as the condition needed for the classical MLS method with linear bases to have a non-singular moment matrix - the support domain needs to contain at least three non-collinear nodes. This demonstration can be easily extended to 3D, where the moment matrix is non-singular if the support domain contains at least four non-coplanar nodes.

These restrictions on nodal distribution are a lot less severe than the restrictions for the classical MLS method with quadratic bases. The shape functions can be computed based on a reduced number of nodes, allowing smaller support domains and increased computational efficiency of the method.
2.4.2 Continuity

One of the major advantages of MLS approximation is its compatibility, which means the approximation field function is continuous and smooth in the entire problem domain.

**Lemma 2:** Let \( w^I(x) = w(||x - x_j||) \in C^l(\Omega) \) (derivatives up to order \( l \) are continuous). If \( \mu \) is a constant vector and the moment matrix \( \mathbf{M} \) is invertible at every point of \( \Omega \), then \( \mathbf{u}^h(x) \in C^l(\Omega) \).

**Proof:** From Eq(2.20), the approximation \( \mathbf{u}^h(x) \) on the whole problem domain is a span of all shape functions. Therefore, like for traditional MLS, the smoothness (or the order of continuity) of the approximation equals the smoothness of the shape functions, which is determined by the functions with the minimum order of continuity in Eq(2.21). Following the assumption that the moment matrix \( \mathbf{P}^T \mathbf{W} \mathbf{P} + \mathbf{H} \) is non-singular and since the monomials in the bases have \( C^\infty \) continuity, the smoothness of the shape functions is determined by the weight function and the vector \( \mu \) (which defines matrix \( \mathbf{H} \)). As a constant, \( \mu \) has \( C^\infty \) continuity; therefore the smoothness of the approximation function is solely determined by the smoothness of the weight functions, similar to the classical MLS approximation.

2.4.3 Consistency

A certain order of consistency is a prerequisite to ensure the convergence of the numerical results when nodal spacing is reduced; convergence implies that the numerical solution should approach the exact solution when the nodal spacing approaches zero (Liu, 2010). In case of MLS, the consistency of the approximation depends on the complete order of the monomials used in the basis. It has been shown in (Liu, 2010) that, if pure polynomial basis is assumed, the MLS shape function will possess \( C^k \) consistency (can represent exactly polynomial fields up to order \( k \)), where
$k$ is the complete order of the monomials in the basis. For the modified MLS, the following consistency property will be demonstrated:

**Lemma 3:** If the weight function satisfies the positivity requirement,

$$w(\|x - x_j\|) > 0, \quad x_j \in \Omega_s$$  \hspace{1cm} (2.31)

where $\Omega_s$ is the support domain of point $x$, then the MMLS shape functions based on quadratic basis ($m = 2$) will possess $C^1$ consistency.

**Proof:** Since the weight function and the components of vector $\mu$ are positive, the energy functional $J$ in Eq(2.13) is semi-positive definite. Therefore, its minimum has to be non-negative. Consider a function $u(x) \in \text{span}\{p_j(x)\}$ given by

$$u(x) = \sum_{j}^{r} p_j(x)a_j, \quad r < m$$  \hspace{1cm} (2.32)

The approximation field can always be rewritten as:

$$u(x) = \sum_{j}^{m} p_j(x)a_j(x)$$  \hspace{1cm} (2.33)

by simply assigning

$$a_j(x) = \begin{cases} a_j, & j \leq r \\ 0, & r < j \leq m \end{cases}$$  \hspace{1cm} (2.34)

For such a field $\bar{J}$ will vanish and it will necessarily be a minimum, and therefore

$$\bar{u}^h(x) = \sum_{j}^{r} p_j(x)a_j(x) = u(x)$$  \hspace{1cm} (2.35)
This proves that any field given by Eq(2.32) will be reproduced exactly by the MMLS approximation.

**Corollary 1:** The shape functions created by the MMLS approximation form a Partition of Unity:

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

(2.36)

This is a necessary condition for the shape functions to be able to represent any rigid motion of the problem domain.

In the application of approximation methods to numerical solutions of partial differential equations, practitioners often judge an interpolation method by its ability to pass the patch test, meaning it can reproduce exactly linear functions. In engineering applications this often applies to exact calculation of constant stress and strain.

**Corollary 2:** The shape functions created by the modified MLS approximation possess the linear field reproduction property:

\[
\sum_{j}^{n} \Phi_j (x)x_j = x
\]  

(2.37)

The above two Corollaries are a direct result of Lemma 3. \(C^0\) consistency, which means constant fields can be exactly reproduced by the approximation, is a particular case of partition of unity. The linear field reproduction property (patch test) is identical to the \(C^1\) consistency condition.

### 2.4.4 Invariance

The invariance to translation and scaling is an important property for meshless shape functions which indicates that the shape functions and the approximation do not depend on the global nodal positions. In other words, any translation or scaling of the nodal positions to a local
coordinate system should not change the values of the shape functions. This allows the use of a canonical domain in the computation of shape functions, which reduces the rounding errors which may occur due to the inclusion of too large or too small entries in the moment matrix. For the sake of simplicity and without loss of generality, the following proofs are presented only for the 2D case.

**Lemma 4:** The modified MLS shape functions are invariant to the translation of nodes - the shape functions for nodes \( \mathbf{x}_j^* = (\mathbf{x}_j + \mathbf{d}) \) have the same values as the shape functions for nodes \( \mathbf{x}_j \):

\[
\Phi^*(\mathbf{x} + \mathbf{d}) = \Phi(\mathbf{x}) \tag{2.38}
\]

**Proof:** Based on Eq(2.21), the shape functions for nodes \( \mathbf{x}_j^* \) can be written as:

\[
\Phi^* = p^T (p^T WP^* + H)^{-1} P^T W \tag{2.39}
\]

with

\[
P = \begin{bmatrix}
p(x_1^*)^T \\
p(x_2^*)^T \\
\vdots \\
p(x_n^*)^T
\end{bmatrix} \tag{2.40}
\]

The weight functions \( W \) do not change since they are functions of relative distances, which are not affected by translations. Therefore,

\[
\Phi^*(\mathbf{x} + \mathbf{d}) = p(\mathbf{x} + \mathbf{d})^T (P^T WP^* + H)^{-1} P^T W \tag{2.41}
\]

It is to be noted that
\[
\mathbf{p}(x + d) = \begin{bmatrix}
1 & x + d_x \\
x + d_x & y + d_y \\
(x + d_x)^2 & (x + d_x)(y + d_y) \\
(y + d_y)^2 & (y + d_y)^2
\end{bmatrix} = \mathbf{T} \begin{bmatrix}
1 \\
x \\
y \\
x^2 + dy^2
\end{bmatrix} = Tp(x) \quad (2.42)
\]

with
\[
\mathbf{T} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
d_x & 1 & 0 & 0 & 0 \\
d_y & 0 & 1 & 0 & 0 \\
d_x^2 & 2d_x & 0 & 1 & 0 \\
d_xd_y & d_y & d_x & 0 & 1 \\
d_y^2 & 0 & 2d_y & 0 & 1
\end{bmatrix} \quad (2.43)
\]

Combining Eq(2.40) and Eq(2.43):
\[
\mathbf{P}^* = \begin{bmatrix}
\mathbf{p}(x_1 + d)^T \\
\mathbf{p}(x_2 + d)^T \\
\vdots \\
\mathbf{p}(x_n + d)^T
\end{bmatrix} = \begin{bmatrix}
\mathbf{p}(x_1)^T \mathbf{T}^T \\
\mathbf{p}(x_2)^T \mathbf{T}^T \\
\vdots \\
\mathbf{p}(x_n)^T \mathbf{T}^T
\end{bmatrix} \mathbf{T}^T = \mathbf{P} \mathbf{T}^T \quad (2.44)
\]

Replacing Eq(2.44) and Eq(2.42) in Eq(2.41):
\[
\tilde{\Phi}^*(x + d) = \mathbf{p}(x)^T \mathbf{T}^T (\mathbf{T} \mathbf{P} \mathbf{W} \mathbf{P}^T + \mathbf{H})^{-1} \mathbf{P}^T \mathbf{W} = \mathbf{p}(x)^T (\mathbf{P}^T \mathbf{W} \mathbf{P} + \mathbf{T}^{-1} \mathbf{H} \mathbf{T}^{-T})^{-1} \mathbf{P}^T \mathbf{W} \quad (2.45)
\]

It is to be noted that if
\[
\mathbf{T}^{-1} \mathbf{H} \mathbf{T}^{-T} = \mathbf{H} \quad (2.46)
\]

then Eq(2.38) is satisfied and the lemma is proved. Eq(2.46) can be re-ordered into:
\[
\mathbf{H} = \mathbf{T} \mathbf{H} \mathbf{T}^{-T} \quad (2.47)
\]

By rewriting \(\mathbf{T}\) from Eq(2.43) using 3x3 sub-matrices and considering the definition of \(\mathbf{H}\) in Eq(2.16), it can be seen that:
\[
\mathbf{THT}^{-T} = \begin{bmatrix}
  T_{11} & 0_{33} \\
  T_{21} & I_{33}
\end{bmatrix}
\begin{bmatrix}
  0_{33} & 0_{33} \\
  diag(\mu) & 0_{33}
\end{bmatrix}
\begin{bmatrix}
  T_{11}^T \\
  T_{21}^T
\end{bmatrix}
= \begin{bmatrix}
  O_{33} & O_{33} \\
  O_{33} & diag(\mu)
\end{bmatrix}
= \mathbf{H}
\tag{2.48}
\]

and therefore the lemma is proved.

**Lemma 5:** Consider the scaling of nodal positions \( \mathbf{x}_j^* = S\mathbf{x}_j \) with \( S \) a scaling matrix:

\[
S = \begin{bmatrix}
  s_x & 0 \\
  0 & s_y
\end{bmatrix}
\tag{2.49}
\]

If the following conditions are met:

i) The influence domains are scaled in such a way that the weight functions remain unchanged:

\[
w(||S\mathbf{x} - S\mathbf{x}_j||) = w(||\mathbf{x} - \mathbf{x}_j||)
\tag{2.50}
\]

ii) The weights for the additional constrains are changed to:

\[
\mathbf{\mu}^* = \begin{bmatrix}
  s_x^4 \mu_x^2 \\
  s_x^2 s_y^2 \mu_{xy} \\
  s_y^4 \mu_y^2
\end{bmatrix}
\tag{2.51}
\]

then the MMLS shape functions are invariant to the scaling of nodal positions - the shape functions for nodes \( \mathbf{x}_j^* \) have the same values as the shape functions for nodes \( \mathbf{x}_j \):

\[
\vec{\Phi}^*(S\mathbf{x}) = \vec{\Phi}(\mathbf{x})
\tag{2.52}
\]

**Proof:** From Eq(2.50), the shape functions for nodes \( \mathbf{x}_j^* \) are:

\[
\vec{\Phi}^* = \mathbf{p}^T (\mathbf{P}^{\ast T} \mathbf{Wp}^* + \mathbf{H}^*)^{-1} \mathbf{P}^{\ast T} \mathbf{W}
\tag{2.53}
\]

with
\[ P^* = \begin{bmatrix} p(x_1^T) \\ p(x_2^T) \\ \vdots \\ p(x_n^T) \end{bmatrix} \] (2.54)

Therefore,

\[ \Phi^*(Sx) = p((Sx)^T(P^*W^* + H^*)^{-1}P^T W \] (2.55)

It is to be noted that

\[ p(Sx) = \begin{bmatrix} 1 \\ x s_x \\ \vdots \\ x^2 s_{x^2} \\ y s_y \\ x y s_y \\ y^2 s_{y^2} \end{bmatrix} \quad = R = \begin{bmatrix} 1 \\ x \\ y \\ x^2 \\ y^2 \end{bmatrix} = Rp(x) \] (2.56)

with

\[ R = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & s_x & 0 & 0 & 0 & 0 \\ 0 & 0 & s_y & 0 & 0 & 0 \\ 0 & 0 & 0 & s_{x^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & s_{x^2 y} & 0 \\ 0 & 0 & 0 & 0 & 0 & s_{y^2} \end{bmatrix} \] (2.57)

Combining Eq(2.54) and Eq(2.57):

\[ P^* = \begin{bmatrix} p(Sx_1)^T \\ p(Sx_2)^T \\ \vdots \\ p(Sx_n)^T \end{bmatrix} = \begin{bmatrix} p(x_1)^T R^T \\ p(x_2)^T R^T \\ \vdots \\ p(x_n)^T R^T \end{bmatrix} = \begin{bmatrix} p(x_1)^T \\ p(x_2)^T \\ \vdots \\ p(x_n)^T \end{bmatrix} R^T = PR^T \] (2.58)

Replacing Eq(2.58) and Eq(2.56) in Eq(2.55):

\[ \Phi^*(Sx) = p(x)^T R^T (RP^T WPR^T + H^*)^{-1}RP^T W = p(x)^T (P^T WP + R^{-1} H^* R^{-T})^{-1}P^T W \] (2.59)

It is to be noticed that if
\[ R^{-1}H^*R^{-T} = H \]  \hspace{1cm} (2.60)

then Eq(2.52) is satisfied and the lemma is proved. Given the definition of \( H \), it is easy to show that relation in Eq(2.60) is satisfied as long as the weights for the additional constrains given by Eq(2.51) are used in \( H^* \).

### 2.5 Numerical Examples

To assess the approximations capability of the modified MLS shape functions with quadratic basis, it is compared with classical MLS shape function with linear and quadratic basis implemented in Matlab for 1D and 2D. A quartic spline weight function with circular domain was used in the definition of all shape functions:

\[ w(s) = \begin{cases} 
1 - 6s^2 + 8s^3 - 3s^4, & s \leq 1 \\
0, & s > 1 
\end{cases} \]  \hspace{1cm} (2.61)

where \( s \) is the normalized distance

\[ s_j = \frac{\|x - x_j\|}{R_j} \]  \hspace{1cm} (2.62)

and \( R_j \) is the radius of the influence domain of node \( x_j \). The influence domain is defined for each node and the radius of influence domain may be constant and different from node to node. For a given background cell, the variable \( R_j \) can be calculated in the preprocessing stage by finding the average distance between the node and its surrounding nodes, multiplied by dilatation parameter. Changing the dilatation parameter scales the influence of nodes (this technique is used to calculate influence domain radius in Section 4.4.3). In this section, a constant radius of influence for all nodes \( (R_j = R) \) is used to test the stability of the MMLS shape functions. The positive weights for the additional constraints are chosen to be small positive numbers. This technique not only ensures that
a stable classical MLS solution is not affected too much, but also favours the solution having the
coefficients of the higher order monomials in the basis close to zero when classical MLS moment
matrix becomes singular. In the examples, same weights for all the additional constraints
($\mu_x^2 = \mu_{xy} = \mu_y^2 = \mu$) were used.

2.5.1 MLS and MMLS shape functions in 1D

A comparison between the MLS and the MMLS shape functions is presented in Figure 2.1
for different values of $\mu$ and $R$.

![MLS and MMLS shape functions comparison](image)

Figure 2.1: MLS and MMLS shape functions comparison for node 6.  
a) $R = 2.5$. b) $R = 1.3$; the classical MLS with quadratic base functions
(BF) has a singular moment matrix for part of the domain
For the radius of influence, the larger value \((R=2.5)\) was chosen such that more than three nodes are in the support domain of any point in the interval, while the lower value \((R=1.3)\) only ensures that two nodes are in the support domain of any point in the interval. For small radius of influence \(R\) the classical MLS with quadratic base functions has singular moment matrix for parts of the domain, while the modified MLS does not have such a problem. The value of the weight parameter \(\mu\) influences the shape functions in such a way that when the value of \(\mu\) becomes smaller, the MMLS shape functions are closer to those of the classical MLS with quadratic base functions.

### 2.5.2 Approximation capability in 1D

Using the same nodal distribution as in the above example, a non-polynomial function, \(u(x) = \sin(x)\), is approximated using MLS and the MMLS for different values of \(\mu\) and \(R\). The approximation accuracy was determined using the Root Mean Square Error (RMSE):

\[
RMSE = \sqrt{\frac{\sum_{i=1}^{N}(u(x) - u^h(x))^2}{N}}
\]  

(2.63)

RMSE is computed using \(N=801\) points equally distributed in the interval \([-4, 4]\). The results are presented in Figure 2.2 and Table 2.1.

**Table 2.1 : Root mean square error in approximating \(u(x)=\sin(x)\) using the points presented in Figure 2.2.**

<table>
<thead>
<tr>
<th>Approximation method</th>
<th>Radius of nodal influence domain, (R)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(R = 2.5)</td>
</tr>
<tr>
<td>MLS, linear BF</td>
<td>0.1765</td>
</tr>
<tr>
<td>MLS, quadratic BF</td>
<td>0.0297</td>
</tr>
<tr>
<td>MMLS, (\mu = 0.1)</td>
<td>0.0355</td>
</tr>
<tr>
<td>MMLS, (\mu = 0.01)</td>
<td>0.0301</td>
</tr>
</tbody>
</table>
The results show that the approximation accuracy of the MMLS is better than that of the classical MLS with linear base functions, approaching the accuracy of classical MLS with quadratic base functions as the value of parameter $\mu$ decreases. Figure 2.2 clearly shows the advantage of using higher degree base functions in terms of approximation accuracy.

### 2.5.3 Approximation capability in 2D

The following functions are used for testing the approximation accuracy in 2D using MLS and the MMLS for different values of $\mu$ and $R$. The chosen function combines rapid transitions between peaks and dips with almost flat regions over the domain.

\[ u(x, y) = xe^{-x^2-y^2} \]  \hspace{1cm} (2.64)

\[ u(x, y) = (x^2 - y^2)e^{-x^2-y^2} \]  \hspace{1cm} (2.65)
2D rectangular problem domain was defined and the geometry was represented using both regular and irregular node distributions consisting of 324 nodes, as shown in Figure 2.3. The irregular nodal distribution consisted of uniformly scattered nodes were obtained by using the Matlab Halton sequence function (Fasshauer, 2007). The approximation accuracy was determined using the root mean square error (RMSE) evaluated using a regular distribution of N=81*81 points. The results are presented in

![Regular Node Distribution, N=324](image1)

![Irregular Node Distribution, N=324](image2)

**Figure 2.3 :** a) Regular node distribution, b) Irregular node distribution.
Table 2.2: Root mean square error (RMSE) in approximating Eq(2.64) using 324 nodes with varying radius of nodal influence domain, R.

<table>
<thead>
<tr>
<th>Approximation method</th>
<th>Regular node distribution</th>
<th>Irregular node distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R = 2.0</td>
<td>R = 1.5</td>
</tr>
<tr>
<td>MLS, linear BF</td>
<td>0.0344</td>
<td>0.0272</td>
</tr>
<tr>
<td>MLS, quadratic BF</td>
<td>0.0081</td>
<td>0.0058</td>
</tr>
<tr>
<td>MMLS, μ = 0.1</td>
<td>0.0109</td>
<td>0.0106</td>
</tr>
<tr>
<td>MMLS, μ = 0.01</td>
<td>0.003</td>
<td>0.006</td>
</tr>
<tr>
<td>MMLS, μ = 0.001</td>
<td>0.0081</td>
<td>0.0059</td>
</tr>
<tr>
<td>MMLS, μ = 0.0001</td>
<td>0.0081</td>
<td>0.0058</td>
</tr>
</tbody>
</table>

Table 2.3: Root mean square error (RMSE) in approximating Eq(2.65) using 324 nodes with varying radius of nodal influence domain, R.

<table>
<thead>
<tr>
<th>Approximation method</th>
<th>Regular node distribution</th>
<th>Irregular node distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R = 1.5</td>
<td>R = 0.8</td>
</tr>
<tr>
<td>MLS, linear BF</td>
<td>0.0366</td>
<td>0.0136</td>
</tr>
<tr>
<td>MLS, quadratic BF</td>
<td>0.0107</td>
<td>Singular M</td>
</tr>
<tr>
<td>MMLS, μ = 0.1</td>
<td>0.0158</td>
<td>0.0127</td>
</tr>
<tr>
<td>MMLS, μ = 0.001</td>
<td>0.0108</td>
<td>0.0058</td>
</tr>
<tr>
<td>MMLS, μ = 0.0001</td>
<td>0.0107</td>
<td>0.0053</td>
</tr>
</tbody>
</table>
Figure 2.4: Approximated function, a) $u(x, y) = xe^{-x^2-y^2}$ and
b) $u(x, y) = (x^2 - y^2)e^{-x^2-y^2}$
From the results, it can be seen that as the nodal influence domain radius is gradually decreased, the classic MLS with quadratic basis fails due to singular moment matrix whereas the modified MLS with quadratic basis is stable. The approximation accuracy of MMLS is found to be better than that of classical MLS with linear basis function. Moreover, it is also evident that if the value of $\mu$ is decreased, the MMLS accuracy approaches the accuracy of classical MLS with quadratic basis function.

Table 2.4 shows the reciprocal condition numbers of the moment matrices in approximating Eq(2.65) using 324 nodes with radius of nodal influence domain, $R = 0.8$. The moment matrix becomes singular if its condition number is infinite or the reciprocal of its condition number approaches the machine’s floating point precision (Golub and Van Loan, 1989). In case of such highly ill-conditioned moment matrices, the solver fails to compute the shape functions, as occurred in computing the classic MLS with quadratic basis functions for the given influence domain radius. Furthermore, it is observed that MMLS has higher condition numbers compared with the classic MLS with linear basis functions as higher order polynomials are included in the MMLS shape functions.

Table 2.4 : Reciprocal condition numbers of Moment Matrices, $\text{rcond}(M)$, in approximating Eq(2.65) using 324 nodes with radius of nodal influence domain, $R = 0.8$.

<table>
<thead>
<tr>
<th>Approximation method</th>
<th>$\text{rcond}(M)$, in approximating $u(x, y) = (x^2 - y^2)e^{-x^2-y^2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Regular node distribution</td>
</tr>
<tr>
<td>MLS, linear BF</td>
<td>$1.9120 \times 10^{-5}$</td>
</tr>
<tr>
<td>MLS, quadratic BF</td>
<td>$7.3646 \times 10^{-8}$</td>
</tr>
<tr>
<td>MMLS (quadratic BF)</td>
<td>$9.5690 \times 10^{-10}$</td>
</tr>
<tr>
<td>$\mu = 0.1$</td>
<td>$1.2080 \times 10^{-10}$</td>
</tr>
<tr>
<td>MMLS (quadratic BF)</td>
<td>$9.5690 \times 10^{-10}$</td>
</tr>
<tr>
<td>$\mu = 0.001$</td>
<td>$1.2080 \times 10^{-10}$</td>
</tr>
<tr>
<td>MMLS (quadratic BF)</td>
<td>$9.5690 \times 10^{-10}$</td>
</tr>
<tr>
<td>$\mu = 0.0001$</td>
<td>$1.2080 \times 10^{-10}$</td>
</tr>
</tbody>
</table>
2.6 Concluding Remarks

The Modified Moving Least Squares (MMLS) presented in this chapter is based on the error functional used in the derivation of the classical MLS approximation augmented with additional terms based on the coefficients of the polynomial base functions. This allows quadratic polynomial base functions to be used with the same size of the support domain as linear base functions, resulting in better approximation capability while maintaining the continuity and smoothness of the approximation. The computation of the MMLS shape functions involves a slight modification of the moment matrix of the classical MLS. Nevertheless, the MMLS has better computation efficiency because it does not need the radius of influence to be as large as for the classical MLS with quadratic basis (requires less nodes in the support domain of each point of interest).

The numerical examples show that the approximation accuracy of the MMLS is better than that of the classical MLS with linear base functions, approaching the accuracy of classical MLS with quadratic base functions as the value of parameter $\mu$ (the weights for the additional constraints) decreases. An important benefit of the proposed method is the ability to provide an approximation for cases when classical MLS with quadratic base functions fails due to singular moment matrices.

It was shown that the MMLS approximation has the required order of consistency to be used as an approximation in a meshless method (the resulting shape functions form a partition of unity and the approximation has the linear field reproducing property). The invariance of the shape functions against translation and scaling have also been derived. These properties are important for a robust numerical implementation of the method, allowing scaling of the domain which contains the nodes involved in the shape functions computation to a canonical domain, in order to avoid rounding errors in the inversion of the moment matrix. In the next chapter, the presented MMLS approximation will be used in the development of a more robust Element Free Galerkin meshless method.
Chapter 3

An Element Free Galerkin Method Based on the Modified Moving Least Squares Approximation
Most of the contents of this chapter have been published on:

3.1 Introduction

The meshless shape functions are important in approximating the unknown field functions to find an approximate solution to a problem governed by PDEs and boundary conditions (Liu, 2010). A number of ways have been proposed over the years to construct meshless shape functions (Liu, 2010, Liu and Gu, 2005, Fasshauer, 2007). According to the literatures, shape functions used in meshless methods should satisfy some basic requirements (Liu, 2010): they should be sufficiently robust for reasonably arbitrarily distributed nodes; should be numerically stable; should have a certain order of consistency; should be compactly supported (should be zero outside a bounded region called the support domain); should be computationally efficient and should ideally possess the Kronecker delta function property. Not having the Kronecker delta function property means that the set of meshless shape functions forms a partition of the unity, but the shape function associated to a particle does not vanish at all other particles; this makes difficult the imposition of essential boundary conditions. The Point Interpolation Method (PIM), proposed by Liu and Gu (Liu, 2010), possess the Kronecker delta function property. However, PIM requires the number of basis functions to be equal to the number of nodes in the support domain, increasing the chance of a singular moment matrix for arbitrarily distributed nodes. PIM shape functions are also not compatible over the problem domain (Liu, 2010).

Due to its continuity and smoothness properties, the Moving Least Squares (MLS) approximation has been the preferred choice in Element Free Galerkin (EFG) method. However, to compute the shape functions, the classical MLS places strict requirements on the nodal distributions inside each support domain; the practical use of higher order polynomial basis, which can generate more accurate approximations of complex deformation fields, is not trivial in classical MLS for randomly distributed nodes. In this context, a modified moving least squares (MMLS) approximation has been developed (Joldes et al., 2015a, Chowdhury et al., 2015) that has already been discussed in Chapter 2.
The MMLS shape functions must also satisfy the requirements that the classical MLS possess to be used in EFG (Liu, 2010). Some of these properties (acceptable node distribution, continuity and the necessary order of consistency) have already been derived in the previous chapter. Furthermore, the invariance properties of MMLS which is necessary for an accurate computation of the shape functions have also been discussed. Other properties (compatibility of the field approximation, compact support) are ensured by selecting appropriate weight functions (Liu, 2010).

In this chapter, the MMLS approximation is used in the Meshless Total Lagrangian Explicit Dynamics (MTLED) algorithm [11, 5] to develop a more robust MM based on EFG principle for surgical simulations. The chapter is organised as follows: a discussion of MTLED algorithm is presented in Section 3.2; the MMLS approximation for 3D is derived Section 3.3; numerical examples are presented in Section 3.4 and the concluding remarks are presented in Section 3.5.

3.2 Meshless Total Lagrangian Explicit Dynamics Algorithm

A brief discussion of the specific details of MTLED algorithm is presented in this section.

3.2.1 Total Lagrangian Explicit Dynamics

The Total Lagrangian (TL) formulation is used for calculating forces from stresses and displacements where all the calculations refer to the initial configuration of the analyzed continuum. All derivatives with respect to spatial coordinates are computed during the preprocessing stage. This eliminates the necessity of such large computations at every time step like the Updated Lagrangian formulation. After introducing MMLS approximation into the weak form of governing equations of solid mechanics using the TL formulation, the global system of discretized equations describing the behavior of the analyzed continuum becomes the following:
\[
M\ddot{\textbf{u}} + \textbf{F}_{\text{int}} = \textbf{F}_{\text{ext}}
\] (3.1)

where \(\textbf{u}\) is the vector of nodal displacements, \(M\) is the mass matrix, \(\textbf{F}_{\text{int}}\) is the global nodal reaction force vector and \(\textbf{F}_{\text{ext}}\) is the vector of externally applied force at time \(t\). The vector of internal nodal forces \((\textbf{F}_{\text{int}})\) is computed as:

\[
\textbf{F}_{\text{int}} = \int_{V_0} \text{\(\bigtriangledown\)}X_0 \text{\(\bigtriangledown\)}B_{L0}^T \text{\(\bigtriangledown\)}S \, dV_0
\] (3.2)

where \(\text{\(\bigtriangledown\)}X\) is the deformation gradient at time \(t\), \(\text{\(\bigtriangledown\)}S\) is the second Piola-Kirchoff stress at time \(t\), \(\text{\(\bigtriangledown\)}B_{L0}\) is the matrix of shape function derivatives and \(V_0\) is the initial volume of the problem domain.

The explicit integration in time domain is applied using the central difference method. Explicit time integration is a direct integration method where nodal accelerations are found directly without any iteration and then integrated to obtain the displacements. There is no necessity in assembling a global stiffness matrix. The time stepping scheme for solving the equation of motion can be expressed as:

\[
^t\textbf{u} + 1 = \Delta t^2 M^{-1}(\textbf{F}_{\text{ext}} - \textbf{F}_{\text{int}}) + 2^t\textbf{u} - ^t\textbf{u}
\] (3.3)

where \(^t\textbf{u}\) is the displacement calculated at time \(t\), \(M\) is the constant diagonal mass matrix and \(\Delta t\) is the time step.

### 3.2.2 Dynamic Relaxation

In the Dynamic Relaxation (DR) algorithm, a damping force is introduced to the equation of motion to dissipate the kinetic energy when the steady state of the deformed continuum needs to be
obtained. 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.

\[ M\ddot{u} + cM\dot{u} = \dot{F}_{ext} - \dot{F}_{int} \]  \hspace{1cm} (3.4)

where \( cM\dot{u} \) is the damping force and \( c \) is the damping coefficient. The resulting equation
describing the iterations in terms of displacements is derived as:

\[ t^{+1}u = t^u + \beta(t^u - t^{-1}u) + \alpha M^{-1}(\dot{F}_{ext} - \dot{F}_{int}) \]  \hspace{1cm} (3.5)

where \( \alpha = 2h^2/(2 + ch) \), \( \beta = (2 - ch)/(2 + ch) \) and \( h \) is a fixed time increment.

In the relaxation stage, the integration time step \( \Delta t \) is kept constant, while the damping
coefficient \( c \) and lumped mass matrix \( M \) are initiated and automatically adjusted to maximize the
convergence rate and improve the computational efficiency without compromising the solution
convergence. An overview is of the MTLED is presented in Figure 3.1.

![Figure 3.1: An overview of MTLED algorithm (Zhang et al., 2013)](image-url)
3.2.3 Material formulation

The soft tissues exhibit non-linear stress-strain relationship and strain rate dependency (Bilston, 2001, Fung, 1993, Miller, 2000). The behaviour of soft tissue is commonly described in literatures using three constitutive models: a) Elastic material, b) Hyperviscoelastic material, and c) Hyperelastic material (Wittek et al., 2010, Wittek et al., 2009, Miller, 2001). Linear elastic almost incompressible constitutive model uses Hooke’s law. Hyperviscoelastic material model represents soft tissue behaviour such as non-linear stress-strain relationship, bi-modal response and non-linear stress-strain rate relationship for strains up to 30% (Miller and Chinzei, 2002). If the strain rate dependency is insignificant, which in case of surgical procedures is true as the strain rates are small and do not have large variations, then the viscoelastic terms can be ignored hyperviscoelastic material to produce a hyperelastic constitutive model.

In all numerical examples presented in this thesis, the deformation field of the soft tissue organ is computed by prescribing loading as forced motion of the boundary. This type of formulation is known as the displacement-zero-traction problem and for such formulation, the predicted deformations within the analysed continuum depend very weakly on the mechanical properties (Wittek et al., 2010, Wittek et al., 2009). Furthermore, the purpose of this thesis is to develop and verify algorithms rather than to perform complete simulations of actual surgery. Following these reasons, the simple hyperelastic Neo-Hookean material model is used in the numerical experiments in this thesis. The Neo-Hookean material model is a fully nonlinear model which is appropriate for the prediction of soft tissue deformation in surgical simulation and also easy to implement.

The Neo-Hookean material model has the following strain-energy density functional:

\[
W = \frac{\mu}{2} (\overline{I}_1 - 3) + \lambda (J - 1)^2
\]  

(3.6)

with
where, $\mu$ and $\lambda$ are the Lamé parameters; $J$ is the elastic volume ratio which equals to the determinant of the deformation gradient, $J = \det(\dot{\mathbf{X}}_0)$; and $\bar{I}_1$ is the first invariant of the right Cauchy Green deformation tensor.

Therefore, the second Piola-Kirchhoff stress in Eq(3.2) is evaluated as:

$$\bar{S}_0 = \lambda J (J - 1) \dot{\mathbf{C}}^{-1} + \mu J^{2/3} I $$  \hspace{1cm} (3.10)

The next section presents the development of the Modified Moving Least Squares (MMLS) approximation for 3D applications.

### 3.3 Modified Moving Least Squares in 3D

The classical Moving Least Squares (MLS) shape function construction starts with the approximation of a function $u(x)$, denoted by $u^h(x)$, as discussed in Section 2.2:

$$u^h(x) = \sum_{i=1}^{m} p_i(x)a_i(x) = \mathbf{p}^T(x)a(x)$$  \hspace{1cm} (3.11)

where $m$ is the number of terms in the basis $\mathbf{p}(x)$, and $a_i(x)$ are coefficients that depend on the spatial coordinates $x$. These coefficients are computed by minimizing an error functional defined based on the weighted least squares errors:
The minimization is done by setting the partial derivatives of the error functional $J$ to zero. However, as explained in Section 2.3, in the context of singular moment matrices, the functional $J$ does not include sufficient constraints to guarantee a unique solution for the given nodal distribution because Eq(3.12) used to compute the coefficients $a(x)$ has multiple solutions. Therefore, in the MMLS approximation for 3D, additional constraints are included in the functional as:

$$
\bar{J}(x) = \sum_{j=1}^{n} \left[ (u^h(x_j) - u_j)^2 w(||x - x_j||) \right] + \mu_x a_x^2 + \mu_y a_y^2 + \mu_z a_z^2 + \mu_{xy} a_{xy}^2 + \mu_{xz} a_{xz}^2 + \mu_{yz} a_{yz}^2 
$$

(3.13)

where $\mu$ are the positive weights for the additional constraints:

$$
\mu = \begin{bmatrix}
\mu_x^2 & \mu_y^2 & \mu_z^2 & \mu_{xy} & \mu_{xz} & \mu_{yz}
\end{bmatrix}
$$

(3.14)

Following the minimization procedure shown in Section 2.3, the MMLS shape functions are derived as:

$$
\Phi(x) = \begin{bmatrix}
\bar{\phi}_1(x) & ... & \bar{\phi}_n(x)
\end{bmatrix} = P^T (P^T W P + H)^{-1} P^T W
$$

(3.15)

where $H$ is a 10x10 matrix with all elements zeros except the last six diagonal entries equal to $\mu$:

$$
H = \begin{bmatrix}
O_{44} & O_{46} \\
O_{64} & diag(\mu)
\end{bmatrix}
$$

(3.16)

It has been shown in Section 2.4.1 that the nodal distributions which are admissible for the classical MLS with linear basis functions are also admissible for the MMLS which uses second order quadratic basis. This requires at least 3 non-collinear nodes in 2D and 4 non-coplanar nodes in 3D to be found in the support domain of any point where the shape functions are computed.
3.4 Numerical Examples

In this section, three cases of biomechanics applications – an extension of a soft tissue sample, craniotomy induced brain deformation and unconstrained compression of a cube – are simulated using the MTLED algorithm. The modified MLS shape functions are incorporated in the MTLED algorithm. For easy imposition of the essential boundary conditions, in Section 3.4.1-3.4.2, a regularized weight function (Most and Bucher, 2005) is used which possesses almost interpolating properties, as shown in Figure 3.2. In Section 3.4.3, the essential boundary conditions are imposed by coupling the finite element method (FEM) and MLS shape functions near the essential boundaries (Li and Liu, 2004, Zhang et al., 2014).

![Figure 3.2: Comparison between Classic MLS and Modified MLS (\(\mu = 0.01\)) with regularized weight function using 12 nodes in 1-D with influence domain radius, \(R = 3\); a) regular, b) irregular nodal distributions.](image)

The results obtained from the meshless simulations are compared with those obtained using the commercial finite element software ABAQUS.
3.4.1 Extension of a Soft Tissue Sample in 2D

For the meshless computation of soft tissue extension, a 2D geometry (10cm x 4cm) was created and the domain and boundary were discretised using 57 nodes. To ensure integration accuracy, a regular background grid was used consisting of 1000 integration cells with one integration point per cell. For each node, the radius of the influence domain was constant ($R = 1.4$). Loading in terms of displacement (3cm) was applied to the nodes on the right hand side boundary and the left boundary nodes were fixed. Explicit integration was performed using the central difference method, with mass proportional damping added (dynamic relaxation) to control the oscillations in order to reach the steady state solution (Joldes et al., 2009a, Joldes et al., 2011).

For simplicity, the hyper-elastic Neo-Hookean material model was chosen in this numerical experiment to capture the behaviour of soft tissues undergoing large deformation. For the finite element analysis in ABAQUS, identical constitutive material laws, loading and boundary conditions were used; the steady state solution was obtained using the static solver with the default configuration. The simulation results and numerical details are presented in Figure 3.3 and Table 3.1.

<table>
<thead>
<tr>
<th>Approximation method</th>
<th>Nodes</th>
<th>Elements (ABAQUS)</th>
<th>Integration points (Meshless)</th>
<th>Average difference (mm)</th>
<th>Maximum difference (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) Classical MLS (Linear Basis)</td>
<td>57</td>
<td>84</td>
<td>1000</td>
<td>0.14996</td>
<td>0.73014</td>
</tr>
<tr>
<td>b) MMLS ($\mu = 10^{-10}$) (Quadratic Basis)</td>
<td>57</td>
<td>84</td>
<td>1000</td>
<td>0.10193</td>
<td>0.48192</td>
</tr>
<tr>
<td>c) Classical MLS (Quadratic Basis)</td>
<td>Fails to compute due to singular moment matrices</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1 : Numerical details of comparison for the extension of a soft tissue sample simulation (Figure 3.3)
For the given nodal influence domain radius, the classical MLS with quadratic basis failed due to the singularity of moment matrix. The differences in computed deformation fields over the whole problem domain are shown in Fig.4. The results in Table 2 shows that the maximum and average difference in displacements between MMLS and ABAQUS are lower compared to those between classic MLS with linear basis and ABAQUS.

Figure 3.3: Differences of the computed deformation field over the whole problem domain, a) between Classical MLS (linear basis) and ABAQUS; b) between Modified MLS and ABAQUS.
3.4.2 Craniotomy Induced Brain Shift in 2D

This example, a craniotomy induced brain deformation simulation in 2D, demonstrates the use of the proposed method in a complex example, involving complicated geometry, non-linear material, large deformations and contacts. Due to a number of physical and physiological reasons, the brain deforms after craniotomy, a phenomenon also known as ‘brain shift’ (Miller, 2011). High quality pre-operative images are used for identifying the areas of interest (e.g. the tumour) and creating the biomechanical model, which is then used for computing the intra-operative position of the tumour within the brain after craniotomy.

In order to simulate brain deformation, based on experimental data (Miller et al., 2000) and previous modelling experience (Joldes et al., 2009a, Zhang et al., 2013), the Young’s modulus for the brain parenchyma and the tumour was set to 3000 Pa and 6000 Pa respectively. Because the brain tissue is almost incompressible (Miller, 2011, Zhang et al., 2013), a Poisson’s ratio of 0.49 was assigned for both parenchyma and tumour. The ventricles are modelled as a cavity as the cerebrospinal fluid can freely move in and out of them. The skull is assumed to be rigid and the interaction between skull and brain is modelled as finite sliding, frictionless contact. Displacements, measured intra-operatively, are applied on the brain surface exposed by craniotomy. The brain model is discretised with 707 nodes, and 4988 integration points were created from a triangular background grid with four integration points per cell. A constant influence domain ($R^2=8$) and same weights for the additional constraints ($\mu=10^{-7}$) were used in the meshless computation. For easy imposition of the essential boundary conditions, a regularized weight function (Most and Bucher, 2005) was used which possesses almost interpolating properties.

The solutions obtained using the MMs are compared with a reference solution obtained using ABAQUS. High order plain strain elements with hybrid formulation are used in ABAQUS to handle the incompressibility of the soft tissues. The constitutive material laws (Neo-Hookean), loading and boundary conditions are identical in both meshless and ABAQUS computations. The
differences of the computed deformation field between classical MLS and modified MLS in comparison with ABAQUS are shown in Figure 3.4. Numerical details of the comparison are presented in.

Table 3.2: Numerical details of comparison for the brain shift simulation (Figure 3.4).

<table>
<thead>
<tr>
<th>Approximation method</th>
<th>Nodes</th>
<th>Elements (ABAQUS)</th>
<th>Integration points (Meshless)</th>
<th>Average difference (mm)</th>
<th>Maximum difference (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) Classical MLS (Linear Basis)</td>
<td>707</td>
<td>1247</td>
<td>4988</td>
<td>0.14509</td>
<td>0.67531</td>
</tr>
<tr>
<td>b) MMLS ($\mu = 10^{-10}$) (Quadratic Basis)</td>
<td>707</td>
<td>1247</td>
<td>4988</td>
<td>0.12332</td>
<td>0.50729</td>
</tr>
<tr>
<td>c) Classical MLS (Quadratic Basis)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For the given support domain radius, the classic MLS with quadratic basis also failed due to the singularity of moment matrix, whereas the modified MLS with quadratic basis had no problem in computing the shape functions. As shown in Table 3, the maximum and average differences between MMLS and ABAQUS are found to be lower compared to those between classic MLS with linear basis and ABAQUS.
Figure 3.4: Differences of computed deformation fields in the brain: a) between classical MLS (linear basis, $R=8$) and ABAQUS, b) between MMLS ($R=8$, $\mu=10^{-7}$) and ABAQUS.
3.4.3 Unconstrained compression of a Cube

In this experiment an unconstrained compression of a cube with an edge length of 0.1 m is simulated. The meshless discretization of the problem domain and the boundary conditions are shown in Figure 3.5. 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 0.02m is applied to the top surface. The chosen deformation mode and material model result in uniform strains in the cube, and the finite element solution should be exact and independent of discretisation.

For the meshless simulations, the cube is discretised using 260 nodes and 5070 integration points are created using tetrahedral background integration cells with five integration points per tetrahedron. A constant influence domain radius \( R = 0.0325 \) is used for all nodes. The same weights were used for all the additional MMLS constraints \( \mu = 10^{-7} \). The essential boundary conditions are imposed by coupling the finite element method (FEM) and MLS shape functions near the essential boundaries. In this way, like in FEM, the prescribed displacement values can be directly imposed to the field variables on the essential boundaries (Belytschko et al., 1995). The meshless simulations are performed using the Meshless Total Lagrangian Explicit Dynamics (MTLED) algorithm, with dynamic relaxation used in order to ensure fast convergence to the steady state solution (Joldes et al., 2009a, Joldes et al., 2011).
Figure 3.5: The discretization (260 nodes, 1014 integration cells) and the boundary conditions for the cube used in the unconstrained compression simulation.

The obtained meshless simulation results are compared with the finite element solution obtained using the ABAQUS static solver and the same discretization. The average and maximum differences between the nodal displacements obtained using the classical MLS and MMLS meshless methods and the reference solution obtained using ABAQUS are presented in Table 3.3 and Figure 3.6. Similar to the previous examples, MMLS outperforms classical MLS in terms of solution accuracy.
Figure 3.6: a) Differences of computed deformation fields between classical MLS (linear basis, $R=0.0325$) and ABAQUS. b) Differences of computed deformation fields between MMLS ($R=0.0325, \mu=10^{-7}$) and ABAQUS. All dimensions are in meters.
Table 3.3: Difference in nodal displacements between the meshless and ABAQUS solutions for the cube compression simulation.

<table>
<thead>
<tr>
<th>Approximation method</th>
<th>Average difference (m)</th>
<th>Maximum difference (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classical MLS (Linear basis)</td>
<td>2.48 x 10^{-5}</td>
<td>9.70 x 10^{-5}</td>
</tr>
<tr>
<td>Modified MLS (Quadratic basis)</td>
<td>1.65 x 10^{-5}</td>
<td>5.35 x 10^{-5}</td>
</tr>
<tr>
<td>Classical MLS (Quadratic basis)</td>
<td>Fails to compute due to singular moment matrices</td>
<td></td>
</tr>
</tbody>
</table>

3.5 Concluding Remarks

Shape functions used in meshless methods are required to be sufficiently robust for reasonably arbitrarily distributed nodes. They should also be numerically stable, have a certain order of consistency, compactly supported, computationally efficient and ideally possess the Kronecker delta function property. In EFG, Moving Least Squares (MLS) shape functions are preferred due to its continuity and smoothness properties. However, the classical MLS places strict requirements on the nodal distributions inside each support domain to compute the shape functions. Therefore, the practical use of higher order polynomial basis, which has the capability to produce more accurate approximations of complex deformation fields, is problematic to implement in classical MLS for randomly distributed nodes. In this context, the Modified MLS (MMLS) approximation allows higher order polynomial base functions to be used under the same conditions as lower degree base functions. This is achieved by augmenting the error functional used in the derivation of the MLS shape functions with additional terms based on the coefficients of the polynomial base functions. These additional constraints are small positive weights parameter and it has been observed that if the value of the positive weight parameter is decreased, the approximation accuracy approaches the accuracy of classical MLS with quadratic base functions.
In this chapter, the MMLS approximation is further developed to be used in 3D and implemented in an Element Free Galerkin method for predicting soft tissue deformation. The MTLED algorithm using the MMLS shape functions was used for the simulation of extension of a soft tissue sample, a craniotomy induced brain deformation and an unconstrained compression of cube having the material properties of soft tissues. The results are compared with results obtained using the commercial finite element software ABAQUS. The 2D and 3D numerical examples demonstrate that MMLS produces more accurate results than classical MLS with linear basis, for the same support domain size. This behaviour stems from the fact that MMLS has better approximation capability compared to classical MLS with linear basis, therefore reducing the discretisation error. For the chosen support domain size, the classic MLS with quadratic basis fails due to singular moment matrices. To further increase the accuracy of the approximation, the method can be extended to use higher order base functions in a similar fashion, by including in the error functional additional constraints on the coefficients corresponding to the higher degree monomials in the basis.
Chapter 4

Essential Boundary Conditions Imposition in
Explicit Meshless Method
Most of the contents of this chapter have been published on:

4.1 Introduction

Despite the rapid advancement of Meshless Methods (MM) in recent years, some aspects still require further research. One such aspect is the imposition of Essential Boundary Conditions (EBC) in Galerkin-based MMs such as the Element Free Galerkin (EFG) method (Belytschko et al., 1994, Li and Liu, 2004, Liu, 2010). The difficulty in imposing EBC in MMs arises due to 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 the differential equations and boundary conditions. In computation, using shape functions, the values of the field variable that are computed at the nodes are used to approximate the values at non-nodal points by interpolating the nodal values. In FEM, the shape functions are interpolating functions by nature and the values of the field variables are explicitly calculated at nodes. Thus, prescribing certain values to the field variables on the essential boundary nodes of the problem domain is sufficient to enforce EBC. However, in MMs, the shape functions are created with overlapping support domains and in many cases are generally not interpolating at nodes. Thus prescribing certain values of the field variables will not yield the exact nodal displacements on the boundary nodes (Liu, 2010). For example, the Moving Least Squares (MLS) (Shepard, 1968, Lancaster and Salkauskas, 1981) 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 other nodes. MLS shape functions are non-polynomial and non-interpolating rational functions and imposing essential boundary conditions is not as trivial as in the FEMs.

Many techniques, therefore, have been developed over the years to overcome the difficulties associated in imposing the EBC in MMs (Fernández-Méndez and Huerta, 2004, Li and Liu, 2004, Liu and Gu, 2005). These methods can be broadly categorized into two main groups: i) modifying weak form, and ii) modifying shape functions.
The general approach of the first group is to operate on the variational or weighted residual formulation of the problem. Among these approaches, Lagrange multiplier and Penalty method are found to be some of the most widely used ones. For example, Belytschko et al. (1994) used Lagrange multiplier method by altering the variational statement to enforce EBC. In this method, additional unknowns, called Lagrange multipliers, are introduced with the trial functions to the Galerkin weak form. As the trial functions do not satisfy the essential boundary conditions, they are imposed by the Lagrange multipliers. Then in order to obtain the discretized formulation, these unknown Lagrange multipliers functions are interpolated using their nodal values and shape functions for nodes on the essential boundaries. Similarly, in case of Penalty method (Zhu and Atluri, 1998), a penalty formulation is devised in order to account for the EBC by the standard principle of minimum potential energy. The penalty formulation includes a diagonal matrix of penalty factors which are generally scalar parameters, and, typically, large values of the penalty factors are required in order to enforce EBC effectively.

The second group of methods to impose essential boundary condition is by modifying the shape functions. In some of the techniques, modifications are done in a way that makes the shape function of MMs almost interpolating by nature. Therefore, EBC can be directly imposed. For example, the use of singular weight functions (Lancaster and Salkauskas, 1981), and regularized weight functions (Most and Bucher, 2005) to enforce almost interpolating properties of shape functions at nodes have been presented in earlier research. Furthermore, the modification of shape functions can also be done in a way to couple them with finite element interpolation near the essential boundary (Krongauz and Belytschko, 1996, Belytschko et al., 1995c, Berger et al., 2008, Zhang et al., 2014). In this case, a string of finite elements along the essential boundaries are created and the shape functions from the finite elements are combined with shape functions of the meshless method so that the essential boundary conditions can be imposed as in FEM. There is a buffer zone between finite element region and meshless region, which is connected via the so-called ramp functions (Li and Liu, 2004).
However, most of these aforementioned methods have their applicability limited only to implicit way of solving the dynamic simulations. In contrast to explicit integrations, implicit method in time domain rely on solving systems of algebraic equations and need to perform iterations at every time step (Bathe, 1996, Belytschko. et al., 2000). Therefore, due to its low computational cost for each time step and also straightforward treatment of non-linearities (Belytschko. et al., 2000), explicit methods have become the preferred choice for many problems which involves large deformations and large strains with geometric and material non-linearities (Miller, 2011, Miller et al., 2006, Wittek et al., 2011). The Meshless Total Lagrangian Explicit Dynamics (MTLED) (Horton et al., 2010) is one of such methods which has already been established as a robust and efficient method in predicting large deformations in cases like soft tissues (Li et al., 2016, Miller et al., 2012). However, traditional methods of imposing EBC are not applicable in MTLED due to the complexities associated with solving for the additional unknowns like Lagrange multiplier or applying penalty methods. Furthermore, modifying shape functions to incorporate almost interpolating properties exhibits serious drawbacks when used with explicit integration, which will be explained in the following section. Therefore, imposing EBC in explicit time integration presents itself as a challenge.

In this chapter, a new approach in implementing the essential boundary conditions in EFG based MTLED framework is proposed. The new method is called Essential Boundary Conditions Imposition for Explicit Meshless (EBCIEM). The chapter is organized as follows: the motivation of EBCIEM in the context of explicit integration is discussed in the next section; the methodology of EBCIEM is presented in section 4.3; numerical examples are presented in section 4.4; followed by concluding remarks in section 4.5.
4.2 Explicit Time Integration and Scope for a New Method

In this section, the shortcomings of existing methods of imposing EBC in the context of explicit time integration are identified.

Explicit time integration is a direct integration method of the equations of dynamics in time domain where nodal accelerations are found directly without any iteration and then integrated to obtain the displacements (Bathe, 1996, Belytschko. et al., 2000). There is no necessity in assembling a global stiffness matrix. The equations of motion and an explicit integration scheme (i.e. central difference) represent a system of equations which is decoupled by using a diagonalized mass matrix. As it was discussed earlier, in the methods of imposing EBC by Lagrange multiplier and Penalty method, additional unknowns are introduced in the Galerkin Weak form. Consequently, the addition of these extra unknowns would result in a system of equations that cannot be decoupled, and, therefore, could not be solved efficiently within explicit time integration framework.

The coupled Finite Element and meshless approach probably is the most efficient and accurate method for imposing essential boundary conditions in the explicit time integration framework (Zhang et al., 2014, Belytschko et al., 1995c, Krongauz and Belytschko, 1996, Berger et al., 2008). However, in this case, the necessity of creating a finite element layer along the boundary as well as in its neighborhood of the boundary could become cumbersome and costly.

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 included in order to obtain stable simulation results. Therefore, it requires an estimation of the maximum stable time step (Bathe, 1996). According to Joldes et al. (2012), the stable critical time step for central difference integration can be estimated as:
\[
\Delta t_{\text{crit}} = \frac{2}{\omega_{\text{max}}} = \frac{2}{\sqrt{\lambda_{\text{max}}}} \approx \min_{l} \left( \frac{2}{\sqrt{\lambda_{\text{Max}}}} \right) \quad (4.1)
\]

where \(\omega_{\text{max}}\) is the maximum frequency of free vibration. The bound for the maximum eigenvalue is given by:

\[
\lambda_{\text{Max}}^l \leq \frac{N^l}{\rho^l} (\lambda + 2\mu). ||B^l||_F^2 = N^l c^{12}. B^l_j B^l_j \quad (4.2)
\]

where \(\rho\) is the material density and \(c\) is the dilatational wave speed and \(B\) is the matrix of Shape Function Derivatives (SFD).

As can be seen from Eq (4.1) and Eq (4.2), the stable critical time step is heavily influenced by the values of SFD. After the discretization, if the nodes are too close to each other, the SFD increases as the slope of the shape function becomes very steep, as shown in Figure 4.1. This would affect the estimated stable critical time-step leading to a very small value, which results in a large number of steps in the simulation, rendering the method impractical.

![Figure 4.1: Comparison between classical MLS and MLS with almost interpolating property.](image)
In the next section, the methodology of Essential Boundary Conditions Imposition for Explicit Meshless (EBCIEM) will be discussed.

### 4.3 Derivation of the Essential Boundary Conditions Imposition for Explicit Meshless (EBCIEM)

The derivation is started from the traditional global system of discretized damped equation of motion (Horton et al., 2010, Miller et al., 2006):

\[
\mathbf{M} \ddot{\mathbf{u}} + \mathbf{F}_d + \mathbf{F}_f = \mathbf{F}_e \tag{4.3}
\]

where \( \mathbf{u} \) is the vector of nodal displacements, \( \mathbf{M} \) is the mass matrix, \( \mathbf{F}_d \) is the global nodal reaction force vector, \( \mathbf{F}_f \) is the damping force, and \( \mathbf{F}_e \) is the vector of externally applied force at time \( t \). For simplicity, the method is derived in 2D. The notations from Bathe (Bathe, 1996) have been used here.

The externally applied force can be considered to compose of two parts:

\[
\mathbf{F}_{ext} = \mathbf{F}_{ne} + \mathbf{F}_e \tag{4.4}
\]

where \( \mathbf{F}_e \) is the force that is externally applied only on the essential boundary, and \( \mathbf{F}_{ne} \) is the force that is applied on the non-essential boundary. Replacing Eq(4.4) in Eq(4.3) and rearranging:

\[
\mathbf{M} \ddot{\mathbf{u}} + \mathbf{F}_d = (\mathbf{F}_{ne} - \mathbf{F}_f) + \mathbf{F}_e \tag{4.5}
\]

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 (Joldes
et al., 2009a). This force can be defined as \( \mathbf{F}_{\text{damp}} = c\mathbf{M}\ddot{\mathbf{u}} \), where \( c \) is the damping coefficient.

Then Eq(4.5) becomes:

\[
\mathbf{M}\ddot{\mathbf{u}} + c\mathbf{M}\dot{\mathbf{u}} = (\mathbf{F}_{\text{ne}} - \mathbf{F}_{\text{int}}) + \mathbf{F}_e
\]  

(4.6)

In EFG method, \( \mathbf{F}_e \) is calculated as:

\[
\mathbf{F}_e = \int_{\Gamma_e} \mathbf{T}(s) \cdot \mathbf{d}r
\]  

(4.7)

where \( \mathbf{\Phi}(s) \) 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. \( \mathbf{T}(s) \) can be defined as an interpolation of the values of the distributed force at the essential boundary nodes:

\[
\mathbf{T}(s) = \sum_{k=1}^{n_e} \mathbf{N}(s) \cdot \mathbf{T}_e^k
\]  

(4.8)

where \( \mathbf{N}(s) \) 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 4.2.

![Figure 4.2: Definition of the distributed force on the essential boundary.](image-url)

Replacing Eq(4.8) in Eq(4.7):
\[ t\mathbf{F}_e = \int_{\Gamma_e} t\mathbf{\Phi}(s) \sum_{k=1}^{n_e} N^k(s) \cdot t\mathbf{T}_e^k \cdot d\Gamma_e \] (4.9)

Eq(4.9) needs to be integrated numerically. If Gaussian quadrature over the essential boundary is used, Eq(4.9) becomes:

\[ t\mathbf{F}_e = \sum_{i=1}^{n_g} t\mathbf{\Phi}(s_i) \sum_{k=1}^{n_e} N^k(s_i) \cdot t\mathbf{T}_e^k \cdot w_i \] (4.10)

where \( x_i \) 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(4.10) can be written in matrix form as:

\[ t\mathbf{F}_e = t\mathbf{V} \cdot t\mathbf{T}_e \] (4.11)

with,

\[ t\mathbf{V}_{jk} = \sum_{i=1}^{n_g} t\mathbf{\Phi}_j(s_i) \cdot N^k(s_i) \cdot w_i \] (4.12)

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, the finite element shape functions are used in Eq(4.8).

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

\[ t\ddot{u} = (t^{+1/2}\dot{u} - t^{-1/2}\dot{u})/h \] (4.13)
\[ t^{-1/2} \mathbf{u} = (t \mathbf{u} - t^{-1} \mathbf{u})/h \quad (4.14) \]

and,

\[ t^{+1/2} \mathbf{u} = (t^{+1} \mathbf{u} - t \mathbf{u})/h \quad (4.15) \]

where \( h \) is a fixed time increment, \( t \) indicates \( t \)th time increment and the superimposed dot represents time derivative. The velocity at time \( t \) is computed as:

\[ t \mathbf{u} = (t^{+1/2} \mathbf{u} + t^{-1/2} \mathbf{u})/2 \quad (4.16) \]

By substituting the above expression for acceleration and velocity in the equation of the motion Eq(4.6) and rearranging:

\[ t^{+1} \mathbf{u} - \alpha \mathbf{M}^{-1} t \mathbf{F}_e = t \mathbf{u} + \beta (t \mathbf{u} - t^{-1} \mathbf{u}) + \alpha \mathbf{M}^{-1} (t \mathbf{F}_{ne} - t \mathbf{F}_{int}) \quad (4.17) \]

where \( \alpha = 2h^2/(2 + ch) \) and \( \beta = (2 - ch)/(2 + ch) \). Eq(4.17) is explicit as long as the mass matrix \( \mathbf{M} \) is diagonal. The right hand side of Eq(4.17) can be identified as the predicted displacement \( \mathbf{u} \) when the load on the essential boundary is ignored:

\[ t^{+1} \mathbf{u} = t \mathbf{u} + \beta (t \mathbf{u} - t^{-1} \mathbf{u}) + \alpha \mathbf{M}^{-1} (t \mathbf{F}_{ne} - t \mathbf{F}_{int}) \quad (4.18) \]

The system of Eq(4.17) is augmented with the equations for the imposed displacement \( (t^{+1} \mathbf{u}) \), defined as the interpolation of the field variable displacements at time \( t + 1 \).

\[ \Phi. t^{+1} \mathbf{u} = t^{+1} \mathbf{u} \quad (4.19) \]

Substituting Eq(4.11) into Eq(4.17) and combining Eq(4.17) and Eq(4.19):
\[
\begin{bmatrix}
1 & -\alpha \mathbf{M}^{-1} \cdot \mathbf{tV} \\
\Phi & 0
\end{bmatrix}
\begin{bmatrix}
t^{+1}\mathbf{u} \\
t^{+1}\mathbf{T}_e
\end{bmatrix} =
\begin{bmatrix}
t^{+1}\mathbf{u} \\
t^{+1}\mathbf{u}
\end{bmatrix}
\] (4.20)

Because the number of degree of freedom on the essential boundary is equal to the number of values in the vector used in the definition of distributed force on the essential boundary, the number of equations in (4.20) is equal to the number of unknowns. Applying static condensation to Eq(4.20) to eliminate \( t^{+1}\mathbf{T}_e \) yields:

\[
t^{+1}\mathbf{u} = t^{+1}\mathbf{u} + \mathbf{M}^{-1} \cdot \mathbf{tV} \left[ \Phi \cdot \mathbf{M}^{-1} \cdot \mathbf{tV} \right]^{-1} \left[ t^{+1}\mathbf{u} - \Phi \cdot t^{+1}\mathbf{u} \right]
\] (4.21)

Eq(4.21) can be written as,

\[
t^{+1}\mathbf{u} = t^{+1}\mathbf{u} + t^{+1}\mathbf{u}_{\text{correction}}
\] (4.22)

where,

\[
t^{+1}\mathbf{u}_{\text{correction}} = t^{+1}\mathbf{P} \cdot \left[ t^{+1}\mathbf{u} - \Phi \cdot t^{+1}\mathbf{u} \right]
\] (4.23)

with,

\[
t^{+1}\mathbf{P} = \mathbf{M}^{-1} \cdot \mathbf{tV} \left[ \Phi \cdot \mathbf{M}^{-1} \cdot \mathbf{tV} \right]^{-1}
\] (4.24)

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 \( t^{+1}\mathbf{V} \) using Eq(4.12);
2. Calculation of matrix \( t^{+1}\mathbf{P} \) using Eq(4.24);
3. Computation of the displacement correction using Eq(4.23) and its application using Eq(4.22).
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 $^tP$, 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(4.22), a displacement correction is added at every time step to the displacement field, enforcing the EBC in a prediction-correction fashion.

4.3.1 A Simplified Essential Boundary Conditions Imposition for Explicit Meshless (SEBCIEM)

The presented EBCIEM method requires computation of integrands on the essential boundary, which requires a discretization for numerical integration. If the distributed force on the essential boundary is lumped at the essential boundary nodes, a simplified version of the method can be obtained. In this case, Eq(4.8) becomes:

$$^tT(s) = \sum_{k=1}^{n_e} \delta(s_k) \cdot T^k_e \quad (4.25)$$

where $\delta$ is the Dirac Delta function. $\delta(s_k)$ can be treated as being zero everywhere except at the essential boundary nodes ($s_k$) and have unit area (Greenberg, 1998). Replacing Eq(4.25) in Eq(4.7):

$$^tF_e = \int_{\Gamma^e} ^t\Phi \cdot \sum_{k=1}^{n_e} \delta(s_k) \cdot T^k_e \cdot d\Gamma_e \quad (4.26)$$

As, by definition (Greenberg, 1998):

$$\int_{\Gamma^e} f(s) \delta(s_k) d\Gamma_e = f(s_k) \quad (4.27)$$
then (4.26) becomes:

\[ \mathbf{tF_e} = \sum_{k=1}^{n_e} \mathbf{t} \Phi (s_k) \cdot \mathbf{T_e^k} \]  

Eq(4.28) can be written in matrix form as:

\[ \mathbf{tF_e} = \mathbf{tV} \cdot \mathbf{T_e} \]  

(4.29)

with,

\[ \mathbf{tV}_{jk} = \mathbf{t} \Phi_j (s_k) \]  

(4.30)

As can be seen from Eq(4.28), it is not necessary to define additional quadrature points along the essential boundary to evaluate \( \mathbf{tF_e} \) in SEBCIEM. The procedures to obtain the constant matrix (\( \mathbf{P} \)) and displacement corrections (\( \mathbf{u_{correction}} \)) are similar to as presented earlier.

### 4.4 Numerical Examples

In this section, overall five cases of biomechanics applications are presented to test the performance of EBCIEM and SEBCIEM. The simulation examples presented in the following sub-sections are based on Meshless Total Lagrangian Explicit Dynamics (MTLED) algorithm (Horton et al., 2010, Miller et al., 2006). 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 discretization 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 (this approach is called in this thesis as the \textit{correction method}). In sections 4.4.1 - 4.4.4, simulation results obtained by using the proposed methods for imposing EBC are compared with the results obtained using the correction method; in this way it would be possible to determine how large the errors are in imposing EBC in the case of using the simple correction method and to justify the need for the proposed methods. In section 4.4.5, the EBCIEM and SEBCIEM results are compared with the results obtained by the commercial finite element software ABAQUS. In each case of comparison, identical geometry and material configurations are used. In order to ensure fast and accurate convergence of the explicit central difference method, dynamic relaxation is used to reach the steady state solution (Joldes et al., 2009a, Joldes et al., 2011).

4.4.1 \textit{Extension of a Soft Tissue Sample in 2D}

This simple example is used 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 discretized 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). According to Eq(4.10) a 1-D interpolating shape function is used along the essential boundary and a 3-point Gauss quadrature is used for each essential boundary arc-segment to integrate the additional applied traction force, as shown in Figure 4.3. The sparse constant matrix \( {}^t\mathbf{P} \) in Eq(4.24) is precomputed as shown in Figure 4.4(b). Therefore, multiplications of this constant matrix \((n \times n_e)\) with the difference of the imposed and classical interpolated displacement are the only additional computations that need to be executed during each the time-step.
Figure 4.3 : 2D soft tissue geometry. Dimensions are in centimetres. Total nodes: $n = 57$; Essential boundary nodes, $n_e = 10$ (Fixed nodes: 5 and displaced nodes: 5). For each segment, 3-point Gauss quadrature was used to evaluate Eq(4.10).

Figure 4.4 : a) Assembled $^tV$ matrix (57x10), b) Precomputed constant matrix $^tP$ (57x10) in Eq(4.24). Node numbers 1-5 represent fixed nodes, and 15-19 represent displaced nodes.

For the rest of the domain, 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 field functions. For each node, the radius of the influence domain is considered to be constant \((R = 2.2)\). The hyper-elastic Neo-Hookean material model is chosen to capture the behavior of soft tissues undergoing large deformation. The simulation results and numerical details are presented in Figure 4.5 and Table 4.1.

![Figure 4.5](image)

**Figure 4.5**: Extension of soft tissue sample in 2D with EBCIEM: deformed configuration. Dimensions are in centimetres.

**Table 4.1**: Comparison between the positions of EB nodes: *Soft-tissue extension in 2D*. The numbers reported in the table are shown for 4\(^{th}\) significant digit but in fact the EBC was imposed to machine precision \((-10^{15})\).

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

As can be seen from Table 4.1, EBCIEM imposes the essential boundary conditions exactly as prescribed.
4.4.2 Craniotomy Induced Brain Shift in 2D

In this example a craniotomy induced brain deformation is simulated in 2D. The brain parenchyma and the tumour are modelled as Neo-Hookean with Young’s modulus of 3000 Pa and 6000 Pa respectively. The Poisson’s ratio of 0.49 is chosen due to the incompressibility of the brain tissue. The skull is assumed to be rigid and the ventricles are modelled as a cavity. The interaction between skull and brain is modelled as finite sliding, frictionless contact. A variable load in terms of displacement is enforced on the essential boundary nodes of the brain surface. One-dimensional interpolating shape function is used along the essential boundary and a 3-point Gauss quadrature is used for each boundary segment to evaluate Eq(4.10). The brain model is discretised with 707 nodes, and 4988 integration points are created from 1247 triangular background integration cells with four integration points per cell. MLS shape functions with a constant influence domain ($R=10$) are used. The deformed configuration is shown in.

![Craniotomy induced brain shift in 2D with EBCIEM: deformed configuration.](image)

Figure 4.6 : Craniotomy induced brain shift in 2D with EBCIEM: deformed configuration.
The average and maximum differences between the final nodal displacements between EBCIEM and by the correction method are shown in Table 4.2. As can be seen from Table 4.2, with the prescribed displacements, EBCIEM imposes the essential boundary conditions exactly to machine precision.

Table 4.2: Comparison between the positions of essential boundary nodes: Craniotomy induced brain shift in 2D.

<table>
<thead>
<tr>
<th>Essential Boundary Nodes</th>
<th>Initial Position (mm)</th>
<th>Imposed Displacement (mm)</th>
<th>Deformed Position (EBCIEM) (mm)</th>
<th>Deformed Position (Correction method) (mm)</th>
<th>Average Difference (mm)</th>
<th>Maximum Difference (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node ID</td>
<td>x</td>
<td>y</td>
<td>x</td>
<td>y</td>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>1</td>
<td>0.4937</td>
<td>88.9587</td>
<td>0</td>
<td>0</td>
<td>0.4937</td>
<td>88.9587</td>
</tr>
<tr>
<td>2</td>
<td>-3.8035</td>
<td>86.4637</td>
<td>0.5</td>
<td>-1</td>
<td>-3.3035</td>
<td>85.4637</td>
</tr>
<tr>
<td>3</td>
<td>-8.0646</td>
<td>83.9021</td>
<td>1</td>
<td>-2</td>
<td>-7.0646</td>
<td>81.9021</td>
</tr>
<tr>
<td>4</td>
<td>-12.6156</td>
<td>81.9091</td>
<td>1.5</td>
<td>-3</td>
<td>-11.1156</td>
<td>78.9091</td>
</tr>
<tr>
<td>5</td>
<td>-16.8794</td>
<td>79.3545</td>
<td>2</td>
<td>-4</td>
<td>-14.8794</td>
<td>75.3545</td>
</tr>
<tr>
<td>6</td>
<td>-21.2002</td>
<td>76.9217</td>
<td>2.5</td>
<td>-5</td>
<td>-18.7002</td>
<td>71.9217</td>
</tr>
<tr>
<td>7</td>
<td>-25.6497</td>
<td>74.6095</td>
<td>3</td>
<td>-6</td>
<td>-22.6497</td>
<td>68.6095</td>
</tr>
<tr>
<td>8</td>
<td>-29.1423</td>
<td>71.1214</td>
<td>3.5</td>
<td>-7</td>
<td>-25.1423</td>
<td>64.1214</td>
</tr>
<tr>
<td>9</td>
<td>-31.9816</td>
<td>67.0233</td>
<td>3.5</td>
<td>-7</td>
<td>-28.9816</td>
<td>60.0233</td>
</tr>
<tr>
<td>10</td>
<td>-34.7779</td>
<td>62.9365</td>
<td>3</td>
<td>-6</td>
<td>-31.7779</td>
<td>56.9365</td>
</tr>
<tr>
<td>11</td>
<td>-37.1032</td>
<td>58.5447</td>
<td>2.5</td>
<td>-5</td>
<td>-34.1032</td>
<td>53.5447</td>
</tr>
<tr>
<td>12</td>
<td>-39.2232</td>
<td>54.0447</td>
<td>2</td>
<td>-4</td>
<td>-37.2232</td>
<td>50.0447</td>
</tr>
<tr>
<td>13</td>
<td>-41.6898</td>
<td>49.7299</td>
<td>1.5</td>
<td>-3</td>
<td>-40.6898</td>
<td>46.7299</td>
</tr>
<tr>
<td>14</td>
<td>-44.2294</td>
<td>45.4555</td>
<td>1</td>
<td>-2</td>
<td>-43.2294</td>
<td>43.4555</td>
</tr>
<tr>
<td>15</td>
<td>-46.3662</td>
<td>40.9662</td>
<td>0.5</td>
<td>-1</td>
<td>-45.3662</td>
<td>39.9662</td>
</tr>
<tr>
<td>16</td>
<td>-47.770</td>
<td>36.1992</td>
<td>0</td>
<td>0</td>
<td>-47.770</td>
<td>36.1992</td>
</tr>
</tbody>
</table>

4.4.3 Compression of Soft Tissue Sample in 3D

In this example, the application of EBCIEM is shown for a 3D case. The effects of simplification introduced for SEBCIEM is also demonstrated.

In this case, the deformation of a cylinder having the material properties of soft tissue is simulated. The cylinder used in the experiments, shown in Figure 4.7(a), has a height of 0.1 m and radius of 0.05 m. One face (z = 0 m) of the cylinder is rigidly constrained while the opposite face (z = 0.1 m) is displaced. A maximum compressive displacement loading of 0.02m is applied smoothly. The hyper-elastic Neo-Hookean material model with Young’s modulus of 3000 Pa, Poisson’s ratio of 0.49 and density of 1000 kg/m$^3$ are chosen to capture the behaviour of nearly incompressible continuum of soft tissues. The cylinder is discretized 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 (Chowdhury et al., 2015, Joldes et al., 2015a). 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 4.7](image)

**Figure 4.7**: 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, shown in Figure 4.8. 2D interpolating shape function and 3-point Gauss quadrature is used in each triangle to integrate the externally applied force on the essential boundary \((\mathbf{F}_e)\). Figure 4.9 shows the assembled \(\mathbf{V}\) and \(\mathbf{P}\) matrices as defined in Eq (4.24). Figure 4.7(b) shows the deformed configuration of the test cylinder using EBCIEM method and Figure 4.10 shows the differences of the computed deformation field for the entire domain between EBCIEM and correction method. In case of SEBCIEM, the externally applied force on essential boundary \((\mathbf{F}_e)\) is evaluated using Eq (4.28). No surface triangulation is necessary in this case. Figure 4.11 and Figure 4.12 show the differences of the computed deformation field for the entire domain between SEBCIEM and correction method and EBCIEM and SEBCIEM respectively. Table 4.3 shows the average and maximum differences of the computed final positions of the essential boundary nodes.
Figure 4.8: Triangulated essential boundary surface (shown for fixed nodes only). 3-point Gauss quadrature used for each triangle.

Figure 4.9: a) Assembled $^tV$ matrix (1252x192), b) Precomputed constant matrix $^tP$ (1252x192).
Figure 4.10: Compression of 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.

<table>
<thead>
<tr>
<th>Average Difference (mm)</th>
<th>Maximum Difference (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.20507</td>
<td>1.1809</td>
</tr>
</tbody>
</table>

Figure 4.11: Compression of 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.

<table>
<thead>
<tr>
<th>Average Difference (mm)</th>
<th>Maximum Difference (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.21928</td>
<td>1.2883</td>
</tr>
</tbody>
</table>
Table 4.3: Compression of Soft Tissue Sample in 3D: Comparison of average and maximum differences of the final positions of the EB nodes obtained using EBCIEM, SEBCIEM and Correction method.

<table>
<thead>
<tr>
<th>EB Nodes</th>
<th>Methods</th>
<th>Average Difference (mm)</th>
<th>Maximum Difference (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed Nodes</td>
<td>EBCIEM and Correction Method</td>
<td>0.27056</td>
<td>0.74101</td>
</tr>
<tr>
<td></td>
<td>SEBCIEM and Correction Method</td>
<td>0.27056</td>
<td>0.74101</td>
</tr>
<tr>
<td></td>
<td>EBCIEM and SEBCIEM</td>
<td>1.5138 x 10^{-14}</td>
<td>6.8982 x 10^{-14}</td>
</tr>
<tr>
<td>Displaced Nodes</td>
<td>EBCIEM and Correction Method</td>
<td>0.27607</td>
<td>0.81965</td>
</tr>
<tr>
<td></td>
<td>SEBCIEM and Correction Method</td>
<td>0.27607</td>
<td>0.81965</td>
</tr>
<tr>
<td></td>
<td>EBCIEM and SEBCIEM</td>
<td>8.4023 x 10^{-14}</td>
<td>2.3972 x 10^{-13}</td>
</tr>
</tbody>
</table>

It was found that both EBCIEM and SEBCIEM also impose the essential boundary conditions exactly to machine precision as in 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.
4.4.4 Extension of Soft Tissue Sample in 3D

The purpose of this example is to demonstrate the application of the new method on a set of essential boundary nodes that do not form a surface. In this case, the test case presented in section 0 is reconsidered. However, an extension displacement loading of 0.02 m is applied only to the edge line of the top surface \((z = 0.1 \text{ m})\), as shown in Figure 4.13(a). The material model, shape functions and numerical integration scheme are the same as described in section 0. In this case, SEBCIEM is used and the externally applied force on essential boundary \((^t \mathbf{F}_e)\) is evaluated using Eq(4.28). The results obtained with SEBCIEM are compared to those obtained with the correction method. Figure 4.14 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 comparison values. Furthermore, Table 4.4 shows the average and maximum differences of the final positions of the EB nodes obtained with the two methods.

![Figure 4.13 : Extension of a cylinder with the material properties of soft-tissue: a) undeformed configuration, b) deformed configuration.](image)
Table 4.4 : Extension of 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.

<table>
<thead>
<tr>
<th>EB Nodes</th>
<th>Average Difference (mm)</th>
<th>Maximum Difference (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed Nodes</td>
<td>0.0886</td>
<td>0.3004</td>
</tr>
<tr>
<td>Displaced Nodes</td>
<td>3.0385</td>
<td>5.8417</td>
</tr>
</tbody>
</table>

The results demonstrate the effectiveness of SEBCIEM in imposing EBC in applications where discretization along the essential boundary is difficult to achieve.
4.4.5 Unconstrained Compression of a Cube

In this example the unconstrained compression of a cube with an edge length of 0.1 m is simulated. For the meshless simulations, the cube is discretized using 260 nodes and 4056 integration points are created using a consistent tetrahedral background mesh with four integration points per tetrahedron. MMLS with quartic spline weight function is used to approximate the unknown field functions (Joldes et al., 2015a, Chowdhury et al., 2015). 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 4.15.

The meshless simulation results are obtained with both EBCIEM and SEBCIEM and compared 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 discretization. Like any FE approach, ABAQUS imposes the EBC exactly as prescribed due to the inherent interpolating property (i.e. Kronecker delta) of the finite element shape functions.
Figure 4.15: *Unconstrained Compression of a Cube*: The discretization (260 nodes, 1014 integration cells) and the boundary conditions.

Figure 4.16 shows the differences between the nodal displacements obtained using the meshless methods (EBCIEM and SEBCIEM) and the reference solution obtained using ABAQUS. Table 4.5 shows the differences of the nodal displacements obtained using EBCIEM, SEBCIEM and ABAQUS for the entire domain. The results demonstrate that the differences between the meshless (EBCIEM and SEBCIEM) and FE solution are found to be within machine precision.
Figure 4.16: Unconstrained Compression of a Cube: Differences of computed deformation field between: a) EBCIEM and ABAQUS, c) SEBCIEM and ABAQUS, and e) EBCIEM and SEBCIEM. Histogram of differences between: b) EBCIEM and ABAQUS, d) SEBCIEM and ABAQUS, and f) EBCIEM and SEBCIEM.
Table 4.5: *Unconstrained Compression of a Cube*: Comparison of average and maximum differences of the nodal displacements obtained using EBCIEM, SEBCIEM and ABAQUS.

<table>
<thead>
<tr>
<th>Methods to Compare</th>
<th>Average Difference (m)</th>
<th>Maximum Difference (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBCIEM and ABAQUS</td>
<td>$1.8691 \times 10^{-5}$</td>
<td>$5.8853 \times 10^{-5}$</td>
</tr>
<tr>
<td>SEBCIEM and ABAQUS</td>
<td>$4.6187 \times 10^{-5}$</td>
<td>$1.9240 \times 10^{-4}$</td>
</tr>
<tr>
<td>EBCIEM and SEBCIEM</td>
<td>$4.2372 \times 10^{-5}$</td>
<td>$1.8890 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

4.5 Concluding Remarks

Imposition of Essential Boundary Conditions (EBC) in Galerkin based MMs has always been a major issue due to the non-polynomial and non-interpolating nature of the meshless shape functions. Many techniques have been developed over the years to overcome the difficulties associated in imposing the EBC in MMs (Fernández-Méndez and Huerta, 2004, Li and Liu, 2004, Liu and Gu, 2005, Hosseini et al., 2017). These techniques can be broadly categorized into two groups: i) modifying Galerkin weak form, and ii) modifying shape functions (Sun et al., 2016, Krongauz and Belytschko, 1996, Belytschko et al., 1995c, Berger et al., 2008, Zhang et al., 2014, Sun and Wang, 2017). However, both of these strategies are only applicable to implicit way of solving the dynamic simulations. In explicit time integration, these traditional techniques becomes impractical to implement due to the complexities associated with solving for the additional unknowns like Lagrange multiplier or penalty methods, and also, for interpolating MLS shape functions, due to reason that the estimated stable critical time-step of explicit method is heavily affected if the nodes are too close to each other.

In this chapter, two new ways of imposing essential boundary conditions in meshless method based on Element Free Galerkin principle have been introduced. The motivations behind developing the new methods, called *Essential Boundary Conditions Imposition for Explicit...*
Meshless (EBCIEM) and Simplified Essential Boundary Conditions Imposition for Explicit Meshless (SEBCIEM), have been explained in Section 4.1 and Section 4.2 in the context of explicit time integration. The mathematical models for the two methods have been derived. The new methods consider the total externally applied force in the equation of motion into two parts and presents two approaches to evaluate the externally applied force on the essential boundary. In the first approach, called EBCIEM, the externally applied forces are considered as distributed force and 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, called 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 get the new displacement field.

The applicability of the new methods has been established in a Meshless Total Lagrangian and Explicit Time Integration framework. However, the applicability of the methods can be extended for Updated Lagrangian formulations as well, if the decreased computational performance is acceptable. The EBCIEM and SEBCIEM have been tested for five cases of biomechanics applications involving both 2D and 3D examples. In each case, it was found that both EBCIEM and SEBCIEM impose the essential boundary conditions to machine precision; 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(4.23), the increase in computation time due to the use of EBCIEM is less than 5% for this problem.
Chapter 5

Adaptive Numerical Integration in Meshless Total Lagrangian Explicit Dynamics
Most of the contents of this chapter have been published on:

5.1 Introduction

The Element Free Galerkin (EFG) methods offer an alternative solution method to well-established solution algorithms like Finite Element Method (FEM) for various classes of problems, as discussed in Chapter 1. EFG methods are based on global weak form and are part of the larger class of meshless methods that do not require a mesh for shape function definition. This makes EFG methods, such as the Meshless Total Lagrangian Explicit Dynamics (MTLED) algorithm, suitable in performing simulations of very large deformations in elasticity as there is no necessity of constructing a high quality mesh. However, one important weakness of EFG is the difficulty in performing numerical integration (Liu, 2010, Dolbow and Belytschko, 1999).

In the EFG based MMs, such as MTLED, Gaussian quadrature over a background mesh are generally used for numerical integration (Li and Liu, 2004, Liu, 2010, Horton et al., 2010). Numerical integration is a time-consuming process for the computation of stresses at each integration points in the Total Lagrangian (TL) formulation. It is crucial to have an accurate numerical integration scheme for predicting deformed geometries accurately. In FEM, the integration scheme is designed to integrate piecewise polynomial functions. In this case, the integration mesh is the same as the element mesh, and, in case of Updated Lagrangian formulation, the shape functions are polynomials over the integration cells. Therefore, it is possible to select a Gaussian quadrature which can exactly integrate the shape functions and their derivatives simply by knowing the degree of the polynomial shape function (note that in TL formulations, the integrands are usually non-polynomials despite the use of polynomial shape functions (Miller et al., 2006). Generally, neither of these two conditions is satisfied in MMs. The meshless shape functions (i.e. MLS, MMLS) are usually non-polynomial rational functions of the spatial coordinates. Therefore, choosing an order of Gaussian quadrature for performing exact integration of the shape function is not feasible in MMs. Furthermore, meshless shape functions usually have a much larger support domain compared to FEM and their local supports may not align with the integration cells. The shape functions are integrated without respect to their support boundaries. These are the two major
sources of error associated with the integration schemes used in EFG based MMs (Dolbow and Belytschko, 1999).

Many techniques have been developed over the years to try to create more accurate numerical integration schemes in meshless methods. Dolbow and Belytschko (1999) presented a technique where integration cells are aligned with the support domains of the shape functions. They show that using this ‘bounding-box’ technique the integration accuracy can be increased with Gauss quadrature. Atluri et al. (1999) and De et al. (2001) implemented specialized integration rules are presented for circular support domains or intersections of circular domains. An adaptive numerical integration scheme is presented by Racz and Bui (2012), which is based on mapping complex integration domains into such simpler domains. However, these types of specialized numerical integration rules are limited to specific types of support domains and also the treatment of irregular boundaries become cumbersome. Efforts have been also made to develop the EFG method without having to use a background mesh which is known as nodal integration (Puso et al., 2008). In this case, integrations are carried out by using the field nodes and without the use of additional Gauss points. In the Galerkin weak form, this technique can be applied to yield a particle-type method where stress and material history are located exclusively at the nodes and can be employed using meshless shape functions. A comparison between nodal and Gaussian integration is presented by Quak et al. (2011). However, nodal integration schemes do suffer from instability due to rank deficiency similar to under-integrated FE and may undergo significant loss in accuracy (Liu, 2010). In general, most of the numerical integration algorithms lack in estimating and controlling the desired integration accuracy.

In this context, Joldes et. al. (2015b) devised a new adaptive quadrature algorithm for EFG methods. The algorithm creates a distribution of integration points within the problem domain and allows the computation of integrals with controlled accuracy. The method introduces new integration points only in the areas where the integration accuracy is not sufficient. The method imposes no constraints on the type of support domains that can be used. Although the adaptive
integration procedure was found effective in solving 1D and 2D Boundary Value Problems (BVPs), the method was never used to solve 3D elasticity problems. In this chapter, the new adaptive integration procedure is further enhanced to use in the Meshless Total Lagrangian Explicit dynamics algorithm.

5.2 Adaptive Numerical Integration Algorithm for Total Lagrangian Explicit Dynamics

In the Total Lagrangian formulation, numerical integration is required to evaluate the global nodal reaction force vector which is defined as:

$$\mathbf{F}_0^T = \int_{V_0} \mathbf{B}_L^T \mathbf{S} \, dV_0$$  

with,

$$\mathbf{B}_L = [\mathbf{B}_L^{(1)}, \mathbf{B}_L^{(2)}, \ldots, \mathbf{B}_L^{(n)}]$$  

$$\mathbf{B}_L^{(i)} = \mathbf{B}_{L0}^{(i)} \mathbf{X}^T$$

where $\mathbf{B}_L$ is the full strain-displacement matrix, $\mathbf{S}$ is the second Piola–Kirchoff stress vector and $\mathbf{X}$ is the deformation gradient. $\mathbf{B}_{L0}$ is the matrix of shape function derivatives in reference to the initial configuration and has the following construction:

$$\mathbf{B}_{L0}^{(i)} = \begin{bmatrix} \partial_x & 0 & 0 \\ 0 & \partial_y & 0 \\ 0 & 0 & \partial_z \\ \partial_y & \partial_x & 0 \\ 0 & \partial_x & \partial_y \\ \partial_z & 0 & \partial_z \end{bmatrix}$$

where $\partial_x$, $\partial_y$ and $\partial_z$ are derivatives of shape functions with respect to x, y and z respectively and their values are taken directly from the precomputed MMLS shape function derivatives matrix $D\Phi(x)$. 


In the adaptive quadrature method (Joldes et al., 2015b), a function of ‘less smooth’ integrand needs to be defined which is integrated to a user-defined accuracy. This procedure generates a collection of integration points and weights over the integration cell. This ‘less smooth’ idea is based on the observation that if the adaptive integration procedure is able to accurately integrate a given integrand over the integration cell, it should also accurately integrate integrands that are 'smoother'. For example, as discussed by Joldes et al. (2015b), if function $f$ and $g$ are approximated by the polynomial of degree $n$ and $m$ respectively, then their product $fg$ can be approximated by a polynomial $p$ of degree $m+n$. Either $f^2$ and $g^2$ is less smooth than $fg$ as it requires for approximation of a polynomial of degree higher than the degree of polynomial $p$. Moreover, the degree of a polynomial approximation for $f^2+g^2$ is $\max(2m,2n)$. Following the observations and considering Eq(5.1)-Eq(5.4), the less smooth integrand is defined based on the MMLS shape function derivatives as:

$$f(x) = \left[ \sum_{k=1}^{n} (\phi^k_x(x))^2 \right]^{\frac{1}{2}}$$

After defining the less smooth integrand, the procedure for performing the adaptive numerical integration is applied in MTLED by the following algorithm:

**Step 1: Pre-computation of integration points ($x_i$) and associated weights ($w_i$):**
- Select desired accuracy ($\tau$)
- Select a quadrature rule to apply to the integration cells.
- For each initial integration cell, apply the recursive integration procedure for the function $f$ in Eq(5.5) used to drive the subdivisions:
  - Select a scheme to geometrically subdivide the initial integration cell ($cell$) into some number of subdivisions ($cell_1, cell_2...cell_n$).
  - Approximate the integral in the initial cell by the selected quadrature rule.
\[
Q, x_i, w_i = \text{integrate} (f, \text{cell})
\]

- Approximate the integrals in the subdivided cells.

\[
[Q_1, x_{i1}, w_{i1}] = \text{integrate} (f, \text{cell}_1)
\]

\[
[Q_2, x_{i2}, w_{i2}] = \text{integrate} (f, \text{cell}_2)
\]

\[
\vdots
\]

\[
[Q_n, x_{in}, w_{in}] = \text{integrate} (f, \text{cell}_n)
\]

- Find the error, \( \varepsilon = \frac{|Q - \sum_{i=1}^{n} Q_i|}{Q} \)

- If \( \varepsilon > \tau \), return:
  - \( Q = \sum_{i=1}^{n} Q_i \)
  - \( x_i = \text{concatenate} (x_{i1}, x_{i2}, \ldots, x_{in}) \)
  - \( w_i = \text{concatenate} (w_{i1}, w_{i2}, \ldots, w_{in}) \)

**Step 2:** Perform numerical integration

- Compute the global nodal reaction force vector over the problem domain by the integration points \( (x_i) \) and weights \( (w_i) \) determined in Step 1 as:

\[
\dot{\mathbf{F}}_0 = \sum_{i=1}^{N_{ip}} \left( \dot{\mathbf{x}}_i \dot{\mathbf{B}}_0^T \dot{\mathbf{S}} \right) \big|_{x=x_i} w_i
\]

where \( N_{ip} \) is the total number of integration points distributed in the problem domain. In the TL settings, there is no necessity to update the distribution of integration points as the MMLS shape functions and their derivatives do not change during the solution process.

### 5.3 Numerical Examples

In this section, 2D and 3D numerical examples are presented to demonstrate the applicability of the adaptive numerical integration procedure in MTLED algorithm. Four cases of biomechanics applications are presented: craniotomy induced brain shift in 2D, extension of a sheep brain sample in 3D and unconstrained compression of a cube.
In the adaptive integration procedure, for a given number of recursive levels, the initial integration domain is subdivided into a number of subdomains at each level. For the two dimensional case presented in this section, a triangular background integration cell is sub-divided into 4 cells by finding the midpoints of the edges of the original integration cell. An example of generated integration points for a 2D case is presented in Figure 5.1

![Integration Points in 2D](image)

**Figure 5.1**: Sample distribution of integration points within a 2D problem space using triangular background integration cells: a) without adaptive integration (using 4 Gauss points per triangle), b) with adaptive integration (using 4 subdivision of each triangular cell)

For the 3D cases presented in this section, the tetrahedral background integration cells are used. During the adaptive procedure, the integration cells are subdivided into either 2 or 4 or 8 subdomains. For the case where the original integration cell is subdivided into 2 subdomains, a new point (vertex corner) is introduced at the midpoint of the longest edge of the original tetrahedral cell. For 4 subdivisions, 3 new points are created at the midpoints of the three edges of the original tetrahedral cell. Similarly, midpoints of all the edges of the original tetrahedral integration cell are connected to create 8 subdivisions within the mother-cell. An example of generated integration points for a 3D case is presented in Figure 5.2.
Figure 5.2: Sample subdivision and integration points generation using adaptive procedure for a 3D case (tetrahedral background integration cell)
5.3.1 Craniotomy Induced Brain Shift in 2D

In this example, the craniotomy induced brain deformation is simulated in 2D to test the adaptive numerical integration procedure. The brain parenchyma and the tumour are modelled as Neo-Hookean with Young’s modulus of 3000 Pa and 6000 Pa respectively. The Poisson’s ratio of 0.49 is chosen due to the incompressibility of the brain tissue. The skull is assumed to be rigid and the ventricles are modelled as a cavity. The interaction between skull and brain is modelled as finite sliding, frictionless contact. A variable load in terms of displacement is enforced on the essential boundary nodes of the brain surface. One-dimensional interpolating shape function is used along the essential boundary and a 3-point Gauss quadrature is used for each boundary segment to evaluate the externally applied force on essential boundary. The brain model is discretized with 707 nodes, and 3741 integration points are created from 1247 triangular background integration cells with adaptive integration (0.1% desired accuracy). Figure 5.3 shows the distribution of integration points in the problem domain. MMLS shape functions with a constant influence domain ($R=8$) are used.

The deformed configuration is shown in Figure 5.4. The differences of the computed deformation field between meshless results obtained with MMLS ($\mu=10^{-7}$) and adaptive integration in comparison with ABAQUS are shown in Figure 5.4. Also, the meshless result obtained without applying the adaptive procedure is presented for comparison. Table 5.1 shows the differences in nodal displacements between meshless and ABAQUS solutions for the entire domain.
Figure 5.3: Distribution of 3741 integration points obtained with adaptive integration in a triangular background integration cells (0.1% desired accuracy): Craniotomy induced brain shift in 2D (Dimensions are in mm)

Table 5.1: Differences in nodal displacements between the meshless and ABAQUS solutions for the Craniotomy induced brain shift simulation in 2D. The model is shown in Figure 5.4.

<table>
<thead>
<tr>
<th>Approximation Method</th>
<th>Basis Function</th>
<th>Adaptive Integration</th>
<th>Integration Points</th>
<th>EBCIEM</th>
<th>Avg. Difference (mm)</th>
<th>Max. Difference (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modified MLS</td>
<td>Quadratic</td>
<td>No</td>
<td>4988</td>
<td>No</td>
<td>0.12332</td>
<td>0.50729</td>
</tr>
<tr>
<td>Modified MLS</td>
<td>Quadratic</td>
<td>Yes</td>
<td>3741</td>
<td>Yes</td>
<td>0.11749</td>
<td>0.44310</td>
</tr>
<tr>
<td>Classical MLS</td>
<td>Quadratic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Fails to compute due to singular moment matrices*
Figure 5.4: Differences of computed deformation field between, a) Meshless with adaptive numerical integration (0.1% accuracy) and ABAQUS, b) Meshless (No adaptive integration) and ABAQUS for the craniotomy induced brain shift simulation in 2D.
From the results, it can be seen that the meshless simulation results obtained with implemented adaptive numerical integration procedure clearly outperforms the meshless results obtained with no adaptive integration. The desired integration accuracy of 0.1% actually resulted in less number of integration points distributed throughout the domain; nevertheless the obtained solution accuracy was better than that of the solution where more integration points are used with no adaptive integration procedure applied. Moreover, for the given support domain radius, the classical MLS with quadratic basis failed due to the singularity of moment matrices, whereas the MMLS (with quadratic basis) had no problem in computing the shape functions.

5.3.2 Extension of a Sheep Brain Sample in 3D

In this example, the extension of a soft tissue in 3D is simulated. A sheep brain sample is modelled in this case (Agrawal et al., 2015). Only the dimension of the brain tissue is taken into account excluding the skull and meninges. The geometry is shown in Figure 5.5-a. Figure 5.5-b and Figure 5.5-c shows the undeformed and deformed configuration of the test sample respectively.

The base (z = 0 mm) of the brain sample is rigidly constrained (169 fixed nodes) while the top surface (z = 5.25 mm) is displaced (169 displaced nodes). A maximum extension loading in terms of displacement (1 mm) is applied smoothly. The hyper-elastic Neo-Hookean material model with Young’s modulus of 3000 Pa, Poisson’s ratio of 0.49 and density of 1000 kg/m$^3$ are chosen. The model is discretized with 1092 nodes, and 18924 integration points are created using the adaptive integration procedure where each original tetrahedral cell is subdivided into two cells by adding a new vertex point at the midpoint of the longest edge of the mother-cell. The results obtained with the adaptive integration procedure are compared with results obtained with no adaptive integration. In the latter case, 4665 tetrahedral background integration cells are used with four integration points per tetrahedron. In each case, the EBCIEM is used to enforce the essential boundary conditions. MMLS shape functions with a constant influence domain ($R= 0.0022$) are
used. For verification, the meshless results are compared with the solution obtained using non-linear finite element solution using ABAQUS. The finite element model consists of 7424 nodes and 4665 quadratic tetrahedral elements with hybrid, constant pressure formulation (C3D10H). Figure 5.6 shows the differences of the computed deformation field between meshless method with and without using the adaptive integration procedure compared with the results obtained with ABAQUS. Table 5.2 shows the average and maximum difference between the two methods.

Figure 5.5: Extension of a sheep brain sample in 3D: a) Dimensions of the sheep brain test sample, b) undeformed, and, c) deformed configuration.

Table 5.2: Extension of a Sheep Brain Sample in 3D - Differences in nodal displacements between the meshless and ABAQUS solutions.

<table>
<thead>
<tr>
<th>Approximation Method</th>
<th>Basis Function</th>
<th>Adaptive Integration</th>
<th>Integration Points</th>
<th>EBCIEM</th>
<th>Avg. Difference (mm)</th>
<th>Max. Difference (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modified MLS</td>
<td>Quadratic</td>
<td>No</td>
<td>18660</td>
<td>Yes</td>
<td>0.021317</td>
<td>0.15124</td>
</tr>
<tr>
<td>Modified MLS</td>
<td>Quadratic</td>
<td>Yes</td>
<td>18924</td>
<td>Yes</td>
<td>0.020046</td>
<td>0.13234</td>
</tr>
<tr>
<td>Classical MLS</td>
<td>Quadratic</td>
<td>Yes</td>
<td>Fails to compute due to singular moment matrices</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 5.6: a) Differences of computed deformation field between meshless (no adaptive integration) and ABAQUS results and b) Histogram of differences; c) Differences of computed deformation field between meshless (adaptive integration) and ABAQUS results and d) b) Histogram of differences - for the extension of a sheep brain sample in 3D. All Dimensions are in mm.

From the results, it can be seen that the meshless simulation with the adaptive integration procedure has higher accuracy compared with the results obtained with no adaptive integration.
5.3.3 Unconstrained Compression of a Cube

In this example, the unconstrained compression of a cube (edge length of 100 mm) having brain-tissue-like constitutive properties is simulated. The meshless discretization of the problem domain and the boundary conditions are as shown in Figure 3.5. The meshless simulations are performed using the MTLED algorithm (Horton et al., 2010), using dynamic relaxation to achieve fast convergence to the steady state solution (Joldes et al., 2009a, Joldes et al., 2011). The cube is discretised using 260 nodes. MMLS uses a constant influence domain for all nodes and the same weights for all the additional MMLS constraints ($\mu = 10^{-7}$). The EBCIEM method is used to impose the EBC and the adaptive numerical integration has an imposed integration accuracy of 0.1%. The constitutive behaviour is described using a hyper-elastic Neo-Hookean material model with Young's modulus of 3000 Pa, Poisson’s ratio of 0.49 and density of 1000 kg/m$^3$. The cube is compressed by displacing the top surface by 20 mm.

For verification, the meshless results are compared with the solution obtained using the non-linear finite element static solver from ABAQUS. The finite element model consists of 1683 nodes and 1014 quadratic tetrahedral elements with hybrid, constant pressure formulation (C3D10H). The imposed displacements and material model result in uniform strain within the cube, and the finite element solution should be exact and independent of discretisation. Figure 5.7a shows the differences in the computed deformation field between results obtained with the classical MLS without applying any adaptive integration procedure to those obtained with ABAQUS. Figure 5.7b shows differences in the computed deformation field between the results obtained with the MMLS approximation with the adaptive integration procedure to those obtained with ABAQUS.
Figure 5.7: Differences in the computed deformation field between, a) classical MLS (linear basis) and ABAQUS. b) MMLS (quadratic basis, \( \mu=10^{-7} \)) and ABAQUS. Boundary conditions – Plane 1: \( dx=0; y, z \) unconstrained. Plane 2: \( dy=0; x, z \) unconstrained. Plane 3: \( dz=-20; x, y \) unconstrained. Plane 4: \( dz=0; x, y \) unconstrained.
The results indicate that the MMLS with adaptive integration procedure with less number of integration points outperforms classical MLS in terms of solution accuracy and provides the results very close to the exact solution obtained using FEA. For the given support domain radius, the classic MLS with quadratic basis failed due to the singularity of moment matrices, as shown in Table 5.3.

Moreover, the unconstrained compression problem is used for meshless simulation with varying desired integration accuracy (and hence varying number of integration points). In each case, the results are compared with the exact finite element solution obtained with ABAQUS. Table 5.4 and Figure 5.8 shows the average and maximum differences between meshless and ABAQUS results with varying integration accuracy. The results show that not much improvement can be achieved in minimizing the differences between the meshless and exact finite element solutions once certain integration accuracy is reached.

### Table 5.3: Difference in nodal displacements between the meshless and ABAQUS solutions for the cube compression simulation. The model is shown in Figure 5.7.

<table>
<thead>
<tr>
<th>Approximation Method</th>
<th>Basis Function</th>
<th>Adaptive Integration</th>
<th>Integration Points</th>
<th>EBCIEM</th>
<th>Avg. Difference (mm)</th>
<th>Max. Difference (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classical MLS</td>
<td>Linear</td>
<td>No</td>
<td>5070</td>
<td>Yes</td>
<td>0.027809</td>
<td>0.10272</td>
</tr>
<tr>
<td>Modified MLS</td>
<td>Quadratic</td>
<td>Yes</td>
<td>4992</td>
<td>Yes</td>
<td>0.016535</td>
<td>0.053706</td>
</tr>
<tr>
<td>Classical MLS</td>
<td>Quadratic</td>
<td></td>
<td></td>
<td></td>
<td>Fails to compute due to singular moment matrices</td>
<td></td>
</tr>
</tbody>
</table>

*The results indicate that the MMLS with adaptive integration procedure with less number of integration points outperforms classical MLS in terms of solution accuracy and provides the results very close to the exact solution obtained using FEA. For the given support domain radius, the classic MLS with quadratic basis failed due to the singularity of moment matrices, as shown in Table 5.3.*
Table 5.4 : Average and Maximum differences between Meshless (varying integration accuracy) and ABAQUS results.

<table>
<thead>
<tr>
<th>Integration Accuracy in Meshless Simulation</th>
<th>Number of Integration Points used in Meshless Simulation</th>
<th>Average Difference (mm)</th>
<th>Maximum Difference (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>4056</td>
<td>0.018707</td>
<td>0.059138</td>
</tr>
<tr>
<td>0.1</td>
<td>4120</td>
<td>0.018321</td>
<td>0.058553</td>
</tr>
<tr>
<td>0.05</td>
<td>5300</td>
<td>0.015415</td>
<td>0.057576</td>
</tr>
<tr>
<td>0.01</td>
<td>14384</td>
<td>0.0076697</td>
<td>0.026414</td>
</tr>
<tr>
<td>0.005</td>
<td>15800</td>
<td>0.0069941</td>
<td>0.020641</td>
</tr>
<tr>
<td>0.001</td>
<td>16180</td>
<td>0.0069182</td>
<td>0.019444</td>
</tr>
</tbody>
</table>

Figure 5.8 : Average and maximum differences between Meshless and ABAQUS results. Meshless results are obtained with varying integration accuracy.
5.3.4 Swine Brain Indentation

In this experiment the usefulness of the adaptive integration procedure is shown for surgical simulation of a body organ. For this purpose, an indentation experiment on a swine brain (Wittek et al., 2008, Zhang et al., 2014) is performed using the meshless algorithms and the results are compared with commercial FEM software ABAQUS.

The swine brain used in the experiment has a mass of 89.9 gm and the dimensions are shown in Figure 5.9. The geometry of the brain was obtained from the MRIs and discretized using 21,498 nodes and 115,029 tetrahedral background integration cells. In the original experiment performed by Wittek et al. (2008), the swine brain was constrained on its base by glue and a custom-made mould which is significantly stiffer than the brain tissue. Therefore, to simulate a fixed base, all nodes on the bottom surface of the brain and the areas in contact with the mould are rigidly constrained. As we emphasis on the evaluation of the performance of our meshless algorithms rather than modelling the interactions between the indentor and the brain, we enforce displacement on nodes contained within a 10mm diameter circle instead of simulating the indentor itself, as shown in Fig. A maximum displacement of 5 mm is applied smoothly on the displaced nodes. Neo-Hookean hyperelastic material model is used for the brain tissue. Poisson’s ratio of 0.49, mass density of $10^3$ kg/m$^3$ and a subject-specific shear modulus of 210 Pa is used (Wittek et al., 2008, Zhang et al., 2014).

The meshless results are obtained with MMLS (vector of positive weights, $\mu = 10^{-5}$, influence radius dilatation = 2) with quartic spline weight function. EBCIEM method is used to enforce the essential boundary conditions exactly. The adaptive integration procedure is used with a desired integration accuracy of 0.1. Tetrahedral background integration cells are used and during the adaptive integration procedure, each cell is subdivided into two sub-cells by introducing a new point at the longest edge of a tetrahedron. Eventually, a total of 528152 integration points is generated by the adaptive procedure. The meshless results are compared with the results obtained
by the commercial finite element software ABAQUS with 162474 nodes and 115029 elements of 10-node quadratic tetrahedron, hybrid, constant pressure (C3D10H) and with a static solver. The difference in computed deformation field between meshless and ABAQUS results are shown in Figure 5.10.

Figure 5.9: Swine brain indentation: meshless discretization and tetrahedral integration grid. The pointed circular area indicates prescribed displacement during the indentation.
Figure 5.10: a) Differences in the computed deformation field between meshless and ABAQUS results, b) Histogram of differences. (all dimensions are in mm)
5.4 Concluding Remarks

EFG methods are based on global weak form and do not require constructing high quality mesh for shape function definition. This makes EFG methods suitable in performing simulations of very large deformations. However, one important weakness of EFG has been the difficulty in performing numerical integration. In the EFG based MMs, Gaussian quadrature over a background mesh are generally used for numerical integration. However, the meshless shape functions are usually non-polynomial rational functions. Also, the shape functions have a much larger support domain compared to FEM and their local supports may not align with the integration cells. Therefore, there are errors associated with the integration schemes used in EFG based MMs, especially in 3D cases with randomly distributed nodes. As discussed in Section 5.1, many integration schemes used in EFG based MMs (Dolbow and Belytschko, 1999, Atluri et al., 1999, De and Bathe, 2001, Racz and Bui, 2012, Puso et al., 2008, Quak et al., 2011) have limitations such as they are only applicable to specific types of support domain, the treatment of irregular boundaries is difficult, instability due to rank deficiency, loss of accuracy etc. Generally, most of the traditional numerical integration algorithms lack in estimating and controlling the desired integration accuracy.

In this chapter, the newly developed adaptive numerical integration procedure (Joldes et al., 2015b) is implemented in the meshless Total Lagrangian Explicit Dynamics (MTLED) algorithm. The motivation behind the development of the adaptive integration procedure is discussed and an algorithm to implement the procedure in MTLED is presented. The adaptive integration algorithm is further enhanced to be used effectively in 3D cases. The presented adaptive algorithm computes integrals with controlled accuracy and generates new integration points only in areas where the desired integration accuracy is not sufficient. In the context of MTLED, the generated integration points for numerical quadrature are not required to be updated as the MMLS shape functions and their derivatives are precomputed and they do not change during the solution process.
Four cases of biomechanics applications are presented to test the adaptive integration procedure, namely, craniotomy induced brain shift in 2D, extension of a sheep brain sample in 3D, an unconstrained compression of a cube having the material properties of soft tissues and a swine brain indentation. The meshless results are compared with the results obtained with the commercial finite element software ABAQUS. In each case of comparison, it is observed that the meshless results obtained with the adaptive numerical procedure is more accurate compared with the results obtained with no adaptive integration.
Chapter 6

Conclusion & Discussion
6.1 Summary of Outcomes

This thesis provided solutions to three major limitations exhibited by meshless methods using EFG formulations. The solutions would enable the computational biomechanics based surgery simulations to be performed by non-experts in the field and thus would be suitable for a clinical workflow. The outcomes of the thesis are as follows:

a) In this thesis, a Modified Moving Least Squares (MMLS) is presented for interpolating scattered data for visualization and performing numerical computations. The new method enables quadratic polynomial base functions to be used with the same size of the support domain as linear base functions, resulting in better approximation capability while maintaining the continuity and smoothness of the approximation. The computation of the MMLS is based on the error functional used in the derivation of the classical MLS approximation augmented with additional base functions to be used with the same size of support domain as linear base function. The computation of MMLS shape function involves a slight modification of the moment matrix. MMLS has better computation efficiency compared to the classical MLS with quadratic basis because MMLS requires fewer nodes in the support domain of each point of interest. MMLS does not require the influence domain radius to be as large as the classical MLS with quadratic basis function. The properties of MMLS such as acceptable nodal distribution, continuity, consistency and invariance are discussed. It was shown that the MMLS approximation has the required order of consistency to be used as an approximation in a meshless method. The invariance of the shape functions against translation and scaling have also been derived. The approximation capability of MMLS has been evaluated as using several univariate and bivariate examples. The numerical examples show that the approximation accuracy of the MMLS is better than that of the classical MLS with linear base functions. It was found that the MMLS has the ability to provide an approximation
for cases where the classical MLS with quadratic basis fails due to the singular moment matrices.

b) In this thesis, the MMLS approximation is implemented in the development of a more robust meshless method based on Element Free Galerkin (EFG) formulation. The MMLS is used for predicting soft tissue deformation using the Meshless Total Lagrangian Explicit Dynamics (MTLED) algorithm. MMLS approximation is expanded to 3D to be used with 3D biomechanics applications. The meshless results are compared with those obtained by the commercial FE software ABAQUS. The simulation results obtained for the biomechanics cases indicate that MMLS has better approximation capability compared with classical MLS with linear basis. For the same support domain size, MMLS showed the ability to provide an approximation when classical MLS with quadratic base functions fails due to singular moment matrices. To further increase the accuracy of the approximation, the method can be extended to use higher order base functions by including in the error functional additional constraints on the coefficients corresponding to the higher degree monomials in the basis.

c) In this thesis, two new ways of imposing Essential Boundary Conditions (EBC) are presented for EFG methods. The new methods are called Essential Boundary Conditions Imposition for Explicit Meshless (EBCIEM) and Simplified Essential Boundary Conditions Imposition for Explicit Meshless (SEBCIEM). The methods are suited for numerical computations using explicit time integration. The new methods consider the total externally applied force in the equation of motion into two parts and presents two approaches to evaluate the externally applied force on the essential boundary. In the first approach, called EBCIEM, the externally applied forces are considered as distributed force and 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,
called 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 get the new displacement field. The applicability of the new methods has been established in a Meshless Total Lagrangian and Explicit Time Integration framework. However, the applicability of the methods can be extended for Updated Lagrangian formulations as well, if the decreased computational performance is acceptable. The EBCIEM and SEBCIEM have been tested for five cases of biomechanics applications involving both 2D and 3D examples. In each case, it was found that both EBCIEM and SEBCIEM impose the essential boundary conditions to machine precision.

d) In this thesis, the adaptive numerical integration procedure (Joldes et al., 2015) is further enhanced to be used in 3D elasticity problems. The algorithm for implementing the adaptive integration procedure in Meshless Total Lagrangian Explicit Dynamics (MTLED) has been presented. The adaptive integration algorithm computes integrals with controlled accuracy and generates new integration points only in areas where the desired integration accuracy is not sufficient. The less smooth function which drives the generation of the integration points during the adaptive procedure is based on the MMLS shape function derivatives. The background integration cells are subdivided into a number of subdomains during the adaptive procedure. In the context of MTLED, the generated integration points for numerical quadrature are not required to be updated as the MMLS shape functions and their derivatives are precomputed and they do not change during time stepping. The applicability of the adaptive procedure is tested for 2D and 3D biomechanics applications. The meshless results are compared with the results
obtained with the commercial finite element software ABAQUS. In each case of comparison, it is observed that the meshless results obtained with the adaptive numerical procedure is more accurate compared with the results obtained with no adaptive integration.

6.2 Limitations of the Study and Directions of Future Research

6.2.1 Limitations of the Study

This thesis has addressed some of the major issues of the explicit meshless methods and implemented new and improved numerical techniques in resolving these issues, as discussed in Chapter 2-5. The accuracy of the numerical methods has been demonstrated in several numerical examples of soft tissue deformation involving large deformations with geometrical and material non-linearities. The scope of this thesis is, therefore, limited in only developing and testing the accuracy of the new and improved numerical methods. However, in real-life surgical simulations, there is a rigorous time constraint in computing the meshless computation for soft tissue deformation as it must be done intra-operatively and should be available to medical personnel within seconds. Addressing the real-time implementation of the developed algorithms is not within the scope of this thesis. Furthermore, the purpose of this thesis is to develop and verify algorithms rather than to perform complete simulations of actual surgery. The practical implementation and experimental validation of the methods to perform non-rigid registration in operating theatre environment is also beyond the scope of this thesis.

6.2.2 Future Work

The following strategies are suggested as some of the scopes for future work in this field of research:
a) A Multigrid Approach for Dynamic Relaxation Solution Algorithm:

In this thesis, to reach the steady-state solution, the Dynamic Relaxation (DR) (Joldes et al., 2009a, Joldes et al., 2011) algorithm has been implemented which is an explicit iterative algorithm. The basic idea of DR is to include a mass proportional damping in the equation of motion that would increase the convergence towards the steady state solution by reducing the oscillations in the transient response. The obtained damped equation is then solved using the explicit integration such as central difference method. Although due to DR’s explicit nature, no large systems of equations have to be solved, still, the number of iterations to obtain convergence can be quite large. This is because explicit integration is only conditionally stable and severe restriction on the time step size has to be included in order to obtain satisfactory results (Bathe, 1996, Miller et al., 2006). Furthermore, as the DR iterative solution is obtained intra-operatively, the computation process has to be very fast and efficient. In this context, implementing a multigrid approach to solve the dynamic relaxation algorithm could be an effective way. The rate of convergence of basic iterative methods can be improved with multigrid methods. The principle is to approximate the smooth part of the error on coarser grids. The non-smooth or rough part is then reduced with a small number of iterations with a basic iterative method on fine grid (Wesseling, 1992). This strategy could significantly reduce the computation time to enable the simulation to be performed in real-time.

b) CUDA-GPU Implementation:

Although the use of multigrid approach for MTLED with dynamic relaxation has the potential to significantly reduce the computation time, parallelization of the algorithms can allow these performances to be enhanced even further. For example, the Total Lagrangian Explicit Dynamics (TLED) algorithm for nonlinear finite element model developed by Miller et al. (2006) was implemented by Taylor et al. (2008) using
GPGPU framework showed up to 16 times speed increase compared with corresponding CPU implementation. Later, Joldes et al. (2010) have successfully implemented the algorithm for finite element model for brain shift computation using CUDA with linear hexahedron with hourglass control, linear tetrahedron and non-locking tetrahedron with contacts between brain and skull. Their method allowed the computations to be performed 20 times faster compared to CPU.

In recent years, the use of GPU has evolved into highly parallel, multithreaded, many-core processor architecture for scientific computation with a very high memory bandwidth. In GPUs, a single unit contains multiple processors and each of these processors operates independently on individual data components. The fundamental reason behind the rapid performance increase in GPUs compared to CPUs is that CPUs are optimized for high performance on sequential code. On the other hand, GPUs use additional transistors more directly for computation. About 80% of transistors are devoted to data processing in GPUs rather than data caching and flow control achieving higher arithmetic intensity (Owens et al., 2007, 2013).

Launched in 2006, CUDA is a general purpose parallel computing platform and programming model which enabled the development of application software to solve many complex problems in a more efficient way than on a CPU. With the increased number of multicore CPUs and manycore GPUs, the mainstream processors are continuously becoming parallel systems. One challenge was to develop application software which could transparently scale its parallelism to facilitate this increasing number of processor cores. CUDA was designed to overcome this challenge by using three key abstractions – a hierarchy of threads groups, shared memories and barrier synchronization. CUDA also comes with a software environment which allows these
abstractions to be implemented by using C as a high-level programming language. Furthermore, these abstractions give the option in partitioning a computational problem into coarse sub-problems that can be solved independently in parallel.

When transforming an application to run on GPU, the data-parallel parts of the application have to be identified first. Furthermore, for effective parallel computation, the computational load among processors has to be balanced while minimizing inter-processor communication (Danielson et al., 2000). Therefore, an important aspect of the parallelization is to minimize the data transfer between host and device. Furthermore, the number of kernel invocation would have to be minimized as well as each kernel invocation contributes a small hardware and software overhead (Danielson et al., 2000, Joldes et al., 2010). Computationally most expensive parts which are calculated intra-operatively at every integration point should be parallelized. In MTLED, CPU usage is dominated by the determination of the vector of internal nodal forces which involves intensive computations of the second Piola-Kirchoff stress vector and the strain-displacement matrix. Therefore, distributing the independent integration points to different processors could be a viable option. All material properties and shape function derivatives may also be calculated and stored at different processors during the pre-operative stage. Further study is necessary for the development and optimization of the CUDA code.
REFERENCES


