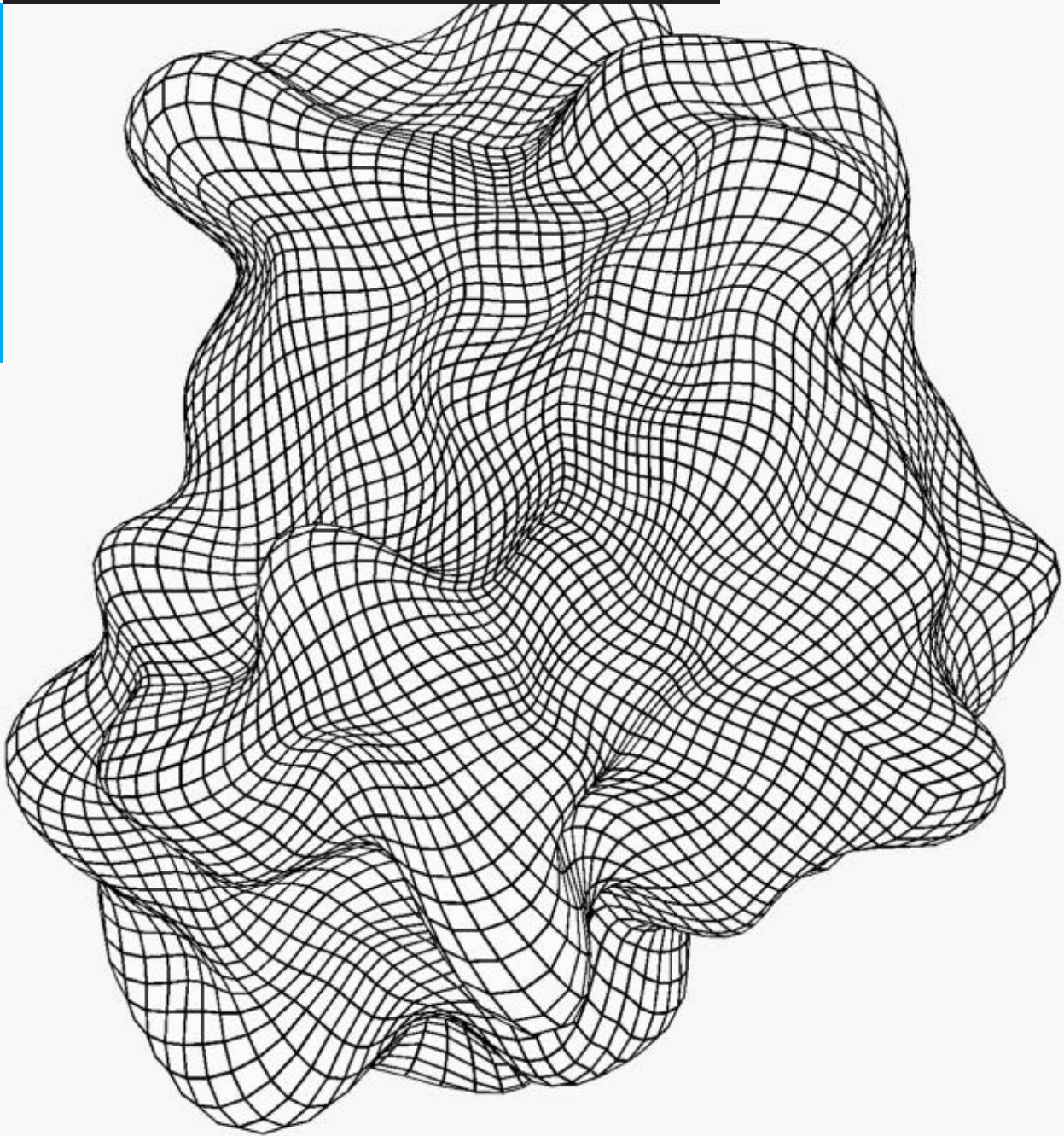


Contact modelling in the Material Point Method

MSc Thesis

Ivaylo Pantev

Delft University of Technology



CONTACT MODELLING IN THE MATERIAL POINT METHOD

MSC THESIS

by

Ivaylo Pantev

in partial fulfillment of the requirements for the degree of

Master of Science
in Civil Engineering,
Geo-Engineering specialisation,

at the Delft University of Technology,
to be defended publicly on Thursday October 27, 2016 at 16:00 PM.

Student number:	4418204	
Project duration:	January, 2016 – October, 2016	
Thesis committee:	Dr P. J. Vardon,	TU Delft chairman
	Prof. Dr M. A. Hicks,	TU Delft
	Prof. Dr Ir. L. J. Sluys,	TU Delft
	J.L. Gonzalez Acosta, MSc,	TU Delft

PREFACE

The following work marks the end of my academic career to date and the start of a new chapter in life. It represents the culmination of two wonderful years in Delft, a time to never be forgotten. The topic of this thesis embodies a core belief held throughout my studies - never to stray from a challenge. I have consistently found tackling a difficult, broad or novel problem has been the best way to learn and improve. This thirst for knowledge influenced my decision to pursue an MSc at TU Delft, one of the finest technical universities in the world. It is my hope that such challenges will never be too far in the future and that they will provide opportunities for continuous development throughout my career.

Here I would like to express my gratitude towards the graduation committee. To my supervisor, Dr Phil Vardon for always remaining positive, even when I was not. You managed to provide inspiration and nurture a deep interest in the topic and numerical modelling in general. I am grateful to Prof. Michael Hicks and Prof. Bert Sluys for being ever present, bringing their humbling experience and providing a critical perspective during our meetings. Not least, Leon Acosta was always available for a chat, brainstorming or debugging session - our conversations made me strive to look for solutions and improve. Here I would also like to thank Noor Pruijn, for setting a stellar example with her work on related issues in MPM and exchanging ideas throughout the thesis. Credit is also due to Bin Wang who was instrumental to the MPM research in the Geo-Engineering section and the start of this project.

I owe a substantial amount to my fellow graduate students and friends in Delft, for forming a tight-knit and supportive community. Being part of a truly multicultural environment has enriched us all, taught us humility and patience among many other things, creating lasting ties across the globe. For all the friendships that started here, thank you.

Finally, I would like to thank my parents and family without whom none of this would have been possible. I am eternally grateful for their unconditional support, timely guidance, wisdom and for giving me the freedom to pursue my own path. This is all one could ever wish for and comprises a debt that can never be repaid.

Ivaylo Pantev
Delft, October 27th 2016

ABSTRACT

In recent years, the Material Point Method has emerged as a promising alternative to the Finite Element Method for solving problems involving localised deformations, large displacements or rotations, fracture, contact/impact problems and others. FEM has been a staple of engineering disciplines both in academia and industry for decades, however its shortcomings make it less suitable for certain applications in geotechnical engineering. MPM, bridges the gap between Finite Elements and meshless methods by using a reference background grid and moving material points to represent the discretised material. This gives it the capacity to faithfully reproduce behaviours which would be difficult to capture without involved procedures with other presently available methods.

In this work, methods of modelling contact between deformable continua were explored in the context of the Material Point Method with the goal of producing usable code applicable to geotechnical engineering scenarios, thus supplementing the work of the Geo-Engineering section at TU Delft. A dynamic, explicit computational framework was developed in the FORTRAN programming language allowing efficient modelling of multi-body interactions using different contact conditions. The influence of decisions regarding the contact logic, definition of surface normals and variable update procedures were compared. Coupling between FEM and MPM was explored for solving problems involving regions with small and large deformations. The work presented here aims to extend the applicability of the Material Point Method to soil-structure interaction problems.

In addition to the contact algorithms explored, an alternative method for material point generation using Centroidal Voronoi Tessellations was explored. This allowed for improved distribution of material points over domains of generic shape, while achieving results similar to a FEM-based approach, which uses Gauss Points. Placing material points at the integration points brings the inherent disadvantage that complex domain shapes cannot be easily discretised using a structured grid. The ability of the two methods to reproduce a uniform density field was compared and the novel approach was implemented for a contact problem - an elastic collision between two disks. Results showed a the potential of the CVT-based scheme for use in further analyses.

CONTENTS

Abstract	v
List of Figures	xi
List of Tables	xiii
1 Introduction	1
1.1 Problem Statement	2
1.1.1 Methodology	2
1.1.2 Code development goals	3
1.2 Report Structure	3
2 Background and Methodology	5
2.1 The Finite Element Method	5
2.1.1 General Procedure	5
2.1.2 Convergence and Stability Criteria	6
2.1.3 Contact in FEM	7
2.2 The Material Point Method	8
2.2.1 Governing Equations	9
2.2.2 Solution Procedure	10
2.2.3 Updating Material Point Properties	13
2.2.4 Implicit and Explicit Integration	13
2.3 Contact in the Material Point Method	14
2.3.1 Detection	14
2.3.2 Resolution	14
2.4 Coupling with FEM	15
2.5 Outstanding Issues in MPM	16
2.5.1 The Explicit MPM code	16
3 Contact Algorithms	17
3.1 Overview of contact algorithms	17
3.1.1 <i>Standard</i> MPM	17
3.1.2 The Multi-Velocity Field-Based Algorithm	17
3.1.3 Notable developments	18
3.1.4 Other approaches	20
3.2 Contact Detection	21
3.3 Contact Geometry	22
3.3.1 Point-To-Point	22
3.4 Multi Velocity Field	23
3.4.1 Procedure	23
3.4.2 Interaction with FEM domain	25
3.5 Point-To-Segment	25
3.5.1 Procedure	25
3.5.2 Alternative	27

3.6	The Contact Normal and Tangent	27
3.6.1	The Normal	28
3.6.2	Nodal support array	28
3.6.3	Gradient of Nodal Mass	29
3.6.4	Gradient of Density	29
3.6.5	Geometric Approach	30
3.6.6	The Tangent	31
3.6.7	Implementation and Comparison	32
3.6.8	Domain Boundary Treatment	33
4	Material Point Generation	35
4.1	Cell Subdivision	35
4.2	Tessellations	36
4.3	Comparison	36
4.4	Application	40
4.5	Conclusion	41
5	Implementation and Benchmark Problems	43
5.1	Dynamic Explicit MPM Formulation	43
5.1.1	Initial Stress Assignment	43
5.1.2	Energy calculation	44
5.1.3	Simple Motions	44
5.1.4	Axial Vibration of a Bar	44
5.2	Contact Problems	46
5.2.1	Elastic Block Collision	46
5.2.2	Elastoplastic Block Collision	49
5.2.3	CVT implementation	49
5.2.4	Block sliding on inclined plane	50
5.2.5	Interaction Between MPM and FEM Domains	52
6	Conclusions and Recommendations	55
6.1	General Conclusions	55
6.2	Conclusions regarding the Multi-Mesh Algorithm	55
6.2.1	Distance between contacting bodies	56
6.2.2	Normal calculation	56
6.2.3	Stability	56
6.3	Conclusions regarding FEM-MPM interaction	57
6.4	Conclusions regarding material point generation	57
6.5	Recommendations	58
6.5.1	Further Work	58
6.5.2	Code Development	58
6.5.3	Meshing	59
	Bibliography	61
A	Contact conditions	65
A.1	Impenetrability condition	65
A.1.1	Contact Force Condition	65
A.1.2	Momentum Equations	66
A.2	Detection of Contact Nodes	66
A.3	Surface Normal Vectors	66

B	Energy conservation and 1-d vibration	69
B.1	1-dimensional vibration	69
B.1.1	Analytical solution for small deformations	70
B.1.2	Initial conditions in numerical model	70
B.2	Results.	70
C	Code development and Data Structures	75
C.1	Data structures	75
C.1.1	Programming	76
C.2	Additional Libraries	76
D	Background mesh	79
D.1	Elements	79
D.2	Shape functions	79
D.2.1	Bilinear quadrilateral elements	80
D.2.2	Shape function gradient discontinuity	80
D.2.3	Quadratic quadrilateral elements	81
E	Constitutive models and relations	83
E.1	Linear Elastic Model	83
E.2	Linear Elastic-Perfectly Plastic Model.	83
E.3	Yield Criteria	84
E.3.1	Von Mises Yield Criterion	84
E.3.2	Mohr Coulomb Yield Criterion	84
F	Variable Update Algorithms	87
G	Elastoplastic Block Collision	91

LIST OF FIGURES

2.1	Contact discretisations in FEM	8
2.2	Disretisation in the Material Point Method with a regular grid.	9
2.3	Stages of an MPM analysis step.	11
2.4	Detection of contact in the multi-velocity field based method.	15
3.1	Application of the penalty method in the contact region [1]	19
3.2	Partitioning of material points into different velocity fields based on damage gradient [2].	20
3.3	Tied interface grid discretisation schematic [3].	20
3.4	FEM-membrane approach summary [4].	21
3.5	Illustration of different contact modes based on detection.	22
3.6	Schematic of the multi-velocity field mapping [5].	23
3.7	Detection of contact between material points and FEM segments [6].	25
3.8	Point-to-segment detection for a single element.	26
3.9	Contact node support - "patch".	28
3.10	Normals and tangents using Mass Gradient for a quarter-circle discretised by different approaches.	29
3.11	Normal and tangent directions at corners.	31
3.12	Normals and tangents using Density Gradient for a quarter-circle discretised by different approaches.	32
3.13	Silent boundaries with dashpots [7].	33
4.1	100 points distributed in a circular domain using Lloyds Algorithm and Monte-Carlo sampling approaches.	37
4.2	100 points distributed in a square domain using Lloyds Algorithm and Monte-Carlo sampling approaches.	38
4.3	Density sampling for patches centred at the original mesh nodes.	39
4.4	Density sampling for patches centred at quasi-random locations.	39
4.5	Normals and tangents using Density Gradient for a quarter-circle discretised by different approaches.	40
4.6	Slope mesh discretised with 500 material points.	40
5.1	Energy of the vibrating bar	45
5.2	Evolution of energy for disk collision with 16MP/el and 0.25m grid size.	46
5.3	Evolution of energy for elastic disk collision under different settings.	47
5.4	Minimal material point volume and nodal mass in Test 4.	48
5.5	Comparison of USL and MUSL for elastic block collision.	48
5.6	Evolution of energy for plastic disk collision.	49
5.7	Colliding disks problem setup.	50
5.8	Results from the CVT-based implementation in disk collision.	50
5.9	Setup used for the sliding block test.	50
5.10	Block sliding on inclined plane for different μ values.	51
5.11	Problem setup for the block-on-beam problem.	52

5.12 FEM-MPM interaction for the block-on-beam problem.	53
B.1 Bar energy from first mode vibration.	71
B.2 Centre of mass velocity for bar in first mode of vibration.	72
B.3 Bar energy from fifth mode vibration.	73
B.4 Comparison of energy errors for fifth mode of vibration.	73
C.1 Overview of data structures used in the program.	75
D.1 Q4 element and local coordinate axes.	80
D.2 Q8 element and local coordinate axes.	81
E.1 Perfect plasticity and the initial stress method [8].	84
G.1 Elastoplastic contact between blocks with different friction modes.	92

LIST OF TABLES

4.1	Density field sampling for a square domain.	39
5.1	Configuration and material properties for axial bar vibration.	45
5.2	Configuration and material properties for elastic block collision.	46
5.3	Configuration and material properties for elastic block collision.	48
5.4	Configuration and material properties for elastic disk collision.	49
5.5	Configuration and material properties for sliding block.	51
5.6	Configuration and material properties for block-on-beam problem.	52
B.1	Configuration and material properties for axial bar vibration.	69
C.1	Subroutines and functions included in module P_IN_POLYGON	77
C.2	Subroutines and functions included in module N_RAPHSON	78
G.1	Configuration and material properties for elastoplastic block collision.	91

1

INTRODUCTION

Over the past decades we have witnessed an exponential growth in the availability and processing power of computers. This has enabled the widespread adoption of a broad range of numerical modelling techniques across scientific and engineering disciplines. The Finite Element Method (FEM) and its variations are a good example of this trend. These have been widely utilized for solving problems across a number of fields, proving themselves as reliable tools for analysis and simulation. FEM has also been adopted in the fields of geotechnical and structural engineering with many commercial software packages using the framework - PLAXIS, ABAQUS, ANSYS, DIANA to name a few. The unique challenges posed by ever more complex projects from engineering practice have continuously driven the development of further numerical approaches. In industry, use of the Finite Element Method has allowed the safe and efficient design of innovative structures. In academia, it has been a valuable tool for modelling and understanding the behaviour of materials, linking physical experiments in the laboratory with the practical application of the gained knowledge.

Despite its widespread adoption and usefulness in engineering practice, FEM has limitations which may preclude the solution of specific problems. Tracking the modelled material through large deformations introduces difficulties related to the calculation mesh, evolving material properties for history-dependent materials, contact detection between different domains and others. For such problems, an alternative approach was needed. The Material Point Method (MPM) is a point-based numerical method with origins in the Particle-In-Cell method (PIC) popular in fluid mechanics, developed for applications in solid mechanics [9]. In contrast to FEM practice, the material domain in MPM is discretised by a set of Lagrangian particles moving through an Eulerian background mesh [6]. These *material points* track the discretised mass and relevant state variables of the material, which are mapped onto the mesh using nodal basis functions.¹ Nodal velocities, accelerations and deformations are computed on the background calculation mesh, followed by a "convection" phase where the positions of the material points are updated and the background mesh reset before the next timestep. This procedure allows the Material Point Method to handle large deformations similar to the Finite Element Method in an Updated-Lagrangian formulation, without suffering from the numerical issues caused by severe mesh distortion.

The presence of a single background calculation mesh in MPM can be beneficial in several aspects of numerical modelling. Use can be made of well-established FEM mesh-generating, data storage, node and element numbering approaches. The mesh also provides a reference for tracking

¹The role of the characteristic function is to define the particle domain, i.e. the way the mass is discretised. In the "standard" MPM implementation, the Dirac delta function is used. The basis functions are similar to the shape functions known from FEM, allowing projection from particles to grid.

multiple objects interacting within the domain, which offers possibilities of modelling contact problems for soil-structure interaction. The point-wise representation of mass also provides opportunities for implementing different point-generation procedures, enabling the easier spatial discretisation of complex structures and irregular domains. These features of the Material Point Method make it attractive for engineering disciplines dealing with complex material behaviour.

At the time of this writing, the Material Point Method has been implemented in a range of geotechnical applications such as modelling pile driving [10], slope stability analysis [7, 11–13], fluid-structure interaction [14], soil anchors [15] and geotextile-soil interaction [4]. Recently, attention has been dedicated to developing and applying the method as a tool for post-failure analysis. An example of this would be evaluating the effects of natural disasters such as landslides on existing structures [14, 16, 17].

1.1. PROBLEM STATEMENT

The Material Point Method currently remains predominantly a tool for academia, with a number of research groups across the world working on developing code for their needs. An area where MPM offers great advantages over FEM is in tackling large deformation problems involving contact such as installation effects or post-failure behaviour of structures. The establishment and implementation of an effective framework for solving these could bring the method closer to adoption in standard geotechnical engineering practice as well as broaden the spectrum of problems which researchers can investigate. The work presented herein explores contact treatment in MPM as well as interactions between domains using different spatial discretisation (FEM-MPM).

Another important part of an MPM analysis is the discretisation stage - generating the material points and background grid. Currently, structured meshes are often adopted in publications. These can face difficulties in accurately representing complex geometries and thus impede schemes which place material points based on the background mesh. An alternative would be the use of unstructured meshes, however these can result in material points of unequal size. An alternative approach, enabling generic-shape domains to be discretised can greatly benefit the use of the Material Point Method.

Based on the problems outlined above, the research questions were crystallised as follows:

- ❏ *How can frictional contact between deformable continua be modelled using the Material Point Method?*
- ❏ *How can effective coupling between the Finite Element Method and Material Point Methods be achieved?*
- ❏ *How can material points be initiated to discretise a continuum?*

1.1.1. METHODOLOGY

Starting from the research questions, a list of objectives to guide the investigation was developed. These were the intended outcomes of this thesis and can be summarised as:

- ❑ Produce and verify an explicit MPM implementation capable of modelling frictional contact between deformable bodies.
- ❑ Investigate the stability of the scheme and quality of modelling results under different definitions of the contact logic. Explore the effects of contact geometry and mesh resolution.

- ❑ Implement an effective procedure to generate material points.
- ❑ Apply the implemented method in problems relevant to geotechnical engineering.

1.1.2. CODE DEVELOPMENT GOALS

The project intends to result in usable code, readily applicable to current and future research efforts involving Explicit MPM. Using the notation and basic structure presented for the explicit implementation of MPM in Wang *et al.*, [12], suitable modifications were outlined to enable the modelling of frictional contact between deformable continua. These includes the definition of different contact conditions such as penalty power scaling of the contact force, adhesion and establishing the surface normal. The developed code was verified against analytical results at different stages, as well as compared to a pure MPM implementation not making use of contact. Finally, the developed and used procedures were designated into a separate code module to promote transferability and code maintenance.

1.2. REPORT STRUCTURE

In line with the research questions and objectives defined, the report structure is summarised below.

Chapter 2 presents an introduction to the procedure of the Finite Element Method and Material Point Method. Key similarities and differences are highlighted, defining the relative advantages in MPM. Recent developments and known issues outside the scope of this works are briefly presented. **Chapter 3** shows a detailed discussion on the implementation of different contact schemes in the Material Point Method. These serve as the basis of the numerical models produced within the scope of the thesis. In **Chapter 4**, material point generation schemes are discussed and a CVT-based procedure is proposed. The implementation of the scheme is presented and this is compared to the current approach (local subdivision at integration points). **Chapter 5** shows the results of the implemented contact algorithms when used to solve benchmark problems. The possibilities and disadvantages are discussed within the context of the solutions obtained. A qualitative application to geotechnical engineering is presented involving the interaction between a deformable barrier and a failing slope. **Chapter 6** summarises the conclusions drawn from the investigation and outlines recommendations for future research.

The included **Appendices** provide additional information used in the implementation of the numerical procedures such as different element types and associated shape functions, constitutive relations, lists of additional subroutines used and a summary of the different contact algorithms employed in the code.

2

BACKGROUND AND METHODOLOGY

In the following chapter, important concepts are introduced in the context of the Finite Element Method as a familiar starting point. Building on to these, an understanding of the Material Point Method, its possibilities, advantages and limitations is developed. The merits of using MPM for applications in Geomechanics are discussed and research goals are defined.

2.1. THE FINITE ELEMENT METHOD

Currently one of the most ubiquitous calculation methods in engineering practice, the Finite Element Method utilises spatial and temporal discretisation to solve differential equations governing mechanical deformation, heat transfer, fluid flow and many other phenomena. Spatial discretisation is done by dividing the domain into a number of subdomains - the *finite elements*. A relationship is established between the local (element) and global space using shape functions, allowing for affine mapping between the spaces. The governing differential equations are then integrated over the problem domain with the application of boundary conditions. The solution is evaluated at *integration points*, fixed locations in the local reference frame of each element and then translated onto the nodes via the shape functions. From there, the solution may be approximated for any point within the domain in the same way.

2.1.1. GENERAL PROCEDURE

The governing equations and their discretisation are presented in an Updated Lagrangian Formulation below (after [18]) as this is directly applicable to the MPM approach used in subsequent modelling. Several governing equations are needed to describe material behaviour in solid mechanics.

- Conservation of mass. Depending on the formulation this may be more or less complex to express. In the Lagrangian formulation *material* coordinates are used, hence the motion of the material can be expressed in an algebraic equation.
- Conservation of momentum. This gives the relationship between change of momentum in the continuum from the internal stresses and body forces.
- Conservation of energy. This is generally not needed for adiabatic processes.
- Deformation measure, relating displacements and strains. This is defined by the formulation chosen. Normally, the deformation gradient F is used and in the Lagrangian formulation $F - 1$ results in the standard engineering strain.

- Constitutive equation, describing the material behaviour. This is a function relating strain, strain rate and any other relevant variables to the stress or stress rate. Typically, stress is thought of as being associated with a point in the material, which is the case in the Lagrangian formulation and the motivation for using this in the Material Point Method. In contrast, a constitutive relation in Eulerian coordinates introduces convective terms in the stress formulation of a history dependent material.

Combining the momentum conservation, constitutive equation and the appropriate strain measure, allows the momentum equation to be expressed as a differential equation in terms of displacements. As such, initial and boundary conditions need to be assigned. Together with the traction boundary conditions these are referred to as the *strong form* of the problem. In order to discretise this with finite elements, the *weak form* (also called variational) must be expressed. The weak form is found by multiplying terms in the strong form equation by a test function and integrating the result over the domain. The principle of virtual work is used to express the work done by the internal and external forces as well as the kinetic forces from material acceleration over virtual (test) displacements. The difference between these must be minimized. Introducing the spatial interpolation in the Finite Element Method is done by using defined shape functions to interpolate the continuous virtual displacements into regions - the finite elements. Expressions can then be developed for the nodal external and internal nodal forces. Once the mass is discretised, the *semidiscrete momentum equation*¹ is found:

$$Ma = f_{ext} - f_{int} \quad (2.1)$$

In FEM, the stiffness and mass matrix are assembled in much the same way resulting in consistent matrices. Equation 2.1 may be solved in a direct way, however it is clear that performing a large matrix inversion for each time step quickly becomes infeasible, hence alternative approaches are usually sought. Generally, simplifying the solution can be done by modifying the mass matrix (e.g. lumping), reducing the complexity of the inverse (e.g. LU decomposition) or a procedure to solve the linear system without inversion (Conjugate Gradient and other methods).

2.1.2. CONVERGENCE AND STABILITY CRITERIA

Convergence of a numerical (discrete) model means that it approaches the analytical solution of the problem. To achieve this, an FEM model must properly describe the problem being solved (Consistency) and be stable so as to arrive at a satisfactory solution (Stability) [19]. Completeness is a requirement for the spatial discretisation employed and means that the shape functions in the element can represent polynomial terms of a given order. Compatibility is a condition for the continuity of the fields across element boundaries and their differentiation within the elements.

Stability criteria are discussed in more detail below. The Finite Element method establishes a geometrical correspondence between the discretised domain(s) and the calculation mesh. Unless some form of mesh refinement is done between time steps, this is maintained until the end of calculation - either upon successful completion or termination due to an error. The correspondence means that in cases of large deformations, the finite elements comprising the mesh may become distorted leading to losses in accuracy and ultimately instability of the solution. This can be linked to a stability criterion called Jacobian Positiveness. As the name implies, it imposes a requirement for the geometry of the element, such that the determinant of the Jacobian matrix remain positive everywhere within the element. J describes the local metric of the element natural coordinates - the

¹Semidiscrete because it is discrete in space, but not discrete in time. The next step is to decide on an appropriate temporal discretisation.

way distances are defined in local coordinate space and a positive determinant of J means the element is convex. In 3-node elements for example, this is equivalent to the nodes forming a triangle and is easily checked. In elements with more nodes, distortions may render portions of the element matrix negative. For higher order elements such as 8-node quadrilaterals, non-corner nodes may produce a concave element shape.

Rank sufficiency requires the element not possessing any zero-energy kinematic mode other than rigid body modes, i.e. it does not allow deformation without a change in energy. This can be checked by examination of the stiffness matrix rank and is ensured by utilizing the appropriate number of Gaussian integration points for the element [19]. An alternative may be to implement a control for such deformation modes (e.g. Hourglass-control).

Other stability conditions are connected with the physical properties of the system (material properties), the discretisation and the integration scheme. Of importance to the explicit integration method used in this work is the Courant-Friedrichs-Levy condition. This defines an admissible maximum timestep size for stability based on the propagation of the mechanical waves and the resolution of the calculation mesh. In reality this is a *necessary*, but not a *sufficient* condition for convergence.

2.1.3. CONTACT IN FEM

A brief discussion on contact modelling in the Finite Element Method is presented below. As MPM uses a FEM mesh for the background grid, some parallels can be drawn in the treatment of these interactions with the two techniques. Based on the spatial discretisation in the area of the contact interface, several general approaches can be differentiated.

Node-To-Node (NTN) schemes are described as simple and stable, however require conforming meshes - a node on one surface matching a corresponding node on the other surface. This limits their applicability to small deformations and makes them poorly suited to frictional contact problems with sliding along the interface. *Note-To-Segment* (NTS) schemes can be more involved and multi-purpose. These require the creation of contact pairs, consisting of a node from one surface and a segment of another. Careful choice of a master and slave surface can make them useable in frictional contact problems. It is worth noting that these generally do not meet the Babuška-Brezzi conditions² and may be unstable. This has been overcome to some extent with more involved techniques such as double-pass application of the penalty method and others [21]. *Segment-To-Segment* (STS) schemes have been successfully implemented, however are not trivial for general cases. An example are "Mortar methods" which use either an intermediate contact surface where contact pressure is defined or assign one of the contacting surfaces as the mortar surface. These have been shown to be consistent for frictional problems, large deformation and non-conforming meshes.

It is worth noting that from a geometrical standpoint, the non-smoothness of the contacting surfaces (piece-wise surfaces in FEM) may preclude the existence of a unique normal projection. The resulting discontinuous normal vector field may cause oscillations and divergence of the solution. Smoothing of the master surface can be used to remedy this or definition of an intermediate, smooth surface solely for the purposes of defining the normal directions.

In general, the frictionless contact problem can be formulated as a variational inequality and

²Also known as the Ladiženskaja-Babuška-Brezzi, LBB or inf-sup condition. It is a necessary condition for the convergence of finite element solutions arising from mathematical analysis. In engineering practice, it is usually replaced by "patch" tests - e.g. assessing the ability of a group of finite elements to accurately represent a uniform stress[20].

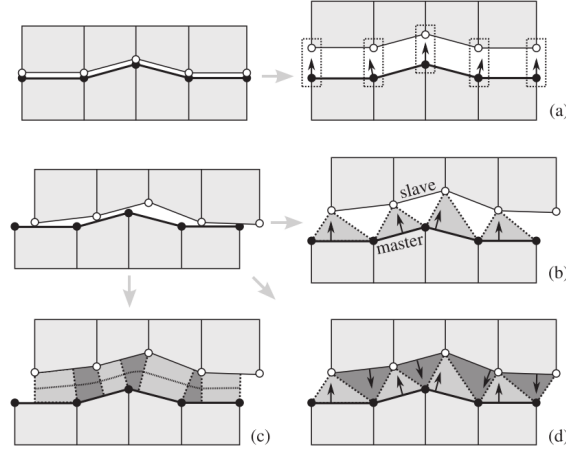


Figure 2.1: Contact discretisations in FEM. a) Node-To-Node b) Node-To-Segment c) Segment-To-Segment d) Discretisation with contact elements [21]

represents a minimisation problem that can be treated in several standard ways (Penalty Method, Lagrange Multiplier, Augmented Lagrange Multiplier, etc.). Unfortunately, the same does not hold for the frictional contact problem as there is no smooth energy functional. The energy of the system (objective function for minimisation) depends on the frictional status, normal pressure, displacements and ultimately - the solution of the problem being solved. A brief overview of key concepts and conditions regarding the treatment of contact is presented in Appendix A. Detailed discussions on the topic can be found in Yastrebov, [21] and Belytschko *et al.*, [18]. Engin *et al.*, [22] describes a practical approach for modelling installation effects applied with little modification to a standard finite element algorithm - the Press-Replace technique. The method is an elegant alternative to the involved application of a true contact algorithm and has promise in areas where one material is displaced by another, however more complex effects such as separation between structure and soil cannot be captured without the use of interface elements.

2.2. THE MATERIAL POINT METHOD

The Material Point Method was first formulated by Sulsky *et al.* as an evolution of the Particle-In-Cell Method, designed for use in solid mechanics. It was developed to address issues, faced when tackling impact, penetration and large-rotation problems using the Finite Element Method. The goal was effectively combining Eulerian and Lagrangian formulations, to avoid convective terms present in the former and mesh distortion characteristic for the latter [9, 23].

Figure 2.2 shows an overview of discretisation in MPM. The influence of PIC methods becomes clear when considering the material points as the representation of the mass. Chen and Brannon, [24] provide a broad overview of the advantages and disadvantages present within the MPM framework. The authors highlight the similarities between MPM and FEM. Indeed, the classic nodal force and mass matrix integrals are discretised by employing the Dirac delta function in the definition of the density field. Thus the mass of the body is discretised into pointwise particle domains - material points, which have no associated volume. This is often referred to as *Standard* MPM, despite the lack of consistent terminology. Love and Sulsky, [25] examined energy consistency within MPM by ensuring momentum conservation and energy dissipation in the finite-element grid dynamics. The authors drew important conclusions such as checker-board pressure modes present when using low-order grid elements. A number of researchers have since highlighted issues in obtaining smooth representations of field quantities. These have been attributed to shear and volumetric

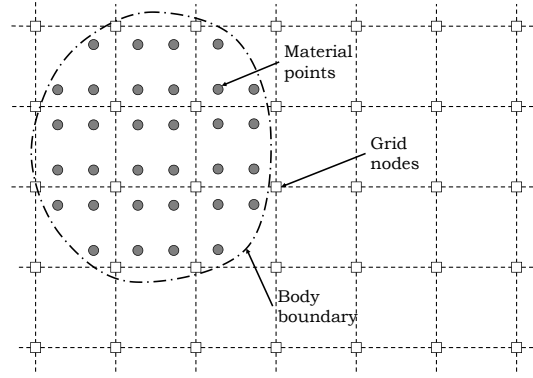


Figure 2.2: Discretisation in the Material Point Method with a regular grid. Fictitious body boundary shown here for reference.

locking [26], as well as the use of low-order interpolation functions in problems with finite deformation [27]. These issues are compounded by the lack of an adopted consistent framework for the mathematical analysis of the Material Point Method to date.

Further variations have since been developed including the Generalised Interpolation Material Point (GIMP) Method [28] and Convected Domain Particle Interpolation (CDPI) Method [29]. A key difference in these methods is the grid basis function used to transfer grid values to particle values and the tracked deformations of the assigned particle domains. Effectively, these approaches were developed in attempts to remedy issues with stress oscillations observed when material points cross between cells of the calculation mesh.

Despite being developed more than a decade ago and its relation to the widely used FEM, the Material Point Method remains a tool predominantly used in academia and under development. This is due to unresolved problems, such as boundary treatments and multiphase interactions which remain an active area of investigation. Due to its inherent advantages over the FEM formulation, MPM holds the promise of becoming a widely-used numerical technique in the fields of Geotechnical Engineering and Soil Mechanics.

2.2.1. GOVERNING EQUATIONS

The two governing equations in the Material Point Method are conservation of mass and conservation of momentum. The following derivations will be carried out in the Updated Lagrangian formulation most frequently employed in the Material Point Method. The solution procedure employed is analogous to that of FEM in the Updated Lagrangian formulation (UL-FEM), with one key difference. Material points are used in MPM to store state variables as opposed to the integration points of FEM. This represents both the main strength and a current weakness of the method. As deformations are accumulated the material points convect away from their original locations. This introduces errors from interpolation into the quadrature used for integrating the governing equations in time.

$$\frac{d\rho}{dt} + \rho \nabla \cdot v = 0 \quad (\text{Conservation of mass}) \quad (2.2)$$

$$\rho \frac{dv}{dt} = \nabla \cdot \sigma + b \quad (\text{Conservation of momentum}) \quad (2.3)$$

Where v is the velocity, ρ the mass density, σ the Cauchy stress and b the applied body force. We will express the body force as the sum of forces due to gravity and applied surface tractions, which become the external force F_{ext} after integration over the material domain:

$$F_{ext} = \int_{\Omega} b dV = \int_{\Omega} g \rho dV + \int_{\Omega^s} \rho s^s d\Omega^s \quad (2.4)$$

Where Ω^s represents the surface on which tractions are applied. Momentum conservation is imposed, by discretising and solving the equations of momentum over the domain. Following standard FEM practice, the weak form of Eq. 2.3 can be obtained by multiplication with a test function w and integration over the current configuration (after [24, 30]):

$$\int_{\Omega} \rho w \cdot a d\Omega = - \int_{\Omega} \rho s^s : \nabla w d\Omega + \int_{\Omega^s} \rho c^s \cdot w dS + \int_{\Omega} \rho w \cdot b d\Omega \quad (2.5)$$

Converting to the notation used here and ignoring the higher-order term $\int_{V^t} \Delta \sigma \cdot \Delta \eta :$

$$\int_{V^t} \Delta \sigma \cdot \delta \Delta e dV + \int_{V^t} \sigma^t \cdot \delta \Delta \eta dV = F_{ext}^{t+\Delta t} - \int_{V^t} \sigma^t \cdot \delta \Delta e dV \quad (2.6)$$

The above equations are discretised by approximating the material domain as a finite number of material points. These have a constant mass and their number remains fixed thereby guaranteeing mass conservation. The mass density can then be discretised into point-wise contributions (material points) using the Dirac Delta function.

$$\rho(x, t) = \sum_{p=1}^{n_p} m_p \delta(x - x_p) \quad (2.7)$$

Substituting Eq. 2.7 into Eq. 2.3 allows the integral to be written as a sum over the material points within the domain.

$$\sum_{p=1}^{n_p} m_p \delta u_i(x_p) a_i(x_p) + \sum_{p=1}^{n_p} m_p \frac{\partial \delta u_i}{\partial x_i} \bigg|_{x_p} \sigma_{ij}^s(x_p) = \sum_{p=1}^{n_p} m_p \delta u_i(x_p) b_i(x_p) + \sum_{p=1}^{n_p} m_p \delta u_i(x_p) \bar{t}_i(x_p) \quad (2.8)$$

At this point, the similarity between the weak formulations of MPM above and FEM can be seen. The material points assume the role of the Gauss points used for numerical quadrature in FEM. In the standard case, field gradients are evaluated on the background grid nodes with a procedure equivalent to that of FEM albeit with the locations of material points within the element, not integration points³. State variables are stored at material point locations and these move through the mesh, hence a convective term describing the material motion as required by Eulerian FEM formulations is not needed, thereby reducing sources of error.

2.2.2. SOLUTION PROCEDURE

The derivations presented below are shown in the context of an explicit formulation as used in subsequent analyses. The general solution procedure employed by the Material Point Method can be separated into three main stages. The general solution procedure employed in the Material Point Method is shown Figure 2.3 below. The derivations presented in subsequent sections are shown in the context of the explicit formulation as used in the analyses throughout this thesis. In particular, the Update-Stress-Last approach is implemented here, referring to the order of updating material point stresses within the time step. A brief discussion on the variable update schemes is shown in Section 2.2.3.

³Additional steps are required where nodal basis functions with an extended support are used, such as in the GIMP [28].

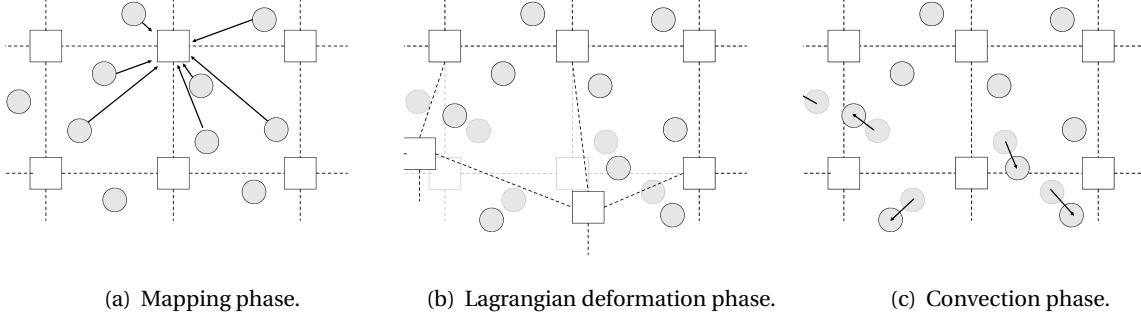


Figure 2.3: Stages of an MPM analysis step.

MAPPING PHASE

During this stage, sometimes referred to as *forward mapping* the mass, velocity, stress and other state variables are interpolated from the material points onto the background grid. First, mass at the nodes is interpolated populating the mass matrix. Following the discussion from the previous section, the mass matrix is then lumped. This is often done for convenience in dynamic analysis as it reduces storage requirements (matrix stored as vector) and greatly simplifies the solution procedure (matrix inversion trivial). Here one of the key differences between FEM and MPM can be highlighted. Due to the motion of the material points through the background grid, the mass matrix (and stiffness matrix where needed) must be updated for each time step.

$$m_i^t = \sum_{p=1}^{n_p} N_i(\xi_p^t) m_p \quad (2.9)$$

Here and in subsequent equations, $N_i(\xi_p^t)$ is the value of the nodal basis function for node i with respect to particle p . The local coordinates of the material point are ξ_p . Next, the momentum at the nodes is interpolated using also the material point velocity v_p .

$$(mv)_i^t = \sum_{p=1}^{n_p} N_i(\xi_p^t) m_p v_p \quad (2.10)$$

Traction (where applied) and gravity together form the external forces acting on the body. Their nodal values are also mapped from the material points.

$$F_{i,gravity}^t = \sum_{p=1}^{n_p} N_i(\xi_p^t) m_p g \quad (2.11)$$

$$F_{i,traction}^t = \sum_{p=1}^{n_p} N_i(\xi_p^t) \bar{F}_{p,traction}^t \quad (2.12)$$

Finally, the internal force at the nodes is given by the divergence of material point stresses σ_p , volume V_p and the gradients of the shape functions $G_i(\xi) = \nabla N_i(\xi)$.

$$F_{i,internal}^t = \sum_{p=1}^{n_p} G_i(\xi_p^t) \sigma_p^t V_p^t \quad (2.13)$$

Note the presence of the volume term V_p in the equation above. While the material points do not have an explicitly defined shape, their volume plays a role in the calculation of internal forces. This is equivalent to the changes in material density that can be observed as the mass associated with individual material points does not change over the course of the analysis.

LAGRANGIAN DEFORMATION

The inertial force at the nodes is found by summing the nodal forces described in the previous section. From that, the nodal acceleration can be calculated ⁴.

$$F^t = M^t a^t = F_{ext}^t - F_{int}^t = F_{traction}^t + F_{gravity}^t - F_{internal}^t \quad (2.14)$$

Where M^t is the mass matrix assembled from the individual nodal contributions. The choice between a consistent mass matrix or a lumped mass matrix can be made here. The lumped mass matrix represents a diagonal matrix with term equal to the row sums of the consistent mass matrix ⁵. Operations with the diagonal matrix are then trivial and avoid the need for costly matrix inversion of the consistent mass matrix. This is often done in practice as a measure to save computational time, however it has been found to incur a dissipation of energy [31–33] in addition to errors in angular momentum conservation [34].

$$M^t = \begin{pmatrix} m_{i,i} & 0 & 0 \\ 0 & m_{i+1,i+1} & \ddots \\ 0 & \ddots & \ddots \end{pmatrix} \quad (2.15)$$

Once the nodal accelerations are calculated, the nodal velocity and displacement fields can be computed. It is here that the distinction between USL and MUSL can be made [9]. In the USL approach nodal velocities can then be updated in a straightforward fashion.

$$v_i^{t+\Delta t} = v_i^t + a_i^t \Delta t \quad (2.16)$$

In order to improve the stability of calculations authors have proposed using MUSL, which updates nodal velocities by first updating material point velocities [10, 30]. In the Modified-Update-Stress-Last method, nodal momentum is calculated from Eq. 2.14 avoiding the division by the lumped mass. The argument for this is the possibility of small mass mapped onto grid nodes potentially causing singular values for nodal velocity.

$$v_p^{t+\Delta t} = v_p^t + \sum_{i=1}^{n_{en}} \Delta t N_i(\xi_p^t) a_i^t \quad (2.17)$$

$$v_i^{t+\Delta t} = m_i^{t,-1} \sum_{p=1}^{n_p} \Delta t N_i(\xi_p^t) v_p^{t+\Delta t} \quad (2.18)$$

CONVECTION PHASE

Once the nodal velocities are integrated, the incremental nodal displacements and material point positions can be updated.

$$\Delta u_i^{t+\Delta t} = v_i^{t+\Delta t} \Delta t \quad (2.19)$$

$$x_p^{t+\Delta t} = x_p^t + \sum_{i=1}^{n_{en}} N_i(\xi_p^t) \Delta u_i^{t+\Delta t} \quad (2.20)$$

Finally, the stress and strain increments at the material points are evaluated. The volumetric strain increment is used to update volume associated with material points.

⁴This procedure is characteristic for the USL and USF approaches. In the case of MUSL, nodal momentum is used along with an extra mapping step.

⁵Called the Row Sum lumping method. Other possibilities include Diagonal Scaling, Quadrature with nodal locations etc.

$$\Delta \epsilon_p^{t+\Delta t} = B(\xi_p^t) \Delta u_p^{t+\Delta t} = B(\xi_p^t) v_i^{t+\Delta t} \Delta t \quad (2.21)$$

$$\sigma_p^{t+\Delta t} = \sigma_p^t + \Delta \sigma_p = \sigma_p^t D \Delta \epsilon_p^{t+\Delta t} \quad (2.22)$$

$$V_p^{t+\Delta t} = (1 + \Delta \epsilon_{p,vol}^{t+\Delta t}) V_p^t \quad (2.23)$$

N.B. Volume update for the material points is based on the assumption of small deformation between separate time steps. Again, smaller timesteps will benefit the accuracy of the Explicit MPM solution.

2.2.3. UPDATING MATERIAL POINT PROPERTIES

Two main parameter updating schemes can be found in literature - Update Stress First (USF)[35] and Update Stress Last (USL)[9], with a Modified Update Stress Last (MUSL)[30] scheme based on the latter. The USF algorithm calculates stresses in the bodies using the material point velocities at the start of the time step before the forward mapping procedure to calculate nodal loads. Wallstedt and Guilkey, [36] argued that this can improve the robustness of calculations as the velocity field used for the stress update is smoothly varying. In USL on the other hand, stresses are calculated at the end of the time step after the motion has been resolved. MUSL represents a solution to problems in calculating nodal velocity of nodes with a low associated mass. Nodal momentum instead of nodal acceleration is used to update the particle velocities in the modified algorithm. No explicit criteria have been found for updating stresses at the start or the end of time steps. Authors have shown the USF scheme to slowly increase total energy over time, while USL and the related MUSL tend to dissipate energy [7, 30, 37]. That being said, a comparison will be made between the different methods to demonstrate their performance in the problems analysed.

2.2.4. IMPLICIT AND EXPLICIT INTEGRATION

In this section implicit and explicit approaches will be discussed in the context of solving the differential equations involved in MPM (and FEM). A good discussion and comparison between IMPM and EMPM is given in [38]. Structural and solid mechanics problems, when discretised and solved numerically fall in two broad categories presented below using generic governing equations.

$$[K](x) = (F) \quad \text{for linear, static analyses} \quad (2.24)$$

$$[M](\ddot{x}) + [C](\dot{x}) + [K](x) = (F) \quad \text{for linear, dynamic analyses} \quad (2.25)$$

An implicit method involves finding the unknown displacement solution u for the current time step by using information from both the current and previous time steps. Different schemes exist for implicit integration of differential equations such as the backward-Euler, Newmark and Crank-Nicolson method. Of these, the Newmark method is often employed in modelling the dynamic response of structures and solids with the Finite Element Method.

$$v^{t+\Delta t} = v^t + [(1-\gamma)a^t + \gamma a^{t+\Delta t}] \Delta t \quad (2.26)$$

$$u^{t+\Delta t} = u^t + v^t \Delta t + \left[\left(\frac{1}{2} - \beta \right) a^t + \beta a^{t+\Delta t} \right] \Delta t \quad (2.27)$$

The implicit definition of the displacement u and velocity v can be easily identified in the expressions above. An iterative solution scheme to find appropriate values of $u^{t+\Delta t}$ is used such as the Newton-Raphson Method. Starting with an initial guess for the displacements in the current step, the solution is incremented until satisfactory convergence is obtained. It is this procedure

that allows significantly larger time steps to be taken with an implicit solution procedure. Both static and dynamic analyses can be solved in a similar fashion, with some differences in the way the mass matrix is treated [12]. Often in civil engineering practice, loading is considered static and the equilibrium solution is of interest, hence a quasi-static analysis is carried out with an implicit formulation. A drawback of the approach is the requirement for the stiffness matrix inverse K^{-1} , which becomes a computationally intensive task as the system increases in size. Approaches such as preconditioning and iterative schemes to find the matrix inverse exist, however these will not be discussed here.

In contrast, an explicit time integration is commonly used to solve for the acceleration term $\ddot{x}$, thereby avoiding inversion of the stiffness matrix. Still, the mass matrix inverse M^{-1} is needed to solve dynamic problems. In the Material Point Method, this can be particularly problematic as the mass matrix changes to reflect the moving material points, hence its inverse must be recalculated at each time step. To this end, a diagonal lumped mass matrix is most often used [7, 12, 24, 38]. Another key difference with the implicit procedure is the fact that equilibrium of the internal forces is not enforced in the explicit scheme, which can cause errors to accumulate. Once the acceleration is known, this is used to explicitly advance the solutions for velocity and displacement as shown in Equation 2.16. This procedure corresponds to the (Forward) Euler method for numerical integration, which is a first-order explicit method. Its accuracy depends on the size of the step taken between solutions, hence the time step size influences the accuracy of the solution. The method can also be unstable depending on the size of the time step. Explicit time integration is normally used where complex loading scenarios are present, such as the contact and impact problems examined in this thesis. A limit for the maximum time step can be determined from the CFL conditions. These represent necessary, but not sufficient for stability and particularly for contact problems a smaller step may be needed.

2.3. CONTACT IN THE MATERIAL POINT METHOD

The Material Point Method has the implicit capacity to handle no-slip contact without the need for contact detection [30, 35]. This however limits the method's applicability for simulations involving frictional materials or soil-structure interaction particularly important in the field of geotechnical engineering.

2.3.1. DETECTION

In MPM, equation of motion are resolved using the background calculation grid. The presence of the grid allows contact in MPM to be detected without the need for predefined contact surfaces. Figure 2.4 illustrates two bodies coming into contact at several nodes, identified as having multiple contributions mapped onto them. When appropriate data structures are implemented, a single sweep of the background grid nodes allows contact locations to be identified. This can be greatly beneficial in modelling the interaction of large groups of bodies, objects with complex geometry or fluid-structure interaction problems. Alternatively, contact may be detected at generic locations, however this requires a more involved procedure for detection.

2.3.2. RESOLUTION

Bodies in the Material Point Method are discretised as sets of discrete particles, making the body boundaries discontinuous and thus not trivial to establish. In order to resolve contact between two interacting continua, the conditions outlined in Appendix A must be imposed at the contact location. Depending on the method chosen to detect interaction, this may be on the nodes of the computational mesh or otherwise. The latter approach is briefly discussed in the next section. If contact is resolved on the background grid, nodal quantities are used to distinguish the interacting

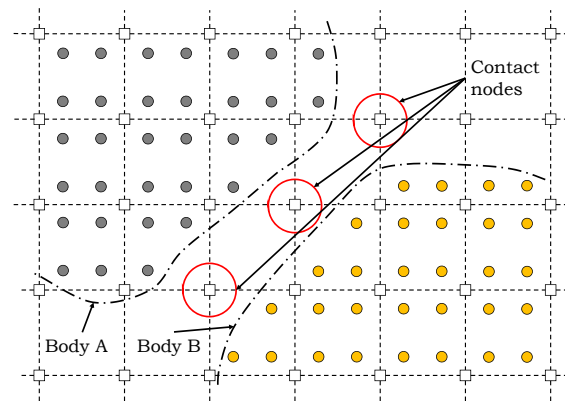


Figure 2.4: Detection of contact in the multi-velocity field based method.

bodies. Establishing the location of occurring contact and the overall accuracy of the calculation depend on the grid density [1]. The idea of combining different mesh spacings in a single model was explored in Lian *et al.*, [3], in an attempt to counteract the computational cost of using a refined mesh over the entire domain. Additionally, the distance between boundary particles and the respective nodes at which contact has been detected introduces inaccuracies in the determination of the normals to contacting body surfaces. Combined, these factors introduce errors of momentum conservation into the solution of the contact problem [35, 37]. Other authors have called for a detailed examination of the influence frictional contact algorithm have on total system energy [25].

Objects are considered to come into contact when their velocity fields are mapped onto the same background node. Observing the impenetrability condition (Appendix A) bodies are not allowed to overlap, hence contact may occur despite a significant distance being present between the boundary particles (Fig. 2.4). In addition to the momentum and energy conservation errors mentioned, oscillations have been observed in the stress solutions at the contact interface. Authors have proposed the use of a penalty function [1] or distribution coefficients for small mass nodes [33] to help alleviate the problem. In a sense, these resemble the GIMP, CPDI and Dual Domain Material Point (DDMP) procedures developed to tackle stress oscillations within the discretised bodies [28, 29, 39].

2.4. COUPLING WITH FEM

Recent research in the field has demonstrated effective coupling of MPM with other numerical techniques, FEM in particular [4, 6, 40–42]. By combining these techniques, their respective benefits can be leveraged, including the ability of MPM to handle large deformation and the superior stress calculation possible with FEM. An example application would be modelling the interaction between a stiff structure, discretised using FEM and a softer soil body discretised using MPM. The setting of the problem may allow the definition of normals to occur with respect to the stiffer body ensuring better energy and momentum conservation properties. Additionally, the surface discretisation in the Finite Element Method could allow for more accurate determination of the contact location without the need to resolve contact on the computational mesh [6]. These approaches have been implemented for fluid-structure interaction and collision of solids, however few applications to geotechnical soil-structure interaction were found at the time of writing [4].

2.5. OUTSTANDING ISSUES IN MPM

Improving stress recovery in the Material Point Method stands as another area of investigation. If no treatment is implemented, stresses are calculated using a procedure analogous to standard FEM. Material point strains are found from the deformation gradient and the stresses determined from the constitutive relation. The shape functions used in mapping from nodes to material points allow a continuous field to be produced, however the gradients of these functions are not continuous across element boundaries. Low-order elements are particularly vulnerable to these errors, which result in a sharp jump in stress evaluated at a material point crossing between cells. Several approaches to overcome this have been proposed, based on modifying the interpolation (shape functions) to provide a continuous gradient across cell boundaries [28, 36, 43]. In addition to this, researchers have highlighted kinematic locking effects and these can cause non-physical stress behaviour [26]. Investigating these issues lies outside the scope of this work, however incorporating a successful strategy for limiting their effect can greatly benefit the accuracy of MPM analyses.

Applying tractions along moving boundaries in the Material Point Method is another problem without a satisfactory solution to date. Surface tractions may be applied to an MPM body either via its outermost particles (defined as a set of boundary particles) or imposed onto grid nodes directly (e.g. via the moving mesh concept). Errors from interpolation may cause a variation in the applied traction, hence the first approach is not desirable [7]. Imposing tractions as forces acting at the grid nodes can also result in errors as the material points move through the grid, changing the magnitude of the traction applied. Several authors have used the moving mesh concept to apply traction boundary conditions to domains discretised with material points [7, 44].

Challenges specific to the field of geomechanics such as large deformations, contact between deformable bodies, fracture modelling, highly non-linear or multiphase materials and coupled problems, define the need for numerical procedures capable of reliably modelling these types of behaviour. As presented in this thesis, the Material Point Method shows great potential for applications in the field of geotechnical engineering and may well continue to be a dynamic field of research in years to come.

2.5.1. THE EXPLICIT MPM CODE

In line with the research objectives defined in Section 1.1.1, all presented analyses were developed in an explicit, dynamic MPM framework, modelled after the work of Wang *et al.*, [12]. At the start of computational work, the code was verified by modelling simple dynamics problems and comparing results to analytical solutions (Appendix B). An Update-Stress-Last scheme was used to update stresses at the end of each time step using nodal velocities calculated from nodal accelerations [12, 31]. This was extended to include the MUSL approach, introducing an additional mapping step to improve stability. Analyses were carried out using a grid of bilinear quadrilateral elements. Following the discussion in [27] on interpolation errors, these were seen as a suitable starting point.

Incorporation of FEM-discretised bodies to be solved as part of the analysis was done using approaches detailed in Smith *et al.*, [8]. The code utilises the libraries **main** and **geom** available in addition to the book. Procedures needed outside the scope of these libraries were compiled in additional libraries (Appendix C).

3

CONTACT ALGORITHMS

The Material Point Method has the implicit ability to handle no-slip contact and separation between bodies, with the advantage of avoiding a dedicated contact detection phase [1]. Not accounting for friction however may cause inaccuracies in many practical applications, particularly in penetration problems [9, 10]. A solution developed by Bardenhagen, Brackbill, and Sulsky [35] incorporates Coulomb friction along the contact surface. A large difference in stiffness of contacting materials may result in velocity and acceleration oscillations at the surface. Another issue may be caused by adopting the Coulomb friction model. This defines shear stress along the surface as proportional to the normal stress. In an undrained analysis of cohesive soils for instance, the shear stress cannot exceed the undrained shear strength of the soil and thus the shear strength may need to be limited.

3.1. OVERVIEW OF CONTACT ALGORITHMS

A number of contact algorithms can be found in literature, detailing contact handling within the MPM framework as well as in coupled FEM-MPM applications. The latter are particularly interesting for simulating soil-structure interaction in geotechnical engineering, where the superior stress recovery possible with the Finite Element can give more pertinent information regarding the structural response.

3.1.1. *Standard MPM*

In the original MPM implementation [9, 30] all elements were assumed to deform in the flow of material. That is, material points move in proportion to the nodes according to the nodal basis (shape) functions. These functions were used to map the nodal velocities and accelerations continuously onto the element interior, creating a single-valued continuous velocity field used to update point positions. Analogously, the velocities of the points were updated using the nodal accelerations. Provided the impenetrability condition is satisfied, no special contact algorithm was required to simulate impact and penetration. As such, the Material Point Method did not allow the treatment of sliding between bodies without a scheme to differentiate between velocity fields. This poses a limitation for applications such as modelling adhesion between a structure and soil, fracture and penetration.

3.1.2. THE MULTI-VELOCITY FIELD-BASED ALGORITHM

Bardenhagen *et al.*, [35] is widely referred to as one of the first attempts to overcome the no-slip condition implicit to the Material Point Method by introducing Coulomb friction along the contact surface. The resulting algorithm offered advantages in terms of simplifying the contact detection stage and a computational cost linear in the number of bodies [35]. The method has since become

one of the most widely used in literature, with a number of researchers proposing modifications based on the original in their work [1, 23, 45, 46]. The approach used to model contact in the MPM framework can be thought of as a *predictor-corrector* scheme. The nodal velocities of each body are predicted separately ignoring potential contact and then corrected using a contact model. The model is applied to contact nodes only, i.e. nodes receiving a contribution from the particles of more than one body.

Pan *et al.*, [46] proposed a modified procedure for calculating the normal and tangential forces at the contact interface and reported improved energy conservation. The approach is based on the statement that while contact is occurring at node i , the normal accelerations of the two bodies mapped onto node i will remain equal with opposite signs. This allows the normal contact force to be expressed using Newton's Second Law to satisfy equilibrium between the normal contact force and internal forces in the body at node i .

3.1.3. NOTABLE DEVELOPMENTS

USF IMPLEMENTATION

Huang *et al.*, [23] detailed two contact algorithms using the multi-mesh scheme. The *First* contact algorithm proposed follows the procedure of the original work of Bardenhagen *et al.*, differing in the way the contact normal is established. The authors used the gradient of nodal mass to establish the surface normal for each body and defined rules to ensure collinearity of the normals for contacting bodies. If n_i^b is the normal used for contact resolution and $\hat{n}_i^b$ the normalised surface normal to body b at node i , then:

$$n_i^1 = \begin{cases} \hat{n}_i^1 = -n_i^2, & \text{if Body 1 is stiffer than Body 2.} \\ \hat{n}_i^1 = -n_i^2, & \text{if Body 1 contact surface is flat/convex and Body 2 surface is convex.} \\ -n_i^2 = \frac{\hat{n}_i^1 - \hat{n}_i^2}{|\hat{n}_i^1 - \hat{n}_i^2|}, & \text{in other scenarios.} \end{cases} \quad (3.1)$$

The researchers employed the Update-Stress-First scheme in their investigation, meaning material point stresses were updated from the material point velocities at the start of the time step. They argued that nodal velocities used in the stress update may violate impenetrability leading to disturbances in the system. A *Second* algorithm was also investigated enforcing the contact conditions both at the start and end of each time step. This was done in the same way as the *First* method.

INTRODUCTION OF AN ADHESIVE TERM

The frictional contact model can be enhanced by the addition of an adhesive term to the tangential contact force component. Al-Kafaji, [7] argues that modelling adhesive contact is often required for geotechnical applications. The additional component can be added to the tangential force definition as follows:

$$f_{k,tan}^{max,t+\delta t} = f_{k,adh}^{t+\delta t} + \mu |f_{k,norm}^{t+\delta t}| = aA_k^t + \mu |f_{k,norm}^{t+\delta t}| \quad (3.2)$$

A key observation can be made is regarding the stress oscillations on the contact interface. The examples developed in the work initiate a pair of bodies in contact and assign to them stresses which are initially in equilibrium. This may not be applicable to scenarios where bodies have come into contact, stopped their relative motion and then prevented from moving again via adhesion.

LIMITING SHEAR AND CONTACT FORCE OSCILLATIONS

Ma *et al.*, [1] built onto the frictional contact algorithm by including a penalty function to reduce velocity and acceleration oscillations in the region of the contact surface, as well as adding a limiting

shear along the interface to represent undrained soil behaviour. This was done by introducing a shear stress ratio (α) between the maximum shear stress due to contact and the undrained shear strength of the soil, where $0 \leq \alpha \leq 1$.

$$\tau_i = \mu_c \sigma_{norm,i} \leq \tau_{max} = \alpha s_u \quad (3.3)$$

The authors argue that oscillations at the contact interface can also be moderated by using a more refined background mesh, but at a significantly increased computational cost. As contact is resolved on the nodes of the computational mesh, the impenetrability condition is imposed as soon as material points from different bodies enter adjacent cells. In essence, the bodies are in contact despite there being a distance between them. The penalty function constitutes a violation of the impenetrability condition, however increases in numerical accuracy were shown with appropriate selection of penalty parameters. A drawback of this approach was the introduction of the penalty factor which also requires calibration and can ultimately affects the results in a significant way.

$$\bar{v}_{norm,i} = v_{norm,i} f_p = v_{norm} \left(1 - \left(\frac{\min(s, d)}{d} \right)^k \right) \quad (3.4)$$

Where d is the cell size, k is a penalty power and s the normal distance to the stiffer body. In practice, s is hard to compute due to the lack of a defined interface and this is replaced by s' , the distance between the contact node and the closest material point of the stiffer body. Figure 3.1 below illustrates the use of the method.

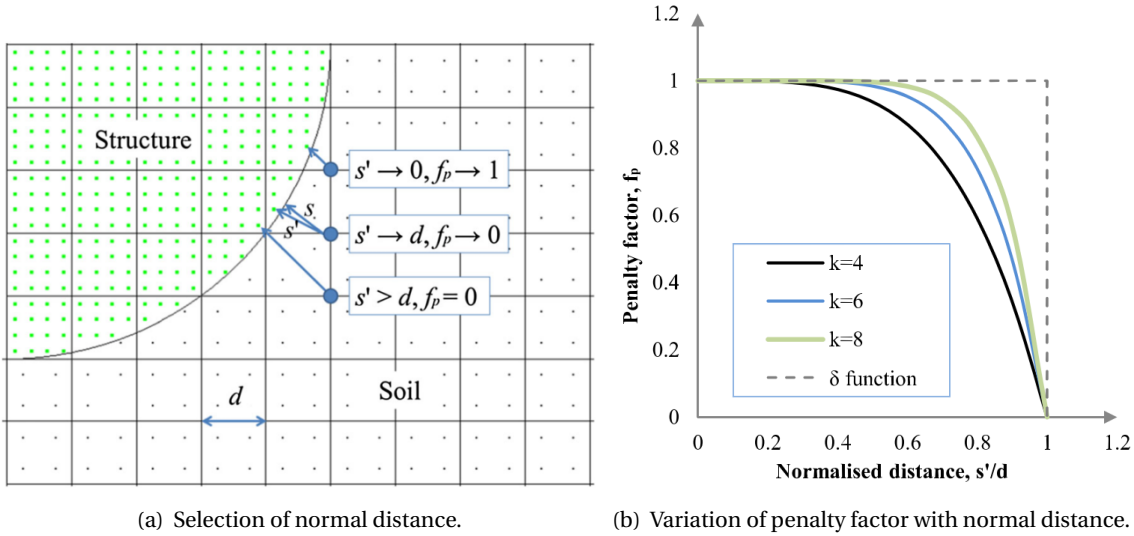


Figure 3.1: Application of the penalty method in the contact region [1]

FRACTURE AND SELF-CONTACT

Recently, [Homel and Herbold](#), [2] proposed an approach to separate material points into different velocity fields once sufficient damage is accumulated in a region of the domain. Once material points are assigned into the different data structures, these can be treated as separate entities. This enabled self-contact and fractures to be captured by the model, starting from a single body. Post-fracture frictional interactions were modelled using the multi-velocity field based algorithm. Conceptually the approach is similar to the determination of the surface normal based on density gradient [35]. The application is particularly interesting as this is one of the first attempts to capture fracture in the Material Point Method without explicit definition of the fracture surfaces.

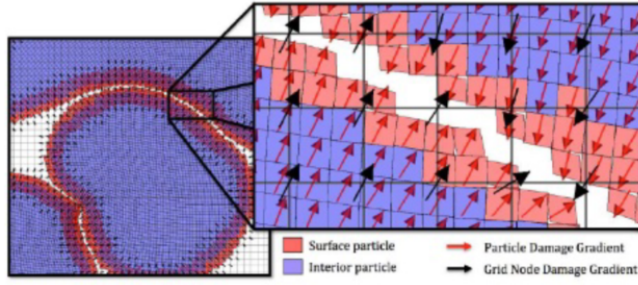


Figure 3.2: Partitioning of material points into different velocity fields based on damage gradient [2].

Particle-centered kernel sampling functions were used to construct a smooth scalar damage field. These are used in other meshless methods for constructing fields and evaluating gradients, where no grid is available to serve as a reference. A similar approach was used in this thesis to evaluate the mass density field produced by different material point generation schemes (Chapter 4).

3.1.4. OTHER APPROACHES

In addition to the contact algorithms shown in the previous section, different approaches for improving the accuracy and efficiency of contact simulation.

TIED INTERFACE GRIDS

Lian *et al.*, [3] proposed an approach which bears similarity to localised mesh refinement as used in FEM. It aims to reduce computational and memory requirements by avoiding a single refined mesh across the entire domain. The method proposed uses one or more regular sub-meshes of different cell size. Particle spacing varies according to the cell size (half of cell size). The interaction between adjacent grids takes place through a "tied interface" - a series of overlapping nodes belonging to the two grids (Fig. 3.3). As particles migrate from coarse to fine mesh numerical fracture may occur, that is particles may become separated by one or more empty elements. This element(s) will then have free degrees of freedom and may prevent neighbouring material points from interacting. To detect and handle the contact between bodies, the frictional contact algorithm proposed by Huang *et al.*, [23] is used.

A moving mesh variant of the approach is also presented where a refined mesh (penetrator) moves into the coarse background mesh. Due to the need for conformity between the refined (slave) and regular (master) meshes, the moving grid may only displace an integer times the cell size of the coarse grid. Once the refined grid is moved, the background is reset and the tied interface re-established. While this presents an interesting opportunity, it was not pursued further due to its perceived complexity. In order to implement such an approach, the requisite components must be present such as the working contact algorithm and meshing schemes.

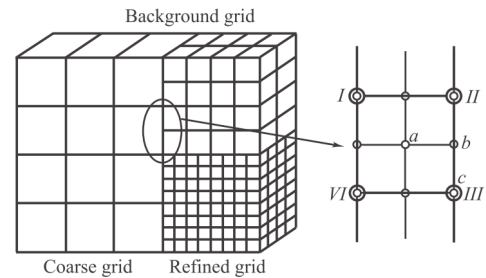


Figure 3.3: Tied interface grid discretisation schematic [3].

FEM-BASED MEMBRANES

Improving the stress recovery in the Material Point Method is currently an area of active research. In a recent work, Hamad *et al.*, [4] proposed embedding 2-dimensional FEM membranes in a 3-dimensional MPM calculation grid in order to enhance the accuracy of stress and strain calculations for structures interacting with soil. A proposed application for the approach is the modelling of geotextile containers.

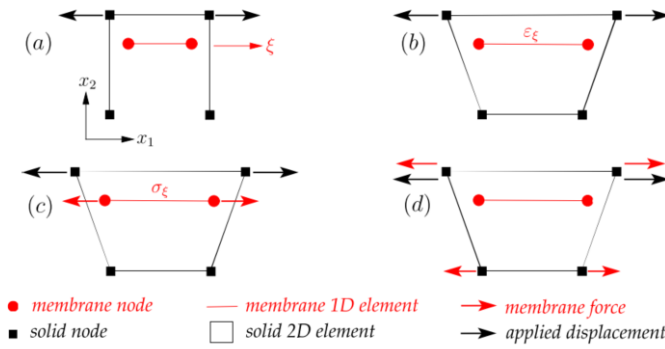


Figure 3.4: FEM-membrane approach summary [4].

the next calculation step (Fig.3.4). A key feature of this approach are the membrane elements used which do not have an out-of-plane stiffness. Strains and stresses for these are evaluated in a local element coordinate system by using a transformation, akin to a corotational FEM formulation.

COUPLED FEM-MPM APPROACHES

Several research teams have recognized potential improvements in accuracy by utilizing a combined FEM-MPM discretisation for portions of the computational domain. Lian *et al.*, [41] introduced a Coupled Finite Element-Material Point (CFEMP) approach, where an FEM-discretised domain interacts with an MPM domain through the background grid. Both of these are explicitly defined at the start of modelling. Alternatively, Lian *et al.*, [42] presented an Adaptive Finite Element-Material Point (AFEMP) approach starting with a standard finite element discretisation. During the analysis excessively distorted elements were then populated with material points to prevent instabilities from mesh tangling.

Both CFEMP and AFEMP model contact between MPM particles and FEM nodes. Following the discussion on Node-To-Node discretisation schemes in Chapter 2, we see that this approach requires a high degree of mesh consistency. Contact conditions are satisfied only at nodes, but not at finite element faces. Ensuring mesh consistency may lead to one region being over meshed and increase computational costs. Later, Chen *et al.*, [6] developed improved versions of the two methods (ICFEMP and IAFEMP respectively), based on a Particle-To-Surface contact scheme. recognised the potential improvements in efficiency By employing a particle-to-surface contact model They introduced two options for tackling the problems arising from non-consistent meshing - ICFEMP and IAFEMP. Adopting a Particle-To-Surface approach, eases the definition of the normal used in contact. A computationally costly global search stage is performed to identify potential contact pairs (point - segment). Once this is done, the normal projection of the point on the surface is found iteratively and the gap calculated. Contact forces are applied based on the magnitude of the gap following the contact conditions outlined in Appendix A.

DEFORMABLE BODY TO RIGID SURFACE INTERACTION

This scenario has been described in literature as being relatively trivial with respect to the others [21]. This can be achieved by setting appropriate boundary conditions which may or may not include friction. An additional advantage would be the ability to define the contact normal based on the rigid surface.

3.2. CONTACT DETECTION

Computational contact mechanics has been an area of active research over the past decades, due to the increasing complexity of numerical simulations for industrial and scientific applications [21].

Displacements for the membrane nodes are found from the nodal displacements of the background grid, in this sense they are treated as material points. From here, strains and subsequently stresses can be evaluated at the integration points of the membrane elements. These are then integrated using the standard Gaussian Quadrature to determine the nodal forces at membrane nodes, which are mapped back onto the background grid nodes and used in

An overview of literature results in a large number of approaches for implementing contact between rigid and deformable surfaces and solids. The latter are particularly interesting for simulating soil-structure interaction in geotechnical engineering and will be explored in further detail. The Material Point Method has the implicit ability to handle no-slip contact between bodies. Not accounting for friction however may cause inaccuracies in many practical applications [9, 10], particularly in penetration problems.

3.3. CONTACT GEOMETRY

As discussed in Chapter 2, depending on the discretisation used contact may be Node-to-Node, Node-To-Surface or Surface-To-Surface in the context of the Finite Element Method [18, 21]. Adapting this vocabulary to the Material Point Method, we can easily imagine Point-To-Point and Point-To-Surface contact to the Multi-Velocity Field-based approach and a MPM-FEM coupling respectively. The latter detects interaction between finite element segments and material points, allowing for more flexibility in the choice of element size.

3.3.1. POINT-TO-POINT

Contact in MPM can be resolved on the background computational grid. From that perspective contact can occur in two general ways. These will be referred to as *odd* and *even* contact in this work. In the *even* case, the contact interface has a thickness of one cellsize¹ - the *contact cell*. As a result, contact forces and impenetrability are imposed on all nodes of the *contact cell*. Resolution of contact in this way may introduce errors due to the calculated normal, specifically for the nodes of the *contact cell* internal to the body. The two possibilities are shown in Figure 3.5. Essentially, the *even* contact case is a violation of the impenetrability condition. As a remedy to this problem, Ma et al., [33] proposed an improvement to the contact logic, based on the distance between the closest material points in the body. This works to limit contact resolution to instances, where the gap is smaller than a predefined margin. An alternative treatment which results in remediation of the problem is the penalty function approach implemented in Ma et al., [1].

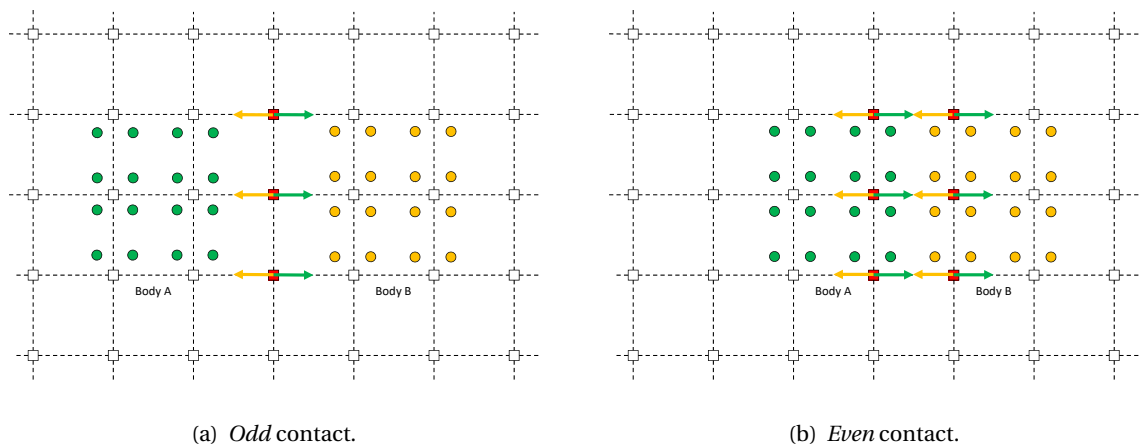


Figure 3.5: Illustration of different contact modes based on detection.

For "odd" contact, the interface has a thickness up to two cellsizes². In this case, contact forces are applied along a "contact line", hence problems with free separation are unlikely. Figure 3.5

¹For a square grid of 4-node quadrilateral elements. This could be greater in cases where the bodies travel along the cell diagonal. Additional considerations apply for different element types and side ratios.

²Again, this may be greater for elements with different side ratios.

shows the normal directions at the contact nodes, assuming these are calculated perfectly for this scenario. This will not always be the case.

3.4. MULTI VELOCITY FIELD

The multi-velocity field-based algorithm (also referred to as multi-mesh) was first proposed by [Bardenhagen *et al.*](#) and has since been implemented by a number of researchers [1, 6, 7, 23, 41, 46]. At its core, it allows for efficient detection of potential contact by utilising the background calculation grid, yielding performance linear in the number of bodies. The Updated Lagrangian phase of MPM is where the deformation and contact resolution occurs. In this sense the Material Point Method corresponds to a UL-FEM formulation, allowing a comparison to be drawn. The single reference mesh that contacting bodies use mimics a Node-to-Node (NTN) contact description. [El-Abbasi and Bathe](#), [47] discuss the necessary conditions to be fulfilled by successful contact algorithms - both of which hold for NTN discretisations.

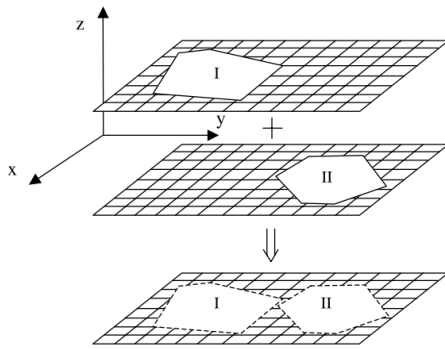


Figure 3.6: Schematic of the multi-velocity field mapping [5].

The approach relies on mapping state variables (momentum and mass) calculated for individual bodies onto the background grid. At nodes where more than one body contribution has been mapped, contact may occur. Fig. 3.6 shows a schematisation of the approach. The "multi-mesh" scheme has been employed in MPM simulations of contact and penetration, as well as coupled FEM-MPM analyses [6, 41]. The latter was named *Coupled Finite Element Material Point Method* (CFEMP) and uses the same approach, comparing velocity contributions from material points and FEM nodes mapped onto the background grid to detect potential interaction. The approach however requires a high degree of mesh consistency between the Finite Element domain and global grid. Where this condition is not satisfied, interpenetration may occur.

3.4.1. PROCEDURE

The discussion below presents the handling of contact assuming the body boundaries and corresponding normals are known. In practice determining these can be challenging, computationally expensive and lead to errors, hence will be more rigorously treated in separate discussions.

The unaltered MPM algorithm resolves the motions and deformations of bodies in a single-valued velocity field (v_i where $i = 1, \dots, N_n$). This field is interpolated from all particles making up the simulation (N_p) and is sometimes also referred to as the *centre-of-mass* velocity (v_i^{cm}). The same procedure is repeated, looping over the particles making up each body (N_b) to resolve its motion ignoring the others. The result is a set of nodal velocities v_i^b , that is each node i of the mesh has a velocity associated with one of the bodies b . In reality, a node is only affected by particles located in one of the neighbouring cells, i.e. the shape functions have compact support. This makes **only** nodal velocities in the vicinity of the body nonzero³.

Contact forces and constraints are imposed at node i only if body b is moving into the interpolated centre of mass in the direction normal to the contact surface.

³Subsequently authors have proposed shape functions with extended support to alleviate issues with cell-crossing errors, however these are only for stress recovery [28, 29, 39].

$$\left(v_i^b - v_i \right) \cdot n_i^b > 0 \quad (3.5)$$

A addition was proposed in a later work, imposing contact only when the surface traction at the node is found to be compressive [45]. Once the above condition is met ⁴, the body velocity must be adjusted to a new value $\bar{v}_i^b$. This needs to ensure the normal component of the body velocity is equal to the normal component of the centre-of-mass, i.e. the body is not penetrating the contact surface, but rather moving with it.

$$\bar{v}_i^b \cdot n_i^b = v_i^b \cdot n_i^b \quad (3.6)$$

$$\bar{v}_i^b = v_i^b - \left[\left(v_i^b - v_i \right) \cdot n_i^b \right] n_i^b \quad (3.7)$$

NORMAL FORCE

The contact force on node i of body b in the normal direction is then applied using Newton's First Law:

$$f_{norm,i}^b = -m_i^b a_i^b = -m_i^b \left[\left(v_i^b - v_i \right) \cdot n_i^b \right] / \Delta t = m_i^b \left[\left(\bar{v}_i^b - v_i^b \right) \cdot n_i^b \right] / \Delta t \quad (3.8)$$

The mass m_i^b of body b at node i is taken as the contributions of the particles making up the body.

$$m_i^b = \sum_{p=1}^{N_p^b} m_p N_i(x_p) \quad (3.9)$$

TANGENTIAL FORCE

Now that the normal component of the contact force has been established, the tangential component can be expressed. Similar to the approach above, the relative tangential velocity is found.

$$\left(v_i^b - v_i \right) - \left[\left(v_i^b - v_i \right) \cdot n_i^b \right] n_i^b = n_i^b \times \left[\left(v_i^b - v_i \right) \times n_i^b \right] \quad (3.10)$$

Setting the body tangential velocity equal to the centre-of-mass tangential velocity allows the limiting tangential force $f_{stick,i}^b$ to be found.

$$f_{stick,i}^b = -m_i^b n_i^b \times \left[\left(v_i^b - v_i \right) \times n_i^b \right] / \Delta t \quad (3.11)$$

Two states are now identified depending on the magnitude of the friction force - the *stick* state and the *frictional slip* state.

$$f_{fric,i}^b = \frac{f_{stick,i}^b}{\left| f_{stick,i}^b \right|} \min \left(\left| f_{stick,i}^b \right|, \mu \left| f_{norm,i}^b \right| \right) \quad (3.12)$$

Alternatively, the friction force can be re-written:

$$f_{fric,i}^b = -\frac{m_i^b}{\Delta t} \frac{n_i^b \times \left[\left(v_i^b - v_i \right) \times n_i^b \right]}{\left| \left(v_i^b - v_i \right) \times n_i^b \right|} \mu' \left[\left(v_i^b - v_i \right) \cdot n_i^b \right] = -\frac{m_i^b}{\Delta t} \mu' \left[\left(v_i^b - v_i \right) \cdot n_i^b \right] \left(n_i^b \times \hat{\omega} \right) \quad (3.13)$$

⁴Note the strong inequality enforced above, meaning the bodies are allowed to separate freely. If the application of adhesion along the interface is desirable, it can be done by introducing an adhesive force when equality is achieved [7].

Where the *effective* friction ratio μ' and tangential unit vector $\hat{\omega}$ can be expressed as follows.

$$\mu' = \min \left(\mu, \frac{\left| (v_i^b - v_i) \times n_i^b \right|}{(v_i^b - v_i) \cdot n_i^b} \right) \quad (3.14)$$

$$\hat{\omega}_i = \frac{(v_i^b - v_i) \times n_i^b}{\left| (v_i^b - v_i) \times n_i^b \right|} \quad (3.15)$$

3.4.2. INTERACTION WITH FEM DOMAIN

As previously noted, the multi-mesh approach can be applied to interactions between FEM and MPM domains. In this case, the contributions of the finite element nodes are mapped onto the background grid and construct the mass, velocity and momentum fields as needed. Once contact is detected and resolved, the calculated contact force is imposed onto the nodes of the FEM domain.

3.5. POINT-TO-SEGMENT

An alternative approach has been utilised by researchers to couple the analysis of material domains discretised using FEM and MPM. In this method, potential interaction pairs (material point - finite element segment) are identified through a contact searching scheme, e.g. bucket-sorting. Trial contact points are iteratively found as the orthogonal projections of the material point onto the corresponding segment and signed *gap* between them is calculated. If the gap becomes negative, a force is applied at the contact point preventing interpenetration. This is used to adjust the material point velocity and respective nodal velocities of the segment. By definition, using the size of the gap to detect contact precludes a non-physical distance between interacting bodies.

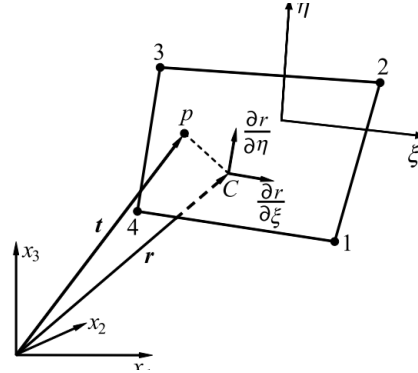


Figure 3.7: Detection of contact between material points and FEM segments [6].

In contrast to Point-To-Segment, the Multi-Mesh scheme may resolve contact when the distance between bodies is about two times the cellsize, depending on aspect ratio, element type and direction of motion. Different approaches have been proposed to include the physical distance between bodies in the contact logic and overcome this limitation [1, 33]. El-Abbasi and Bathe, [47] discuss the suitability of Point-To-Segment schemes, concluding that these require additional treatment to pass the patch test (reproducing uniform pressure).

3.5.1. PROCEDURE

Once potential contact pairs are detected, the distance between material points and the finite element segments must be determined. Segments can be described by the following parametric equation in 2 dimensions (also seen in Figure 3.7 for a 3 dimensional case).

$$r = f_1(\xi, \eta) i + f_2(\xi, \eta) j \quad (3.16)$$

Here, i and j stand for the unit vectors in the along the x and y dimensions, while $f_i(\xi, \eta)$ are the global coordinates of a point (ξ, η) on the segment. These are found using the standard FEM

interpolation from the nodal coordinates.

$$f_i(\xi, \eta) = \sum_{n=1}^4 \varphi_n(\xi, \eta) x_{in} = \sum_{n=1}^4 \frac{1}{4} (1 + \xi_n \xi) (1 + \eta_n \eta) x_{in} \quad (3.17)$$

Using the parametric equation for the segment, the local coordinates of the potential contact point C can be described with respect to the position vector t of a material point in the contact pair. There is one unknown local coordinate in 2 dimensions as the point C lies on the element boundary. An iterative procedure such as the Newton-Raphson method is well suited to solving the equation minimising the orthogonal projection of the vector connecting the material point and contact point for the unknown local coordinate of point C .

$$\frac{\partial r}{\partial \xi}(\xi_C, \eta_C) \cdot [t - r(\xi_C, \eta_C)] = 0 \quad (3.18)$$

Once the local coordinates of C are resolved, the distance from the material point to the segment (gap) can be found.

$$g = n \cdot [t - r(\xi_C, \eta_C)] \quad (3.19)$$

Here the normal to the segment n can be found simply as the difference between the position vectors of the material point and the contact point. Different approaches can be used to determine the derivatives used in the Newton-Raphson scheme here. The complex-step derivative proved to be an attractive method, despite the cumbersome syntax for complex algebra in FORTRAN. The number of iterations required depends on the tolerance defined for the local coordinate. In practice, fewer than 20 iterations were sufficient to determine the contact point with a tolerance of 10^{-6} units when starting with an initial guess at the center of the FE segment. A simple example is presented in Figure 3.8, resulting from a Matlab implementation of the approach summarised above. In this case, the CVT-based discretisation procedure discussed in Chapter 4 is employed.

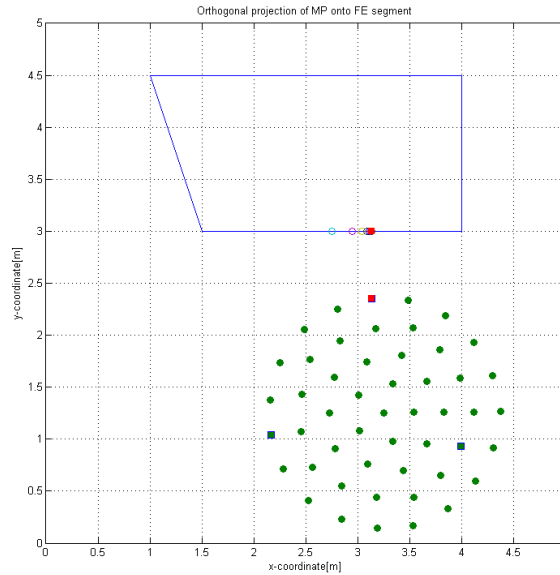


Figure 3.8: Point-to-segment detection for a single element.

Following determination of the gap, contact forces may be applied as needed between the material point and contact location at the element boundary. These are defined in a manner similar to the multi-velocity field algorithm, i.e. a force is applied onto each body in order to achieve continuity of the normal velocity at the contact point which enforces impenetrability.

N.B. The mass interpolated at FE segment contact point will only be influenced by the nodes forming the corresponding edge of the element - all other shape functions are zero. Particular care should be taken with the definition of boundary conditions. Depending on the implemented data structures these affect the way nodal mass is stored in the mass matrix - in the current implementation all fixed DOFs are lumped together and discarded.

3.5.2. ALTERNATIVE

In the case of 2-dimensional simulations with linear quadrilateral elements, the local coordinates of the orthogonal projection may be calculated without the need for iteration. It is known that the contact point will always lie on the edge of a element, hence one of the local coordinates will remain fixed. The other varies linearly between the adjacent nodes. Calculating the area of the triangle formed by the material point and the element edge allows the gap to be determined directly. Subsequently, the local coordinates of the contact point can be easily found. Such an approach can be used in 3-dimensional modelling with linear elements as well, in order to limit the instances of iterative contact point searching. Despite that, the Newton-Raphson procedure as described in the previous section is more flexible and easier to extend to different element types.

As implemented in this work, the bucket-sorting procedure proposed by [Chen et al., \[6\]](#) was not pursued here due to the complex data structure required and the dependence on the chosen bucket size. Instead, the convex hull of the FEM domain was determined immediately after mesh generation, establishing the external segments of the FEM body. As the current implementation did not allow for fracture or mesh refinement, these remained unchanged throughout the analysis. Only the external segments of the FEM mesh need be checked for contact with the MPM body. Further, a check regarding the relative position of a material point with respect to the FE segment was added. This allowed contact resolution to continue only when the material point was found to be within the region defined by the segment x- and y-coordinates. A summary of the approach is shown in Algorithm 3.1 below and a brief description of the employed subroutines included in Appendix C.

- Identify external FEM segments using **convex hull**
- For each time step, loop over the external segments
 - Loop over the material points
 - Check if material point falls between the segment x- or y-coordinates
 - ◊ If yes, find projection of material point on segment
 - ◊ Evaluate the gap
 - ◊ If gap is found negative, resolve contact

Algorithm 3.1: Proposed procedure for detecting Point-to-Surface contact in FEM-MPM interactions.

3.6. THE CONTACT NORMAL AND TANGENT

As discussed previously, the Material Point Method can be thought of as a *meshless* method in the sense that there is no requirement for rigid connectivity between the particles making up the body. This feature gives MPM its relative advantages over mesh-based methods such as FEM in the areas of contact, continuum separation, cracking etc. At the same time, resolving the boundaries and evaluating gradients of state variables for the discretised continuum becomes more challenging. Other meshless methods e.g. the Element Free Galerkin (EFG), use techniques such as kernel sam-

pling to identify and track interfaces [48]. In this respect, the background calculation grid used in MPM allows for a reference against which gradients are evaluated and the identification of different bodies within a domain.

Determination of the body surface normals is an important part of the contact algorithm. It has implications for the effectiveness of contact resolution and the momentum conservation properties of the scheme. Additionally, the tangent used to implement friction between contacting bodies is based on the calculated normal.

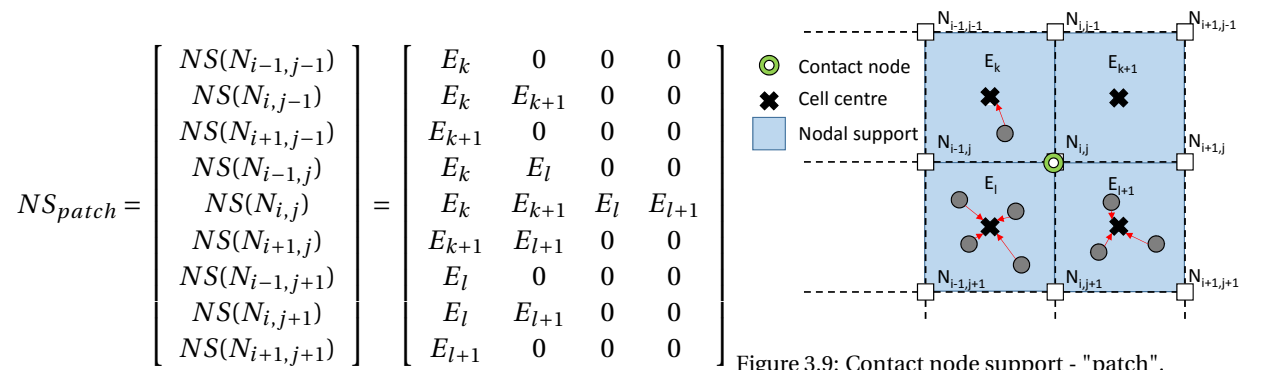
3.6.1. THE NORMAL

In the original implementation of the multi-velocity field algorithm, the gradient of density over the nodal support evaluated at each contact node was used to determine the normal to the body surface. First, density is approximated at the cell centres by using an interpolation function, defined as a product of quadratic B-Splines along the two dimensions. Afterwards, the normal direction is found as the gradient of densities in the cell centres surrounding the node of interest. As a starting point, a bilinear regression can be performed to fit a plane to the density field in the nodal support. From there, determination of the normal direction is a simple matter of taking the reverse of the density gradient.

N.B. In order to satisfy the impenetrability condition (Appendix A) a contact node must fall on or outside the body boundary, i.e. the boundary defined by the active body elements. This means that for a 2-dimensional problem, such a node can have between 1 and 3 active cells in its nodal support. As previously seen, the *even contact* scenario represents a violation of impenetrability and hence may require additional treatment to guarantee satisfactory resolution of the contact.

3.6.2. NODAL SUPPORT ARRAY

Evaluating the normal at a detected contact node requires an approximation of the body surface in the vicinity of the node. This uses information from the nodes in the surrounding elements - the nodal support. As the background grid remains unaltered throughout the calculation, it is convenient to prepare the needed information prior to the time-stepping loop. This was achieved by constructing the *nodal support array*. The size of this array was defined as $[nn, 2^{ndim}]$. Each row of the array stores the element numbers in the support of the node with the corresponding number. Zeros are used to fill positions in the array for nodes with smaller nodal supports (i.e. nodes along the domain boundary). The structure of the nodal support array is illustrated for an element patch in Figure 3.9 below (nodal numbering is done left-to-right and top-to-bottom).



3.6.3. GRADIENT OF NODAL MASS

Authors have proposed using the gradient of nodal mass to evaluate normal direction at the detected contact nodes [1, 23].

$$n_i^b = \sum_{p=1}^{N_i^b} G_{ip}^b m_p^b \quad (3.20)$$

Here, G_{ip}^b are the spatial derivatives of shape functions for node i w.r.t. body b and material point p . This represents the summation of shape function gradients weighted by the particle mass, carried out over the elements within the nodal support of the contact node. The computational cost of the approach is relatively low as the functions and required information are readily available. Figure 3.10 shows the normal and tangential directions calculated using the mass gradient approach for a quarter-circle domain with different distributions of material points. In both cases, good agreement is achieved between the calculated (shown in red) and true (green) directions.

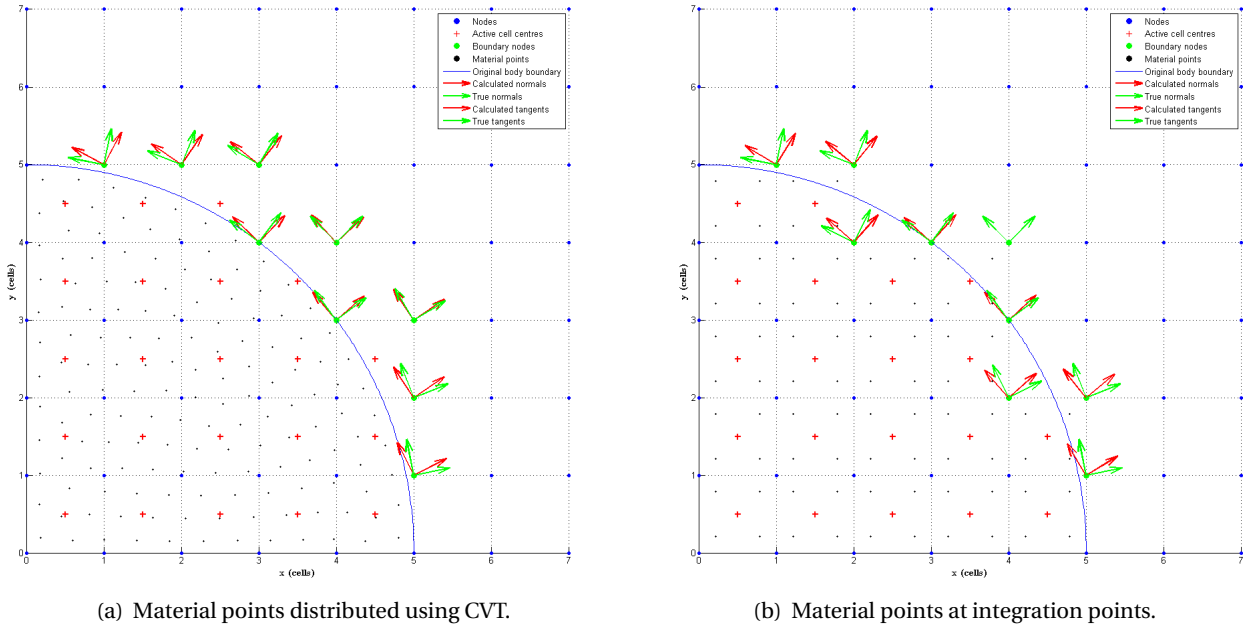


Figure 3.10: Normals and tangents using Mass Gradient for a quarter-circle discretised by different approaches.

3.6.4. GRADIENT OF DENSITY

The approach initially proposed by [Bardenhagen et al.](#), [35], used the gradient of density interpolated at the contact node. Density was interpolated at cell centres of the nodal support using shape functions (S^2) obtained as products of quadratic B-splines along the two axes $S^{(2)} = S^x(x)S^y(y)$ [49]. Each quadratic B-spline had the form show below.

$$S^x = \begin{cases} \frac{1}{2h^2}x^2 + \frac{3}{2h}x + \frac{9}{8}, & \text{if } -\frac{3}{2h} \leq x \leq -\frac{1}{2h} \\ -\frac{1}{h^2}x^2 + \frac{4}{3}, & \text{if } -\frac{1}{2h} \leq x \leq \frac{1}{2h} \\ \frac{1}{2h^2}x^2 - \frac{3}{2h}x + \frac{9}{8}, & \text{if } \frac{1}{2h} \leq x \leq \frac{3}{2h} \\ 0, & \text{otherwise} \end{cases}, \text{ where } h \text{ is the cell spacing in the } x \text{ direction.} \quad (3.21)$$

This allowed the density at cell centres to be calculated as:

$$\rho_e^b \Big|_i = \left[\sum_{p=1}^{N_i^b} m_p^b S^{(2)}(x_c - x_p) / \Omega_c \right] \Big|_i \quad (3.22)$$

It can be noted that this procedure resembles that employed in GIMP for mapping particle information onto grid nodes or in a more general sense - a kernel smoothing procedure.

BILINEAR REGRESSION

The normal to the body surface was taken as the negative of the density gradient. For each contact node, density was approximated at the centres of cells in the nodal support as outlined in Fig. 3.9. The plane was fitted to the generated x-,y- and density data governed by the equation:

$$\rho(x, y) = Ax + By + C \quad (3.23)$$

The fitting was carried out using the linearised Ordinary Least Squares approach after [50].

$$X\beta = z \quad (3.24)$$

$$X^T X\beta = X^T z \quad (3.25)$$

$$\beta = (X^T X)^{-1} X^T z \quad (3.26)$$

Where, β is a vector of the bilinear coefficients and X is given as the following:

$$\beta = [A \quad B \quad C] \quad \text{and} \quad X = \begin{bmatrix} \text{mean}(x^2) & \text{mean}(xy) & \text{mean}(x) \\ \text{mean}(xy) & \text{mean}(y^2) & \text{mean}(y) \\ \text{mean}(x) & \text{mean}(y) & 1 \end{bmatrix}$$

DENSITY GRADIENT

Following the calculations presented above, the normal is established as the negative of the density gradient:

$$n_i^b = \left[n_{i,x}^b \quad n_{i,y}^b \right]^T = - \left[\frac{\partial \rho(x, y)}{\partial x} \quad \frac{\partial \rho(x, y)}{\partial y} \right]^T = -[A \quad B]^T \quad (3.27)$$

Finally, the normal vector was normalised to obtain the unit normal vector $\hat{n}_i^b$.

$$\hat{n}_i^b = \frac{n_i^b}{\sqrt{A^2 + B^2}} \quad (3.28)$$

N.B. It is worth noting that this approach only holds for 2-dimensional problems. In a 3-dimensional situation, the fitting procedure would need to be extended.

3.6.5. GEOMETRIC APPROACH

Pan *et al.*, [46] proposed an approach for calculating the normal using the coordinates of active element nodes within the nodal support (j) and the contact node (i). For bodies 1 and 2 then:

$$n_i = V_i^1 - V_i^2 = \sum_j \left(x_i^1 - x_{j,c}^1 \right) - \sum_j \left(x_i^2 - x_{j,c}^2 \right) \quad (3.29)$$

Effectively, this represents a very simplified approach and also guarantees collinearity of normal vectors for the interaction. It does not account for the location of material points within the elements of the nodal support. Despite this, it may be used to treat contact regions problematic for the other approaches.

3.6.6. THE TANGENT

Once the normal direction is established, the tangent is found as its perpendicular. The rotation matrix can be employed in this regard:

$$t_i^b = R \cdot n_i^b = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}, \text{ where } \theta = 90^\circ \quad (3.30)$$

Rotating the normal 90° in the counter-clockwise direction establishes a consistent sign convention. This represents a simple way of obtaining the tangent, however case must be taken with the sign of the applied tangential contact force. Once the differential velocity $d\mathbf{v}$, i.e. the difference between the body- and center-of-mass velocity fields at the contact node, is found it can be projected onto the surface normal. From there, the differential normal velocity $d\mathbf{v}_n$ can be calculated using the normal and subtracted from $d\mathbf{v}$ to give the velocity difference in the tangential direction $d\mathbf{v}_t$. The key point here is that friction, if present, will act to minimise the absolute value of $d\mathbf{v}_t$ and its sign should be defined accordingly.

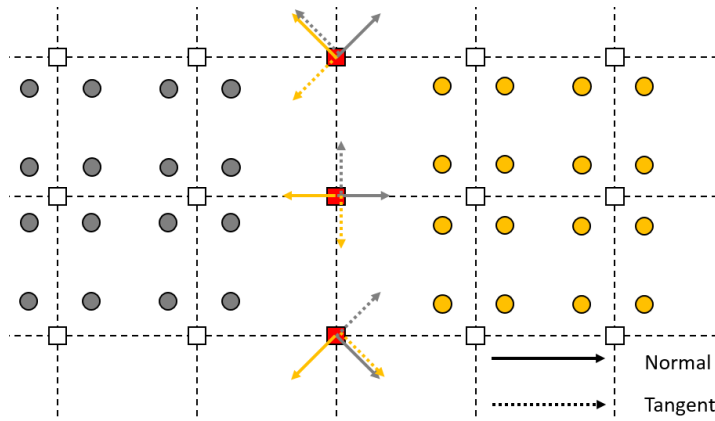


Figure 3.11: Normal and tangent directions at corners.

The tangent at *corner* nodes suffers from the same issues as the normal. It should be noted that for *corner* contact nodes, that is nodes with one *active* element in their support at which contact is detected may cause problems. In general, the normal to a surface is not uniquely defined at such discontinuities and the same holds true for the tangent [21]. Such scenarios require specific treatment in order to mitigate the associated momentum conservation errors. Figure 3.11 shows the normal and tangential directions for a block collision as found using the gradient of density method without additional treatment. Enforcing the impenetrability condition involves adjusting the contact velocities along the normal and tangential directions. It can be seen that adjusting the nodal velocities for the extreme top and bottom contact nodes will initiate a clockwise rotation for the left body and a counter-clockwise rotation for the right. Collision of elastic spheres has been demonstrated in a number of published papers [12, 46]. In most cases, the initial velocity of contacting bodies is chosen to be along the direction of the contact normal for a corner node thus avoiding errors from the ill-defined tangent. Huang *et al.*, [23] demonstrated a collision between elastic rings moving along the x-direction only. The authors corrected the normal directions for the interaction to guarantee collinearity. Use of a refined grid size also reduces the influence of poorly defined normals at *corner* nodes.

It should also be noted that Fig. 3.11 shows a situation where the particles have retained a shape similar to the original configuration (block) relative to one another. This may not be the case in a general contact situation adding uncertainty in the quality of contact resolution. This discussion is intended as a commentary on the effects of discretisation on the accurate resolution of contact.

3.6.7. IMPLEMENTATION AND COMPARISON

A Matlab script was used to illustrate and later analyse the process of normal and tangent calculation. A quarter-circle domain was discretised using several different approaches and the *boundary* nodes identified. In the case of a single stationary body, *boundary* nodes were those with a nodal support of 4 elements *and* 1,2 or 3 active elements in the nodal support. These are the nodes where contact *could* occur.

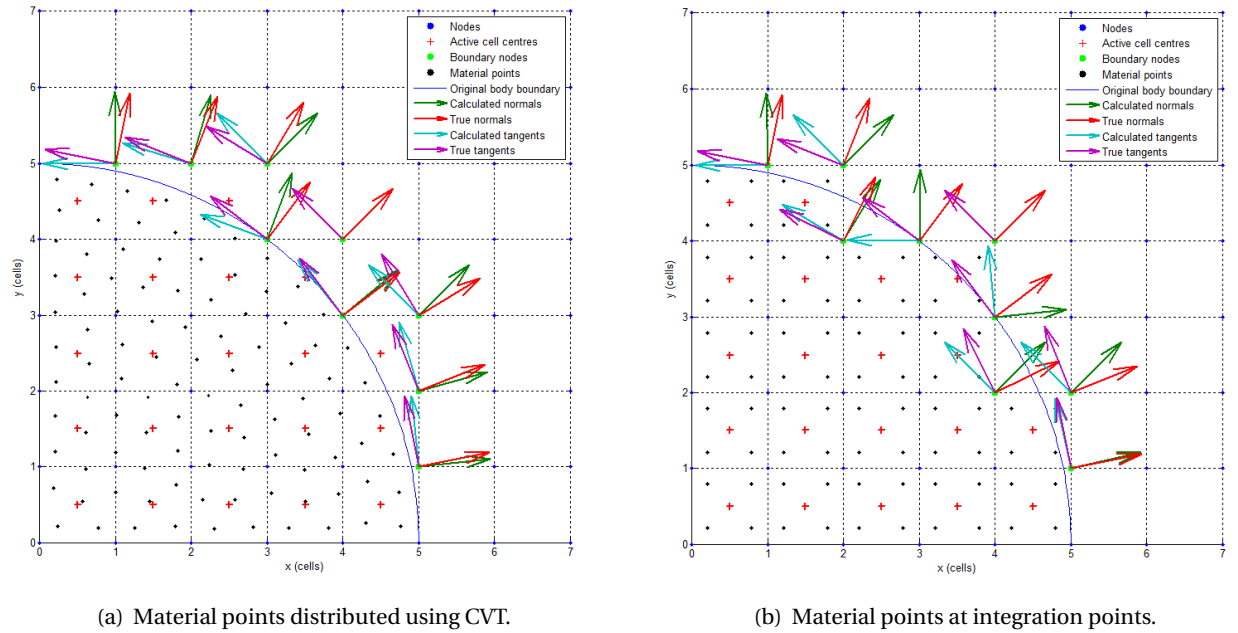


Figure 3.12: Normals and tangents using Density Gradient for a quarter-circle discretised by different approaches.

Figure 3.12 shows the result of normal direction calculation for a discretised domain. A circular arc was used as the *true* vectors are also defined. The normal and tangential directions are not uniquely defined at sharp geometrical discontinuities such as corners hence these have been avoided. In one case, the material points are initiated at the element integration points. This was done using the Local Subdivision approach [44]. The calculated normal and tangential directions at the body surface show larger deviations from the *true* vectors than in the case of the CVT-based discretisation. A more detailed discussion on different ways to generate material points are presented in Chapter 4. Figure 3.12 also serves to illustrate one of the key issues in implementing contact algorithms with MPM - dependence on the background mesh resolution. The original circular arc results in a stepped boundary after discretisation and mapping onto the grid nodes, with steps being geometrical discontinuities. This can affect the contact detection and is influenced by the grid cell size.

Nodes with a single activated element in their nodal support represent a special case. These are termed *corner nodes* in this report. Figure 3.11 highlights the potential problem of using the gradient of density approach for nodes at sharp geometrical discontinuities. Increasing the resolution reduces the overall impact of these nodes, but does not solve with the underlying issue. An approach allowing the effective resolution of these *corner nodes* was needed.

3.6.8. DOMAIN BOUNDARY TREATMENT

While not strictly a part of the discussion on the contact algorithms, some attention should be given to the boundaries of the domain as these may affect the outcome of the explicit analysis. Quasi-static approaches normally adopted in engineering problems are appropriate for slow loading rates or sustained loading, where the goal is to find a steady-state solution (e.g. deflection of a beam under constant load). In contrast, dynamic formulations offer the possibility to capture rapid loading rates and observe wave propagation through the spatial domain. Unless sufficient damping is provided by the material, spurious wave reflection may occur at the fixed domain boundaries and influence the solution. These also introduce a largely non-physical response for geotechnical problems, where waves are damped due to soil material properties and reflection is precluded by the spatial dimensions of the domain (usually treated as unbounded below surface level). To this end, an effective method to mitigate this phenomena was sought.

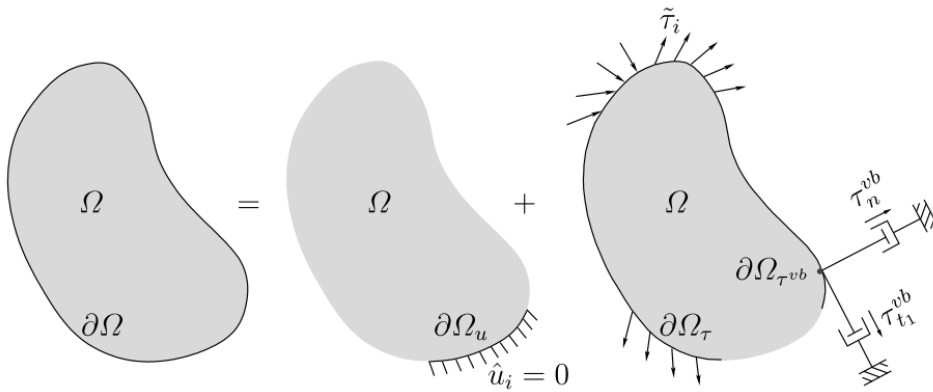


Figure 3.13: Silent boundaries with dashpots [7].

Al-Kafaji, [7] presents a detailed discussion on the topic. Mapped infinite elements, viscous (dashpot) boundaries [51] and Kelvin-Voigt elements [7] are different options to treat wave reflections at domain boundaries. Shen and Chen, [51] discussed viscous damping boundaries in the Material Point Method by defining damping forces applied to material points located adjacent to the fixed boundary.

$$\tau_{p,n}^t = -\rho c_p v_{p,n} \quad (3.31)$$

$$\tau_{p,t}^t = -\rho c_s v_{p,t} \quad (3.32)$$

These could then be mapped onto the grid nodes in a manner similar to other body forces. A notable difference is the presence of the Δx term, referring to the thickness of the boundary layer in the appropriate direction.

$$f_{i,n}^t = \sum_p m_p \frac{\tau_{p,n}^t}{\rho_p \Delta x} N(x_p^t) \quad (3.33)$$

$$f_{i,t}^t = \sum_p m_p \frac{\tau_{p,t}^t}{\rho_p \Delta x} N(x_p^t) \quad (3.34)$$

Essentially, these are local approximate non-reflective boundary conditions. They have been reported to treat waves with normal incidence almost exactly, with effectiveness degrading as the

incidence angle decreases. Still, the approach presents a simplistic treatment for silent boundaries. [Al-Kafaji, \[7\]](#) proposed an extension to the method, substituting the dashpots for Kelvin-Voigt elements to prevent creep of the boundaries under sustained loading. The formulation remains similar to the viscous boundaries in Equations 3.34 and 3.34.

$$\tau_{i,n}^{vb} = -\alpha \rho c_p v_n - \kappa_p u_n \quad (3.35)$$

$$\tau_{i,t}^{vb} = -\beta \rho c_s v_t - \kappa_s u_t \quad (3.36)$$

The coefficients α and β shown above are normally taken as 1, which guarantees effective damping across a wide range of incidence angles. As can be seen from the formulations presented above, the damping forces for the silent boundaries can be found as part of the initial *Forward Mapping* stage and added to the solution in the *Lagrangian Deformation* stage.

$$F^{vb} = C v + K^c u \quad (3.37)$$

Procedure for implementation:

- Identify absorbing boundary nodes using **boundaries** subroutine
- Free the DOFs of the absorbing boundary nodes members of active elements for each body
- Assemble global steering vectors for body solutions
- Impose forces at absorbing boundaries from Eq. 3.37

Algorithm 3.2: Procedure for absorbing boundary implementation.

4

MATERIAL POINT GENERATION

There is no *standard* way to discretise domains into material points. A practical way to achieve this is generating finite elements across the domain and then placing MPs within them according to a specific rule. Indeed, many schemes have been developed for generating meshes in the Finite Element Method and MPM is well suited to make use of these. Placing material points at a uniform spacing becomes a complex problem for irregular geometries and higher dimensions.

In the MPM, material points are able to move through the computational grid and travel a significant distance away from their original locations. The distinction between material and mesh allows permits the use of structured calculating grids for many applications of the Material Point Method, while giving more freedom to the user for the material point generation process. An interesting example is placing material points onto pixels or patches of an image.

4.1. CELL SUBDIVISION

Material points can be initiated by placing these at the integration points of finite elements used to construct the background grid. Provided this mesh is also used in calculations, the material points will initially be located at the points of optimal quadrature giving a solution identical to FEM in the first time step. Standard FEM elements however, have an associated number of integration points and if more material points are required, additional steps must be taken to produce these. The mesh can be refined by subdividing cells into smaller regions and using their integration point locations.

A potential drawback of this approach is the fact that it may not preserve the contour of the original body when using a structured mesh in the generation procedure. Where the shape need be reconstructed faithfully, a larger area could be populated with material points and those not within the boundary discarded. This does not guarantee a reasonable distribution of material points particularly in the vicinity of the body boundary. Alternatively, an unstructured mesh can be used to generate material points. This allows for significant flexibility in meshing complex-shaped domains, however elements are not limited to a single size. Consequently, material points generated in this way would have different masses and volumes associated with them, which could introduce further issues with cell-crossing events.

At the start of this work, a local subdivision approach was used to split grid cells and place material points at the Gauss points of the resulting sub-elements [12, 44]. For an example slope stability problem, the mesh and material point generation included a number of steps to initially treat a structured mesh before placing material points at the required locations.

4.2. TESSELLATIONS

Dividing a domain into subdomains or *tiling* is a problem encountered across many fields such as image compression, quadrature, distribution of resources and others [52]. In numerical simulations these are useful for generating calculation meshes over the desired domain. Indeed, Voronoi Tessellations and their dual Delaunay Triangulations are useful tools for generating unstructured meshes. Centroidal Voronoi Tessellations (CVTs) have been reported to provide superior results in terms of mesh quality. These represent tessellations, where borders of regions are constructed to be equidistant from neighbouring generating points. Additionally, when converged the generating points are the centroids of their regions and in 2-dimensions these regions are congruent to a regular hexagon¹. Weighting can also be applied to the generation of CVTs so that the cell areas obey a certain rule, effectively acting as local mesh refinement. Du and Gunzburger, [53] compared the quality of solutions to the Poisson equation over a rectangular domain discretised with a regular Cartesian grid and an unstructured grid generated using CVTs. Observed errors were lower in the simulation using a CVT-based grids and these showed good convergence with increasing number of elements.

MPM can be thought of as using two separate types of discretisation - the material points and the background grid. MPs are used to track the mass and state variables of the material. In the standard implementation of MPM, the density field is discretised with the use of the Dirac delta function, thus giving the material points a point-wise spatial domain. Initiating material points at the centroids of the tessellation cells allows the desired density field to be reproduced (be it uniform or other). A widely used approach for generating CVTs is Lloyd's Algorithm. This is an iterative procedure that partitions a domain into n sectors starting from a set of n generators. Several other approaches have been explored by researchers to speed-up convergence for larger problems [54], however these were not explored in this work. Alternatively, a Monte-Carlo based scheme can be used, which produces a large number of sampling points within the target domain and calculates the distances to the generators. The generator locations are then adjusted in order to reduce the *energy* of the point set, i.e. the integral of the squared distances between sampling points and generators over the domain. An external library of FORTRAN subroutines² was incorporated in the code and used to produce CVT tessellations and initiate material points.

Figure 4.1 shows the two approaches used to distribute 100 material points within a circular domain. In the Lloyds Algorithm case³ the circle was approximated by a convex 100-gon. The Delaunay triangulations of the generated point sets are shown and coloured to enhance visibility. It should be noted that the CVT-based material point generation scheme may not as effective in domains of regular shape, which lend themselves to straightforward discretisation with structured grids (Fig. 4.2). This can be overcome by generating material points in a *template* element using the tessellation approach and replicating it over the desired domain. An advantage of this would be the ability to select the number of material points per element, unlike subdivision where this is strictly a multiple of 2. Additionally, the arbitrary number of material points would nonetheless be distributed evenly over the element, whereas with subdivision these may inherit a certain directionality if the number of regions defined along the two dimensions are not the same.

4.3. COMPARISON

The material point generation strategies were compared according to two metrics. Firstly, the generated material points were required to accurately represent the uniform density field used in the analyses. The other key feature investigated was the accuracy of the normal and tangential direc-

¹For internal cells, located away from the boundary.

²Produced by John Burkardt at the Florida State University. Source code available at http://people.sc.fsu.edu/~jburkardt/f_src/cvt/cvt.html licensed under the GNU LGPL license.

³Based on the Matlab implementation by Aaron Becker, Univ. of Houston.

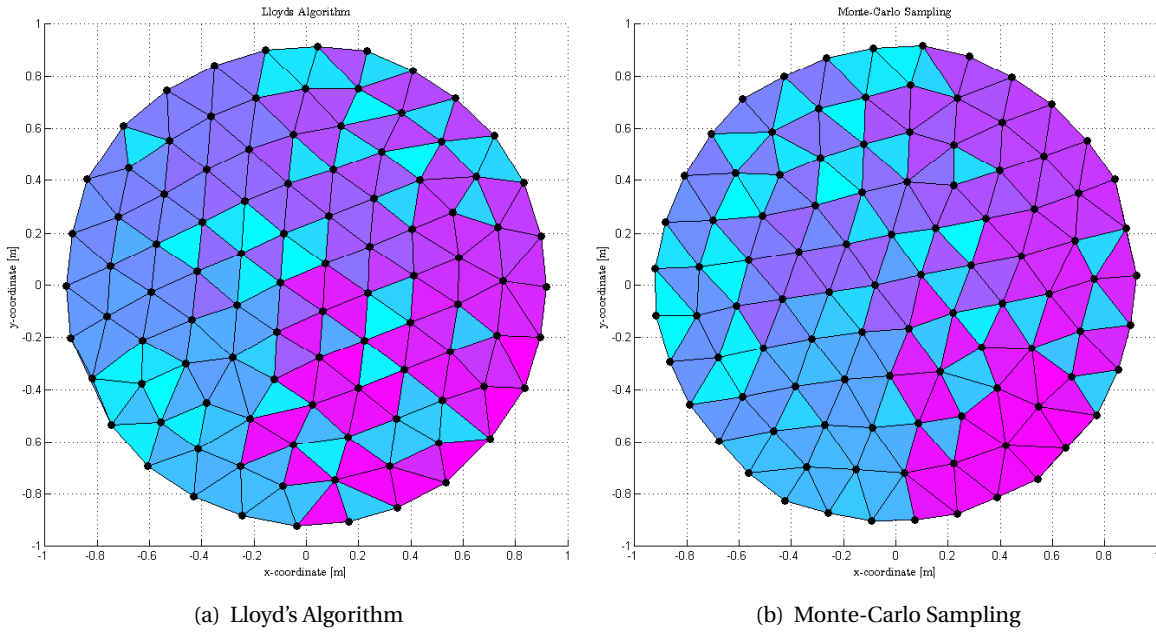


Figure 4.1: 100 points distributed in a circular domain using Lloyd's Algorithm and Monte-Carlo sampling approaches.

tions resulting from the different strategies. A comparison of the mass density field was made using a kernel sampling approach characteristic for meshless methods [48], which coincides with an application of GIMP when sampling locations are chosen as the nodes [55]. Different weight functions can be chosen in order to interpolate mass and respectively mass density. Cubic spline, quartic spline and bilinear weight functions are shown in Equations 4.1, 4.2 and 4.3 respectively.

$$w(r) = \begin{cases} \frac{2}{3} - 4r^2 + 4r^3, & r \leq \frac{1}{2} \\ \frac{4}{3} - 4r + 4r^2 - \frac{4}{3}r^3, & \frac{1}{2} < r \leq 1 \\ 0, & r > 1 \end{cases} \quad (4.1)$$

For a cubic spline weight function.

$$w(r) = \begin{cases} 1 - 6r^2 + 8r^3 - 3r^4, & r \leq 1 \\ 0, & r > 1 \end{cases} \quad (4.2)$$

For a quartic spline weight function. Finally, a bilinear weight function is given below. Combined with a rectangular support corresponding to the element size, this produces the same result as the shape functions of bilinear elements.

$$w(r) = \begin{cases} 1 - r, & r \leq 1 \\ 0, & r > 1 \end{cases} \quad (4.3)$$

In all cases input r is found using the Euclidean distance between the kernel center (node) i and the sampling point.

$$r = \frac{\|x_i - x\|}{d_i} \quad (4.4)$$

Weight functions were applied along a single dimension and then combined to form a rectangular support surrounding the node.

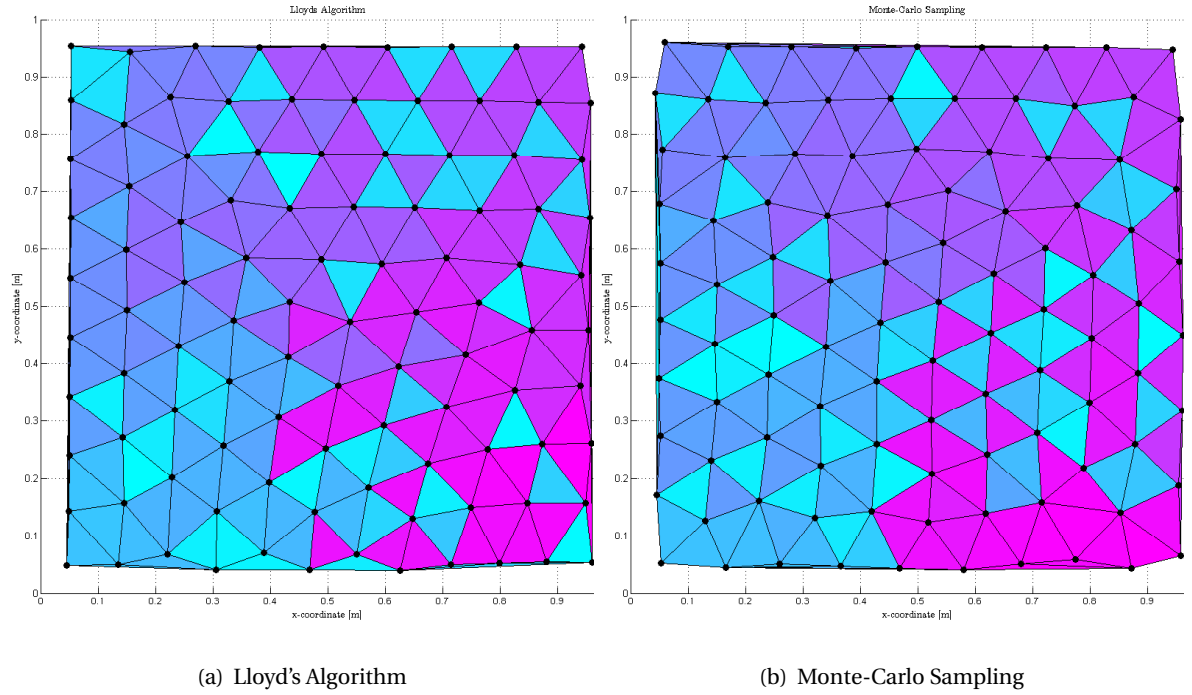


Figure 4.2: 100 points distributed in a square domain using Lloyd's Algorithm and Monte-Carlo sampling approaches.

$$w(x - x_i) = w\left(\frac{|x_i - x|}{d_i^x}\right) w\left(\frac{|y_i - y|}{d_i^y}\right) \quad (4.5)$$

Using the above approach allows mass to be sampled in different regions (patches) of the domain. This was then used to find a density associated with the centre of the patch. A square kernel was used to approximate a patch of elements belonging to the background grid. Figure 4.3 shows the results from density sampling with patches centered at the original mesh nodes. [Homel and Herbold, \[2\]](#) used a similar sampling approach to construct a continuous damage field. In contrast, here sampling centers are intentionally not taken at material point locations as the interest is to evaluate the quality of the field resulting from interpolation to the grid.

The density field interpolated at the node locations showed good agreement between the two approaches. Sampling for nodes at the domain boundary yielded lower density as there were fewer material points located in the corresponding patches. A similar procedure can be adapted for sampling at random locations. This is intended to model scenarios in which the material points have displaced through the mesh, away from their original positions. To this end, a set of quasi-random sampling locations were generated using the Halton sequence. Figure 4.4 shows the results from this sampling. Material point locations are not shown, however these correspond to the ones in Fig. 4.3. In both figures, 400 material points were distributed on a unit square with each point assigned a mass of 1 unit.

The sampling program was carried out on basic shapes in order to compare the schemes. Table 4.1 shows the results from sampling the density field resulting from a discretised square domain. Increasing the number of sampling points and distributing them in a quasi-random fashion (indicated as Halton sampling), showed similar results as sampling at the nodal locations with mean density closer to the true value for both generation procedures. At the same time, a higher density was observed around the center of the domain, which was considered non-physical.

In Chapter 3 different approaches to calculating the surface normal direction were shown and

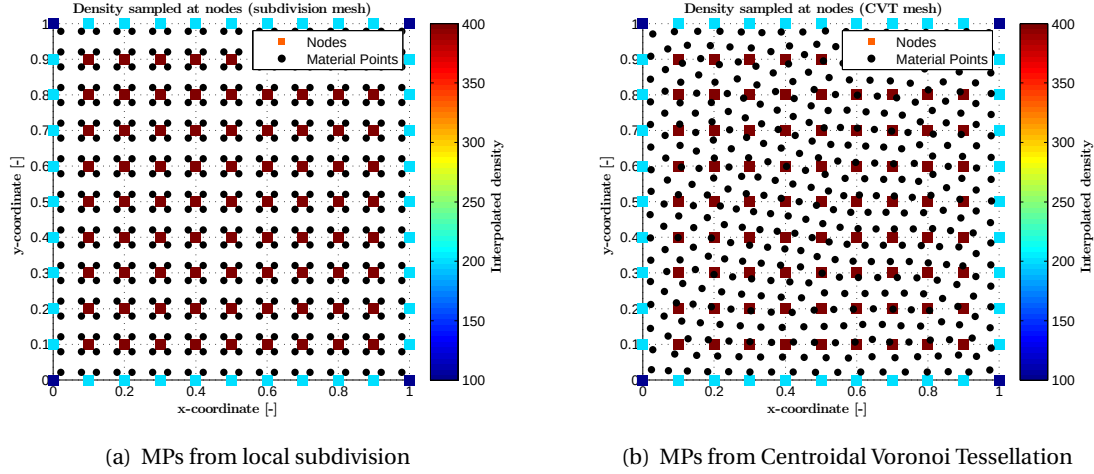


Figure 4.3: Density sampling for patches centred at the original mesh nodes.

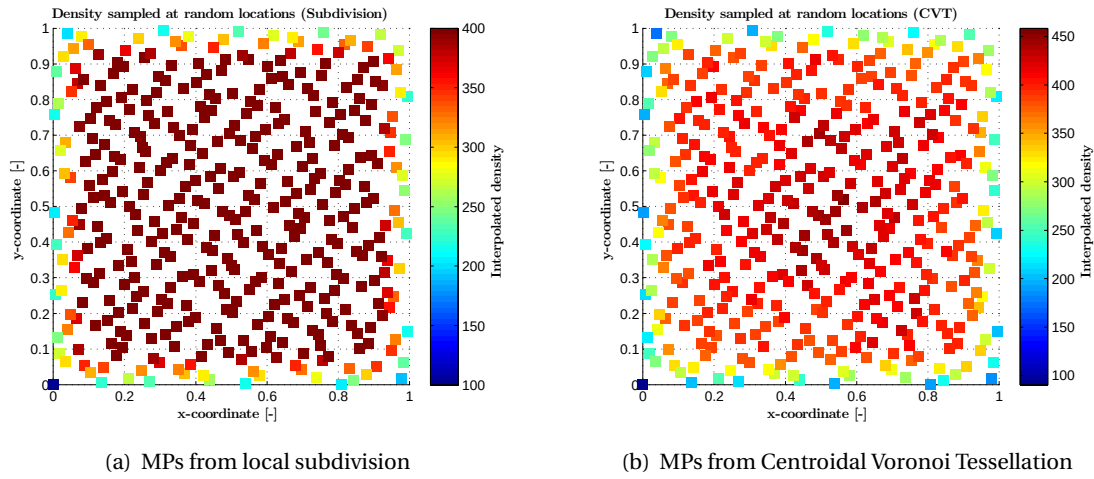


Figure 4.4: Density sampling for patches centred at quasi-random locations.

potential effects of non-collinearity of the contact normals were discussed. In addition to this however, the choice of material point distribution was seen to affect the calculated normal direction for a body. A similar approach was employed to showcase this using a quarter-circle domain.

Figure 4.5 shows a comparison between the true and calculated surface normal and tangent directions for a domain discretised using different approaches. Both approaches resulted in a stepped body boundary as defined by the activated cells (those containing material points). This effect however will become less pronounced with increasing mesh refinement. While difficult to compare, the accuracy of the normal directions calculated for the CVT-discretised mesh show better agreement with the true vectors.

Table 4.1: Density field sampling for a square domain.

Nmpps	Subdivision (Node)		CVT (Node)		Subdivision (Halton)		CVT (Halton)	
	Mean	St.dev	Mean	St.dev	Mean	St.dev	Mean	St.dev
400	330.6	100.7	330.6	110.9	374	51.1	377.1	59.6

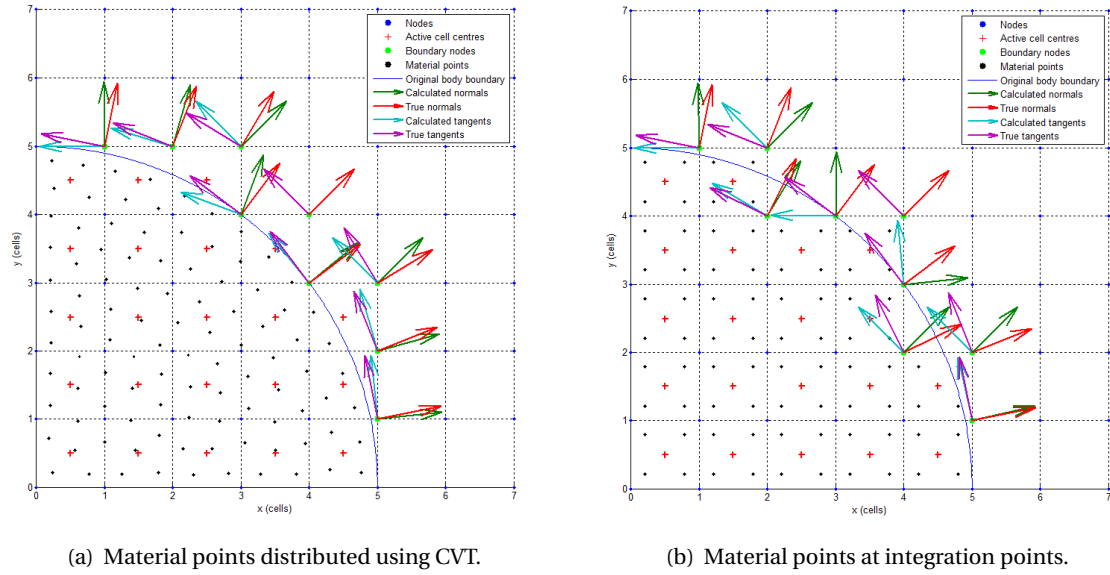


Figure 4.5: Normals and tangents using Density Gradient for a quarter-circle discretised by different approaches.

4.4. APPLICATION

As previously discussed, a major advantage of the CVT-based approach is the simplicity of populating arbitrary-shaped domains with material points. Here, a practical example is presented relevant to a typical analysis scenario encountered in the field of geotechnical engineering - discretising a slope.

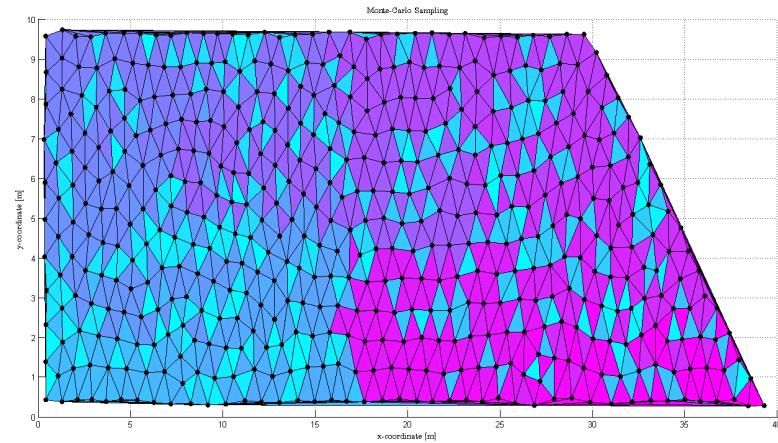


Figure 4.6: Slope mesh discretised with 500 material points.

Using local subdivision to generate the material points for the slope requires a finite element mesh to be produced initially, which can be a challenging task for structured meshes and irregularly shaped domains. In contrast, the CVT-based procedure requires only a definition of the region to be populated with MPs. This can be done either in terms of a bounding polygon given as a set of vertices or a set of equations governing the distribution of sampling points. The above example was produced with the coordinates $(0,0)(40,0)(30,10)(10,0)$. It should be noted that more iterations

could be performed to produce a better distribution of material points.

4.5. CONCLUSION

The implications of using a CVT-based approach to discretise continua were investigated. It was shown that this approach provides a flexible, adaptable method for generating equally spaced material points. In the *initial* configuration, i.e. when evaluating a field at the nodal locations, both approaches yielded satisfactory results.

5

IMPLEMENTATION AND BENCHMARK PROBLEMS

The algorithms discussed in previous chapters were implemented within a dynamic, explicit MPM-framework. These have been applied to several widely used benchmark tests - axial vibration of a bar [37], impact between elastic (or elastoplastic) bodies [23], ball rolling/sliding on a plane [45, 46]. An additional example showcasing the coupled FEM-MPM approach implemented is given as a beam deflecting under the weight of a deformable block, which slides off once the beam is sufficiently inclined.

The MPM FORTRAN code developed makes use of the subroutine libraries *main* and *geom* for their mesh-generation capabilities, definitions of the finite elements, shape functions and other auxiliary functions. These are available as an addition to Smith *et al.*, [8]. Further details regarding the computational implementation can be found in Appendix C.

5.1. DYNAMIC EXPLICIT MPM FORMULATION

Initially, the MPM formulation used was applied to a simple dynamics problem in order to understand its limitations and sources of error for subsequent analyses. The ability of the code to model free motion through a domain and interact with the boundaries were investigated. Further, the energy conservation properties of the scheme were tested. Free vibration of mass-spring systems or axial bars provide good insights and allow comparison to other works [7, 24, 37].

5.1.1. INITIAL STRESS ASSIGNMENT

Initiating bodies in the simulation in equilibrium between internal and external forces allows modelling to commence without changes to the initial geometry due to self-weight deformations. Similar approaches are implemented in a number of Finite Element Programs, in particular for geotechnical engineering analysis this is referred to as the *K0-procedure*. Similar to a step within an implicit formulation, initialisation of stresses can be done via an initial stage with gravity-only loading and local non-viscous damping applied. The end of this stage is defined by the out-of-balance forces and a ratio between kinetic energy and the work done by external forces [12].

$$\epsilon = \frac{||F^{ext} - F^{int}||}{||F^{ext}||} \quad (5.1)$$

$$\delta = \frac{E_{kin}}{W^{ext}} \quad (5.2)$$

Alternatively, material point stresses can be assigned directly, based on location, material properties and loading (.e.g. gravity). This has to be tailored to different bodies, depending on their boundary conditions.

5.1.2. ENERGY CALCULATION

The kinetic, strain, potential and total energies of the system and/or separate bodies were defined after [12]. The equations used are shown below:

$$E_{kinetic}^{t+\Delta t} = \frac{1}{2} \sum_p m_p \left(v_p^{t+\Delta t} \right)^2 \quad (5.3)$$

$$E_{strain}^{t+\Delta t} = \sum_p V_p^{t+\Delta t} \left(\frac{\sigma^{t+\Delta t} + \sigma^t}{2} \right) : \Delta \epsilon \quad (5.4)$$

$$E_{potential}^{t+\Delta t} = \sum_p m_p g h_p^{t+\Delta t} \quad (5.5)$$

$$E_{total}^{t+\Delta t} = E_{kinetic}^{t+\Delta t} + E_{strain}^{t+\Delta t} + E_{potential}^{t+\Delta t} \quad (5.6)$$

In the case of perfectly elastic materials in a purely mechanical analysis, the total energy of the system as presented above is expected to remain constant. Friction represents a dissipation of energy and its inclusion would register as a net decrease in system energy where friction forces are acting.

POTENTIAL ENERGY FOR INCLINED PROBLEMS

In problems where friction between a sloped plane and an object was modelled, the inclination was modelled by directing the gravity vector at an angle w.r.t. the vertical axis of the domain. The calculation of potential energy was altered to reflect the changed distance to the datum.

$$E_{potential}^{t+\Delta t} = \sum_p m_p g H_p^{t+\Delta t} \quad (5.7)$$

Here, H is the perpendicular from the material point to the datum, selected at corner of the domain closest to the tip of the inclination angle.

$$H_p = A \cos \delta + B \sin \delta \quad , \text{where} \quad A_p = \left| n e_y \Delta y - y_p \right| \quad \text{and} \quad B_p = n e_x \Delta x - x_p \quad (5.8)$$

5.1.3. SIMPLE MOTIONS

Initially, the ability of the MPM implementation to model a body moving freely through a domain was established. Two scenarios were tested - one under an initial velocity without gravity and one "freefall" case. In both situations the body was allowed to interact with the domain boundary in order to verify the correct application of boundary conditions.

Comparison with the analytical solutions from show that the implementation is able to accurately resolve the kinematics of the moving material domain. Additionally, the energy calculation scheme could be verified. Originating from a Particle-In-Cell method, MPM can handle the convection of material through the domain without specific treatment and this was clearly seen in the examples.

5.1.4. AXIAL VIBRATION OF A BAR

Modelling the behaviour of a vibrating axial bar is a commonly used benchmark in mechanics. The problem can be simplified to a 1-dimensional scenario and has an analytical solution for small deformations. Its analysis can yield valuable insights into the energy conservation properties of the

numerical method used, as well as the influence of the time step size on stability and accuracy [37]. Further, in subsequent analyses contacting bodies are subjected to shock waves, which may lead to oscillations. quasi-static analyses, vibrations are damped out during the iterative solution procedure and the response is dominated by the first mode.

	# Elements	# MP/El.	E [Pa]	ν [-]	ρ [m/kg ³]	Δy	Δx	Δt	#Steps
Bar	25 (1x25)	4	100	0.0	1	1	1	0.001	9000

Table 5.1: Configuration and material properties for axial bar vibration.

As shown in Figure 5.1, the numerical model is able to reproduce the energy of the vibrating bar well, despite the total energy showing oscillation. Some energy dissipation is observed attributed to the numerical scheme employed. The analytical solution to the problem as well as a more extended discussion on the energy conservation of the method was included in Appendix B.

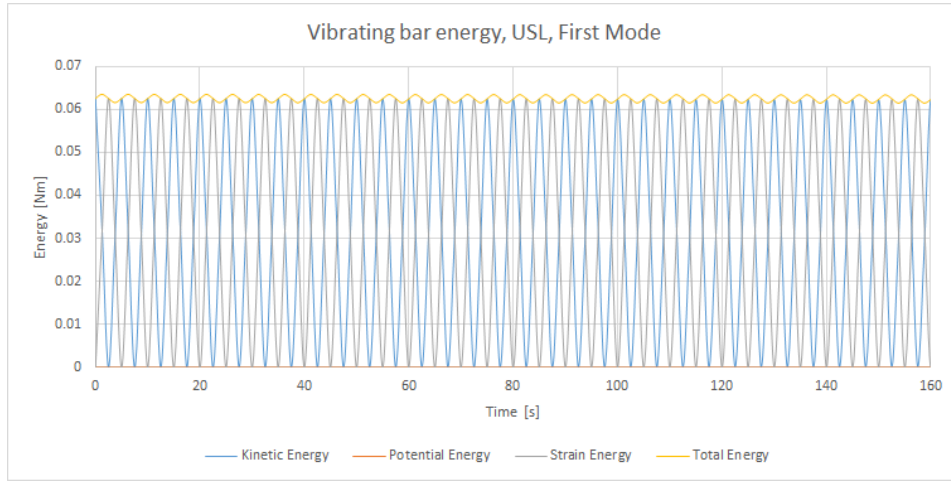


Figure 5.1: Energy of the vibrating bar (First mode of vibration)

The dissipation of energy observed for vibrations with wavelength approaching the grid cell length (higher modes of oscillation) will become important in subsequent discussions, on contact and impact between bodies. Bardenhagen, [37] analysed the energy conservation errors for different time integration schemes, concluding that the Update-Stress-Last scheme used here dissipated energy over time. This was corroborated by the results produced with the implemented code, however the USF response did not exhibit the expected characteristics. Comparison with the MUSL approach, showed a superior results for energy, displacement and velocity when using the modified procedure even for the fifth mode of oscillation.

Analysis using the USL and USF methods was able to complete with time step sizes up to $8.25 \times 10^{-2}s$, corresponding to 82.5% of the critical time step determined using the CFL condition. The results however were inadmissible as the strain energy calculated showed non-physical behaviour, expressed in a significant jump, completely altering the response of the system. Good results were produced using time step sizes smaller than $5.0 \times 10^{-2}s$, or 50% of the critical time step. Even in simplified problem such as 1-dimensional continuum bar, relatively small time steps were required to produce a reliable analysis. This is an important note for subsequent analyses, where the time steps used will be kept lower for the purposes of stability and accuracy.

5.2. CONTACT PROBLEMS

The example problems implemented and presented below were used to investigate the performance of the dynamic explicit MPM framework for contact problems.

5.2.1. ELASTIC BLOCK COLLISION

The collision of two elastic blocks was modelled to assess the effectiveness of the contact algorithm. To this end, the energy of each block was calculated according to the procedure shown in Section 5.1.2. The blocks were initiated with velocities in opposite directions, such that the bodies would impact each other first, then move away and collide with the domain boundary. The aim of the model was to observe the evolution of energy of the elastic body under different kinds of interaction - with a rigid wall (domain boundary) and with a second deformable body. The contact occurred and was resolved only in the normal direction, i.e. there were no contact forces in the tangential direction. As a simplified problem, the effects of additional treatments such methods to reduce the distance between contacting bodies, different variable update schemes and the effects of mesh refinement could be observed.

Body	# El.	# MP/El.	E [Pa]	ν [-]	ρ [kg/m ³]	v_0 [m/s]	Δy	Δx	Δt	#Steps
1	4 (2x2)	16	100	0.1	1.0	1.0	0.25	0.25	0.001	3000
2	4 (2x2)	16	100	0.1	1.0	-1.0	0.25	0.25	0.001	3000

Table 5.2: Configuration and material properties for elastic block collision.

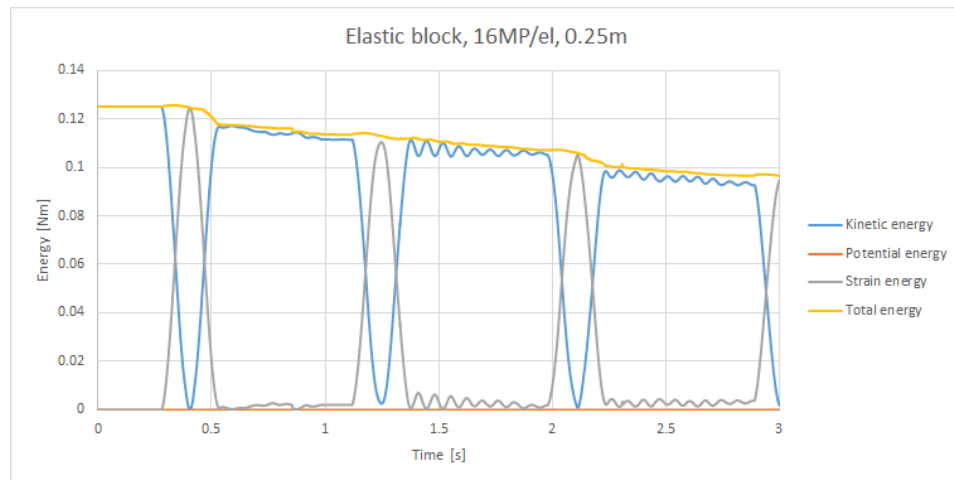


Figure 5.2: Evolution of energy for disk collision with 16MP/el and 0.25m grid size.

Figure 5.2 shows the evolution of energy calculated for the right block. The collisions can be clearly identified by the rise and fall in strain energy. First, contact between the two deformable disks occurs, followed by contact with the rigid boundary and so on. It can be seen that some energy is lost during contact between the elastic bodies. This is not so for the impact with the domain boundary. Prior to the first instance of contact, the energy of the system remains constant as the body translates through the domain without internal stresses present. Following contact, free vibrations as the body moves away result in energy dissipation. Appendix B provides a more detailed discussion on the dissipative behaviour during oscillation.

INFLUENCE OF BACKGROUND GRID RESOLUTION AND MATERIAL POINT NUMBER

Increasing the number of material points and reducing the grid cell size both have a positive effect on overall energy conservation for the contact resolution. To investigate the effect of smaller mesh

elements, further tests were carried out by refining the background grid and increasing the number of material points. Body mass and dimensions were kept constant throughout the two simulations. Figure 5.3 shows four different combinations of cell size and material points per element. Increasing refinement resulted in a smaller dip in total energy for the contact between the two bodies.

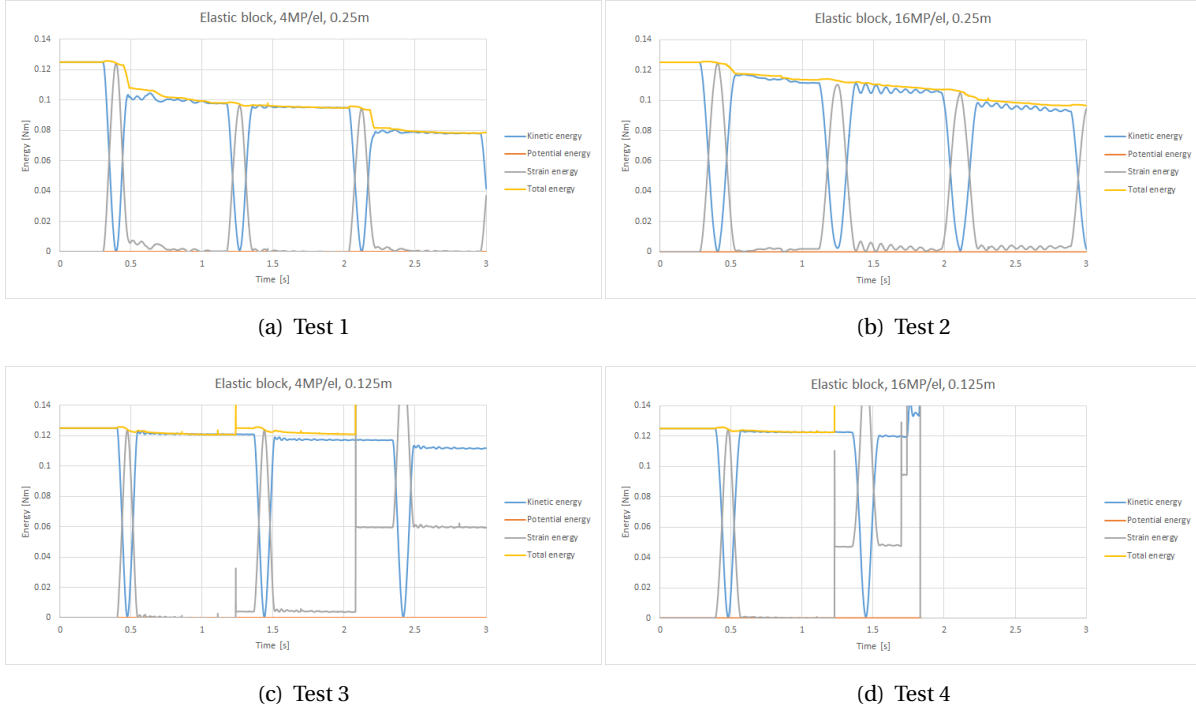


Figure 5.3: Evolution of energy for elastic disk collision under different settings.

At the same time, Tests 3 and 4 in Figure 5.3 show a sudden spike in the calculated strain energy occurring as the body moves away after the first body-to-body collision. Despite that, kinetic energy was reproduced without visible errors indicating that the motion of the material is captured adequately. More material points representing the same quantity of material decreases the individual MP mass m_p . This may cause problems for single or small groups of material points crossing into adjacent elements in that the furthest nodes receive small mass contributions from the MPs. With the USL formulation used here, nodal accelerations are calculated using the inverse of the mass matrix. A node with very low mass could receive an artificially large acceleration and in turn updated velocity. This propagates into the strain and stress calculation for the material points. In order to visualise this issue, the smallest material point volume and the smallest nodal mass were plotted on a second axis, alongside the energy values for the block of interest (Fig. 5.4). Material points crossing over background grid boundaries can be identified by the dips in the lowest nodal mass. On the other hand, material point volume changes visibly when the bodies are in contact, due to the compressive strains in the material. Instances of cell crossing and the related issues become more pronounced with increasing mesh resolution hence some treatment is needed for the cell-crossing problem.

The strain energy spikes seen in the Test 4 setup coincided with these cell crossing events. Issues with material point crossing between cells are inherent to the low order elements used. Clearly, some treatment would be needed to reduce these effects and maintain stability in subsequent analyses. It should be noted, that the initial contact was resolved accurately. In the first instance, bodies do not have internal stresses which are associated with these errors. Developing specific approaches

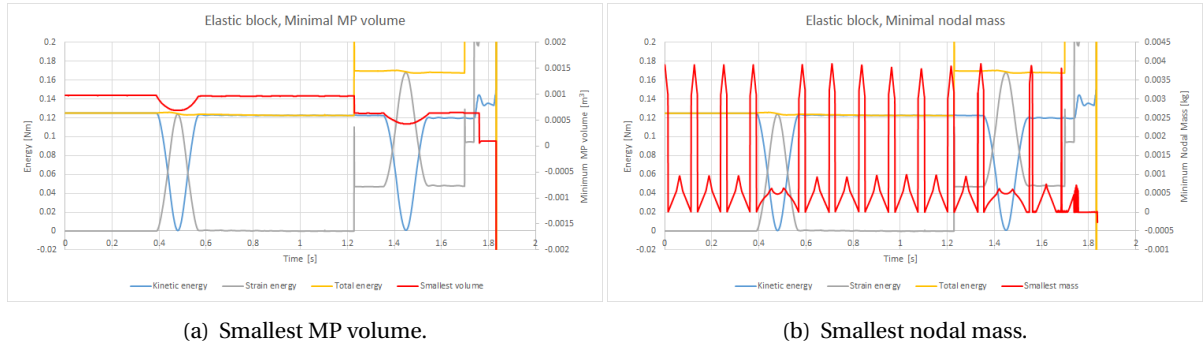


Figure 5.4: Minimal material point volume and nodal mass in Test 4.

to mitigate the erroneous stress behaviour was outside the scope of this thesis, however another solution to increase stability was implemented.

As previously discussed, the Modified-Update-Stress-Last approach proposed by [Sulsky and Schreyer, \[31\]](#) was originally developed to smooth the effects of artificially large accelerations caused by nodes with a small associated mass. This entails an additional interpolation step in which material point velocities are updated directly from nodal accelerations. These are then used to interpolate a smoother velocity field at the nodes. Appendix F shows the summarised variable update procedures for the USL and MUSL methods as employed here. Figure 5.5 shows the effects of the modified algorithm applied to a similar problem. In this instance, larger bodies were used in the analysis and the USL implementation was able to maintain stability. The MUSL implementation showed less energy dissipation during free vibration after contact with the other block. This corroborates the observations made for the vibrating bar. At the same time, a more significant energy loss was observed during the contact.

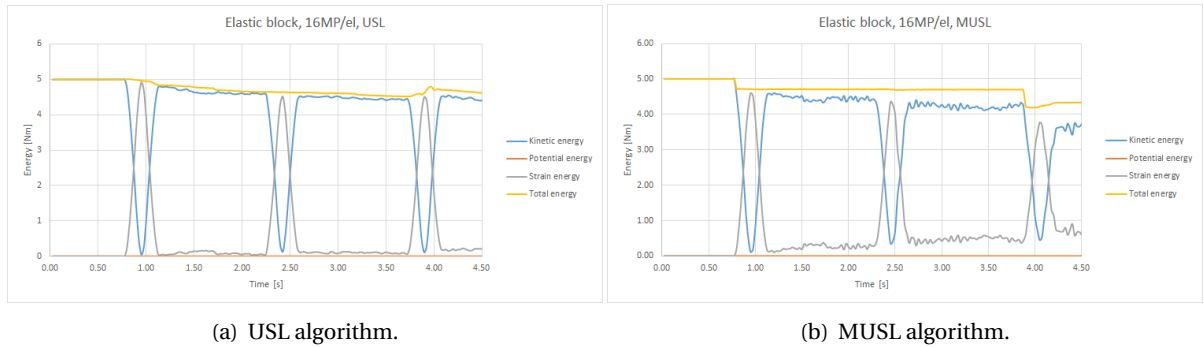


Figure 5.5: Comparison of USL and MUSL for elastic block collision.

For this and subsequent analyses the material properties were altered to reflect a stiffer material. Density was increased proportionally to keep the same elastic wave velocity. The time step size was decrease by a factor of 0.5 and the number of steps increased by a factor of 3.

Body	# El.	# MP/El.	E [Pa]	ν [-]	ρ [kg/m³]	ν_0 [m/s]	Δy	Δx	Δt	#Steps
1	16 (4x4)	16	1000	0.2	10.0	1.0	0.25	0.25	$5e-4$	9000
2	16 (4x4)	16	1000	0.2	10.0	-1.0	0.25	0.25	$5e-4$	9000

Table 5.3: Configuration and material properties for elastic block collision.

REDUCTION OF DISTANCE BETWEEN COLLIDING BODIES

Including a distance-based criterion into the contact logic has been shown as beneficial to the realistic resolution of contact. A penalty factor based on the distance between the contact node and the nearest material point was implemented after [Ma et al., \[1\]](#). This did not influence the outcome of results for the case of *odd* contact, where material points from the different bodies do not share the same element. It can be noted that this approach introduces a extra parameter for the contact resolution stage - the penalty power k .

5.2.2. ELASTOPLASTIC BLOCK COLLISION

The effects of adding perfect plasticity initiated by the Von Mises failure criterion were evaluated by re-analysing the above scenario. As before, the energy of a block was calculated showing a conversion of kinetic energy into strain energy during collision. Plasticity corresponds to a dissipation of energy and is reflected in the simulation as an irrecoverable increase in strain energy. As before, no frictional component was applied to allow for direct comparison with the purely elastic case.

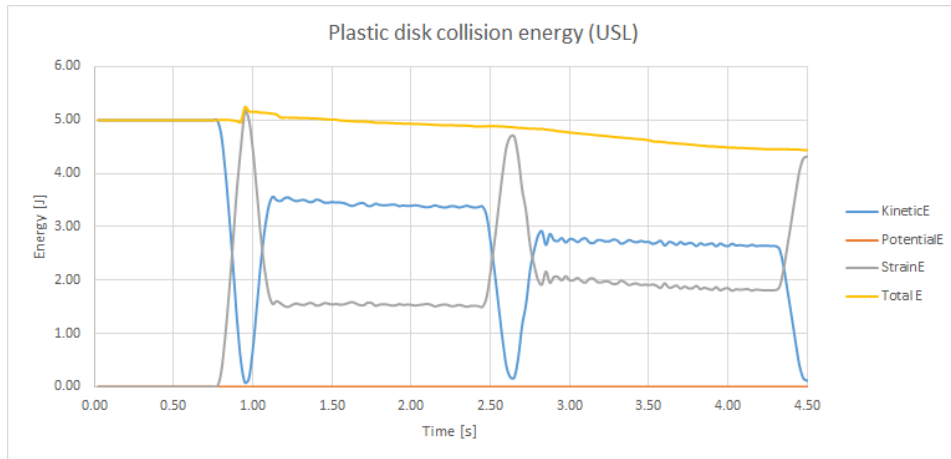


Figure 5.6: Evolution of energy for plastic disk collision.

5.2.3. CVT IMPLEMENTATION

The elastic collision of two disks, discretised using the CVT-based approach was carried out. This served to demonstrate the applicability of the proposed method and compare it qualitatively to the previous analyses. The material properties (Table 5.4) and setup of the test are shown below (Fig. 5.7). Material properties were chosen after [Huang et al., \[23\]](#).

The analysis was initially carried out using defined, colinear surface normals for interaction and applying the penalty method with a penalty power of 6. Examination of the calculated energy throughout the analysis revealed that the penalty method resulted in energy loss. In contrast, the analysis showed spikes in the strain energy similar to those observed in the block collision visible in Figure 5.8.

Body	# El.	# MP/El.	E [Pa]	ν [-]	ρ [kg/m ³]	v_0 [m/s]	Δy	Δx	Δt	#Steps
1	16 (4x4)	18.75	7.71e5	0.286	2e3	1.0	0.25	0.25	5e-4	9000
2	16 (4x4)	18.75	7.71e5	0.286	2e3	-1.0	0.25	0.25	5e-4	9000

Table 5.4: Configuration and material properties for elastic disk collision.

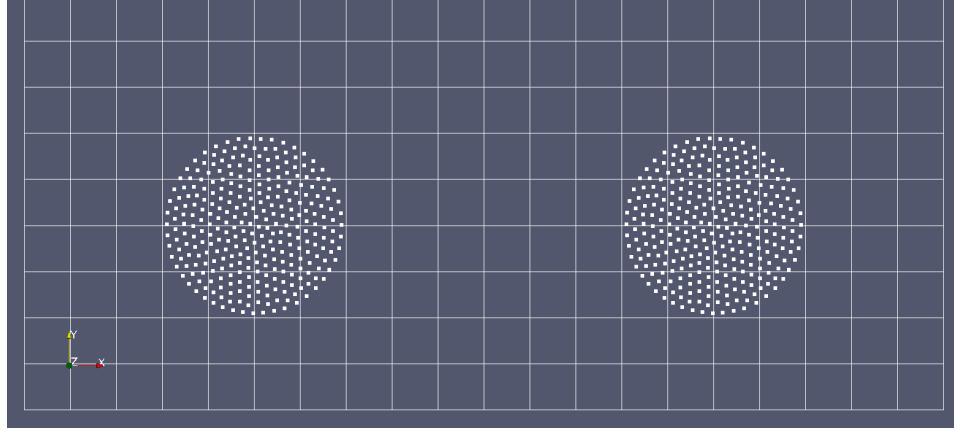


Figure 5.7: Colliding disks problem setup.

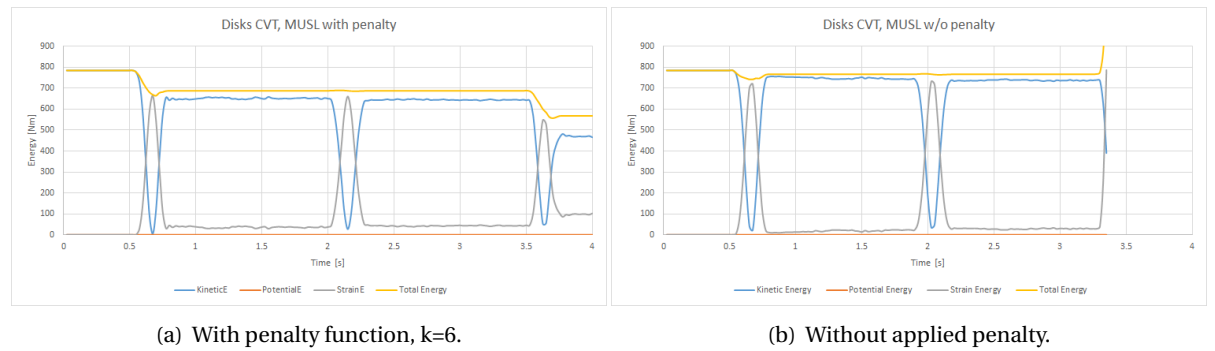


Figure 5.8: Results from the CVT-based implementation in disk collision.

5.2.4. BLOCK SLIDING ON INCLINED PLANE

Including frictional slip in the contact logic allowed further problems to be analysed. A block sliding on an inclined plane was deemed as a setup with a simple analytical solution. It should be noted that this problem would be significantly more difficult to treat using the Finite Element Method, where an involved contact detection procedure would be needed. The details regarding the calculation of energy values are included in a previous section. Multiple researchers have performed the test with a sphere rolling down a plane and documented the motion of the sphere, rarely publishing results regarding the energy of the system. [Huang et al., \[23\]](#) indicate the potential energy of the sliding object is transformed into kinetic and strain energy for the object, but also transferred to the deformable plane through the interaction. The authors have chosen not to show potential energy in plots, however its inclusion allows the expected downward trend in total energy to be observed.

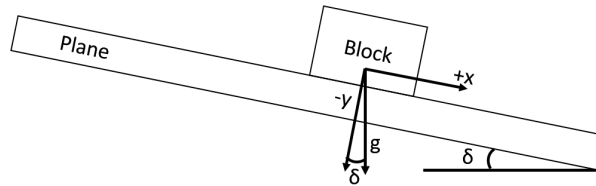


Figure 5.9: Setup used for the sliding block test.

An analytical solution can be obtained from simple kinematics for a rigid block and plane. This was compared to the numerical results. Notably, the mass (densities) of the block and plane do

not appear in the analytical solution. This precludes direct comparison with the numerical results, however these have been included as qualitative guidelines for the expected behaviour.

$$x_{cm} = \begin{cases} x_0, & \text{if } \tan\theta \geq \mu \quad (\text{Stick}) \\ x_0 + \frac{1}{2} |g| t^2 (\sin\theta - \mu \cos\theta), & \text{if } \tan\theta < \mu \quad (\text{Slip}) \end{cases} \quad (5.9)$$

Body	# Elements	# MP/El.	E [kPa]	ν [-]	ρ [kg/m ³]	v_0 [m/s]	Δy	Δx	Δt	# Steps
Block	8 (8x1)	16	7.71e5	0.286	2e3	0	0.25	0.25	5e-4	4000
Plane	70 (35x2)	16	7.71e6	0.286	2e3	0	0.25	0.25	5e-4	4000

Table 5.5: Configuration and material properties for sliding block.

Table 5.5 shows the material properties employed in the analysis. These were selected in order to prevent excessive deformation of the plane and ensure equivalent wave speeds in both materials. Initially, a test without friction was carried out. The only force acting on the block was gravity, directed at an angle so as to mimic the slope inclination. Figure 5.10 shows agreement between the numerical results and the analytical model. The same problem can be analysed for all combinations of plane inclination and friction ratio. Of particular interest is the transition between the sliding and sticking state.

N.B. If the block reaches the domain boundary, there will be a node the block, the plane interact and the domain boundary interact. At this point, special attention must be paid to the boundary conditions, as a fixed boundary freedom will disable the interaction along the DOF. Depending on the arrangement, this may cause the block to penetrate into the plane. If the domain is fixed only in one direction, this may influence the correction of the nodal velocity needed for the successful resolution of contact. In such scenarios it would be appropriate to have the boundary fully fixed.

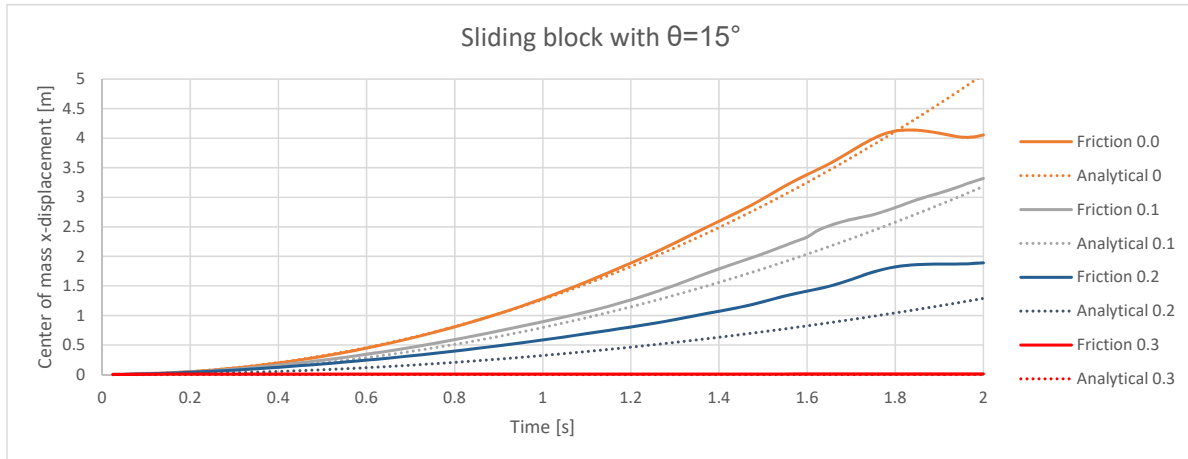


Figure 5.10: Block sliding on inclined plane for different μ values.

For higher coefficients of friction, the block should remain static. It should be noted that oscillations resulting from the perfectly elastic material and plane influenced the results of the analysis. Several possibilities for mitigating these effects exist. As a first step, bodies should be initiated with internal stresses arising from an equilibrium with external loading. Additionally, wave reflection from the fixed domain boundaries should be limited by introducing local non-viscous damping or

a silent boundary. The former is straightforward to apply, however it can influence the frictional interaction between the block and the plane by artificially altering material point velocities along the contact interface. To this end, a silent boundary implementation was deemed superior. It should be noted, that none of the consulted publications discussed boundary treatments in their presentation of similar analyses.

Effectively, oscillations in the elastic block allowed it to slide along a small amount before the full frictional interaction stopped its motion. This is an artefact of the purely elastic materials used in the scenario. Applying a small amount of non-viscous damping to the plane material was used to remedy these effects with the argument that the motion of the block is of interest.

5.2.5. INTERACTION BETWEEN MPM AND FEM DOMAINS

The multi-mesh contact algorithm was applied to achieve interaction between an FEM-discretised domain and an MPM-discretised domain (after [41]). This approach would be beneficial in situations, requiring interactions between materials of different stiffness. The alternative Point-to-Segment approach, based on ICFEMP [6] was not completed in time to produce useful results. The required code structure and procedures have been collected and the implementation can be completed as a further research interest of the section.

BLOCK RESTING ON BEAM

As a way to demonstrate the possibilities offered by the FEM-MPM coupling, the interaction between a cantilever beam and a block resting on it was analysed. The beam, discretised using Q8 finite elements with reduced integration was fully fixed at the left boundary and modelled with a linear-elastic perfectly-plastic material. Gravity was applied only to the block, which was discretised using material points. Both bodies were initiated in contact with one another. When the beam deflected sufficiently the MPM block was able to slip off and fall, provided slip is allowed between the two objects. The material properties of the scheme is outlined in Table 5.6 and the initial setup is shown in Figure 5.11.

Body	# El.	# MP/El.	E [Pa]	ν [-]	ρ [kg/m^3]	Y [Pa]	Δy	Δx	Δt	# Steps
Block	12 (4x3)	16	$1.5e4$	0.0	$1e2$	-	0.25	0.25	$5e-4$	9000
Beam	30 (15x2)	-	$1.5e7$	0.0	$6e3$	$1.5e7$	0.25	0.25	$5e-4$	9000

Table 5.6: Configuration and material properties for block-on-beam problem.

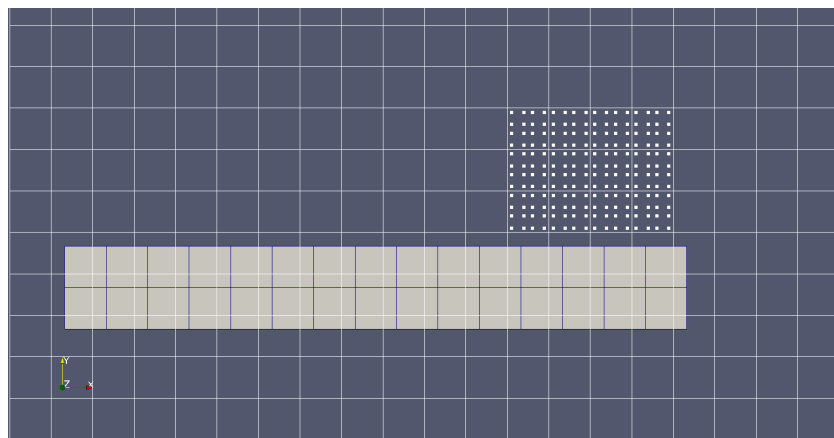


Figure 5.11: Problem setup for the block-on-beam problem.

In this case the multi-velocity field contact algorithm is employed, meaning contributions from material points and FEM nodes are mapped onto the background grid to detect contact. This requires consistency between the element sizes. The example illustrates that interaction between different element types can be achieved. Problems were encountered with this setup when using full integration of the Finite Elements, hence reduced integration of the finite elements was used. The material of the FE beam was modelled as elastoplastic, however the beam did not yield in this instance.

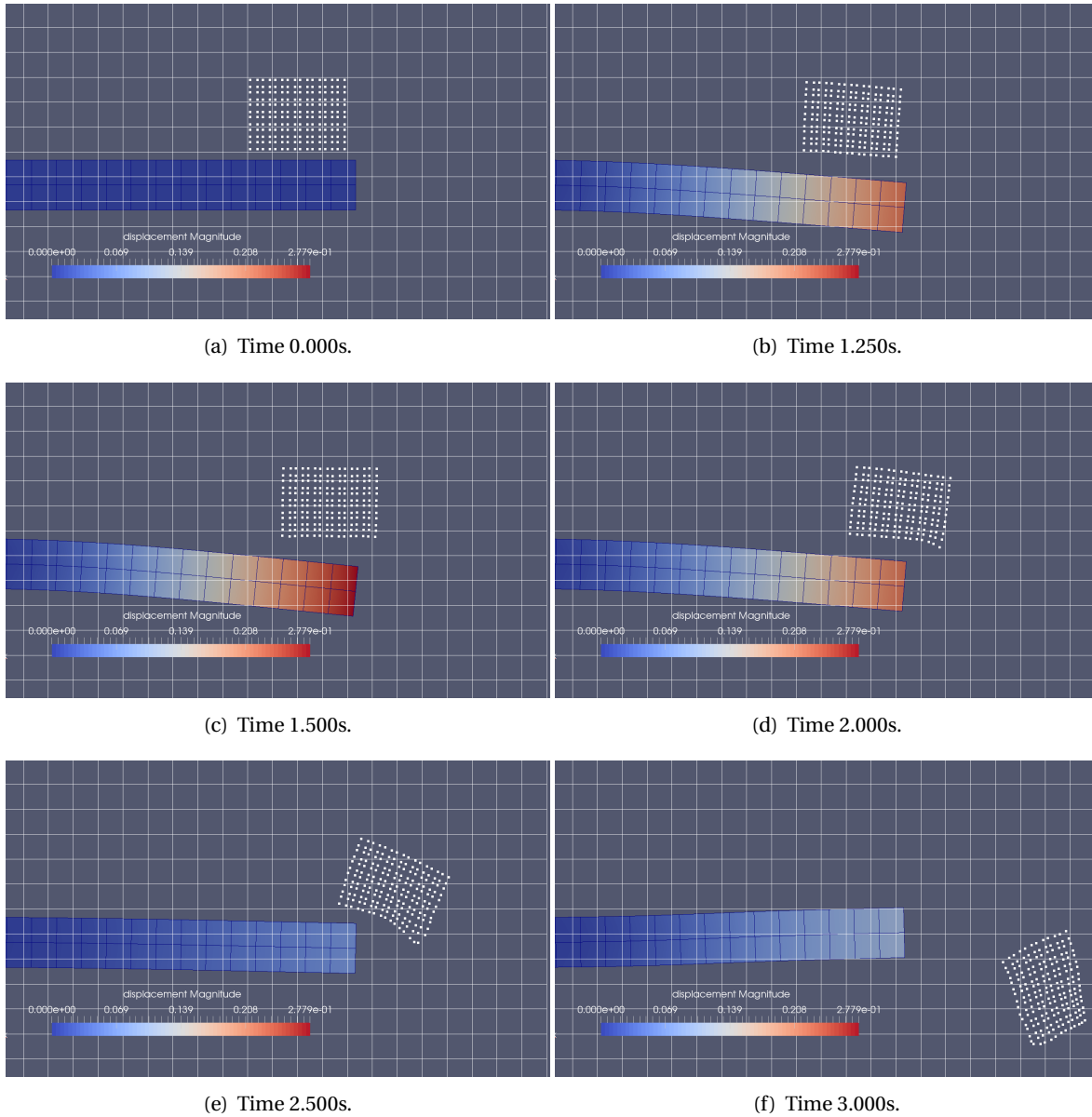


Figure 5.12: FEM-MPM interaction for the block-on-beam problem.

6

CONCLUSIONS AND RECOMMENDATIONS

Following the presented results and their discussion in the previous chapters, conclusions regarding the effectiveness and applicability of the contact schemes implemented are presented here. Recommendations for future work on the topic of contact modelling are given based on the conclusions and the research program of the Geo-Engineering section of TU Delft.

6.1. GENERAL CONCLUSIONS

The application of the Material Point Method for modelling contact problems was analysed by producing a dynamic explicit MPM code. Possibilities for interaction between domains of different discretisation were explored. During the literature survey carried out, a large number of publications were consulted. The general conclusion made from the literature observed was that there is little consensus on the most appropriate methods for use in the Material Point Method. This was found particularly true when considering literature regarding the stress discontinuities observed in low order elements. On the topic of contact, the difference in modelling approaches was small as most researchers adopted variations on an approach elaborated by [Bardenhagen *et al.*, \[35\]](#). This meant there was opportunity to explore the topic, however the other issues observed in MPM made it difficult to produce reliable results.

6.2. CONCLUSIONS REGARDING THE MULTI-MESH ALGORITHM

The multi-velocity field based algorithm was implemented and tested for several example problems. Results confirmed the ability of the approach to capture frictional contact between deformable continua. The ability to separate bodies into different velocity fields allows other phenomena such as fracture to also be modelled.

Several shortcomings in the implementation were identified. The inaccurate determination of collinear normals for the interaction had a negative effect on the quality of the solutions. Additionally, the poor stress response arising from the low-order elements used made the implementation of more advanced contact logic conditions, such as imposing a requirement for compressive stress at contact nodes. Despite that, no issues with free separation of the contact pair were encountered due to this.

As expected, increasing the resolution of the background grid was beneficial for the accuracy of contact detection and calculation. At the same time, additional considerations were needed to maintain the stability of the program. Using the MUSL approach proved successful in decreasing

the negative effects of small mass nodes, however it was noted that this impacted the energy calculated during contact.

6.2.1. DISTANCE BETWEEN CONTACTING BODIES

Two schemes to reduce distance between contacting bodies were investigated. Of these, the penalty power approach proved more successful in reducing the distance and alleviating issues associated with free separation. These were traced back to errors in the definition of the normal and tangential directions, in particular at geometrical discontinuities. Despite introducing another parameter into the contact formulation which also needs to be calibrated, it was shown that selecting an appropriate value for the penalty power proved beneficial to the stability of the model. A penalty power of 6 was found to be appropriate for most analyses.

The problematic scenario, referred to here as *odd contact* may be avoided for certain test setups, particularly in symmetric problems. This is however not the general case and appropriate treatment is needed to ensure stability during more involved simulations such as e.g. pile penetration or landslide flow interaction with obstacles. A scheme to guarantee collinearity of the contact normals [23] or a penalty function to reduce the thickness of the contact interface [1] are both viable options. The distance cutoff criterion, proposed by Ma *et al.*, [56], did not remedy these issues and caused erratic strain energy to develop.

6.2.2. NORMAL CALCULATION

As expected, the accurate determination of the surface normal was shown to influence the momentum conservation properties of the contact algorithms. Most research surveyed has opted for the simplest possible approach and increased the resolution of the background grid. This works to reduce the effect of errors when the contact normal is ill-defined. At lower resolution however, these can have a dramatic impact on the quality of the solution, going so far as to completely alter the end result. This was observed in the impact of rectangular block, without additional treatment to the surface normal at the *corner* nodes. Adding friction between the colliding blocks resulted in an erroneous rotation.

The solution to this could include an alternative approach to track the body surface. Level-sets are an interesting option to reconstructing and tracking the body contour once discretised with material points [57]. These can provide a continuous approximation to the discrete body surface and thus allow for a robust definition of the normal direction as needed. Alternatively, a continuous reconstruction of the body surface can be achieved by fitting splines in the region of contact grid nodes.

6.2.3. STABILITY

The model encountered stability issues stemming from erratic stress-strain behaviour in the Update-Stress-Last and Update-Stress-First implementations ¹. This behaviour became more pronounced with increasing number of material points and decreasing the background cell size. To address the issue, the Modified-Update-Stress-Last scheme was implemented. Results demonstrated improved stability and energy conservation properties of the model. In the benchmark tests performed, the MUSL approach demonstrated significantly less energy dissipation for freely vibrating bodies moving through a domain. A careful revision of the *double mapping* procedure may help reduce the additional computational cost incurred by the additional mapping.

The timestep sizes required to maintain stability in the explicit integration scheme remained

¹The latter was only used in the vibrating bar problem and abandoned from further investigation.

small throughout the work, which can hamper the further application of the model to larger analyses. Introducing non-viscous damping as proposed in [58] could improve stability and allow for larger time steps to be used. It is important to note this may introduce non-physical effects into the solution. The CFL condition was used to estimate the critical time step for stability, however the resulting increment was generally too large to achieve satisfactory results. The time step was kept below 10% of the CFL-predicted value throughout the work.

6.3. CONCLUSIONS REGARDING FEM-MPM INTERACTION

The interaction between FEM and MPM-discretised domains was investigated using a local multi-mesh scheme. This approach was able to produce satisfactory results where cell sizes of finite elements and the background grid were similar. Both 4-node and 8-node elements were investigated, however the latter exhibited significant instability when using full integration to the point where analysis was no longer feasible. Reduced integration used for the quadratic quadrilateral elements resolved these issues. Integrating Q8 elements with 4 integration points is prone to producing hour-glass modes, however these are typically not communicable and do not occur in groups of elements.

With respect to the issues highlighted above, a preference for using Q4 elements seems natural despite their "stiffer" behaviour. This can be remedied by adopting "improved" Q4 elements (sometimes called Q6), which add additional internal degrees of freedom to permit accurate capture of bending. These have been found to alleviate the overly stiff response of bilinear quadrilaterals, despite introducing non-physical deformation modes. Despite this, they are widely adopted by FEA software (e.g. ABAQUS).

The interaction between the material points and finite element nodes was achieved by mapping nodal contributions onto the background grid. Despite the different number of elements used for the FEM solver, the analysis was able to produce results.

6.4. CONCLUSIONS REGARDING MATERIAL POINT GENERATION

The local subdivision algorithm described by [44] was predominantly used in the code, generating a finite element mesh and placing material points at Gauss points. The advantages of this approach were seen as its relative ease of incorporating into the workflow for simple geometries and the good quality of mapping at the start of the analysis. A potential drawback of the approach discussed was the need to initially produce a mesh of the material domain, which would then be populated by material points. In combination with the structured mesh approach used in the current codebase, this made the analysis of any non-rectangular geometries significantly more complex. A tailored solution for generating the mesh would be required for each problem, unless a different approach is developed. Unstructured meshes may be desirable for analyses of more complex domain shapes, however additional difficulties related to locating the material points are then encountered. Elements would also have different sizes, resulting in material points of different mass. The implications of these for the analysis remain to be investigated.

An alternative method to generate material points using the concept of Centroidal Voronoi Tessellations was presented. The iterative scheme utilised, required more time to complete the material point generation than the comparison Subdivision algorithm. At the same time, the CVT-based approach was seen as a more flexible method to discretise the geometry. The material points could be generated and evenly distributed across a domain of arbitrary shape, resulting in a density field comparable to the one obtained using local subdivision. Applying this to an elastic collision problem revealed satisfactory results both in terms of resolving the kinematics of the problem and energy conservation compared to instances using the local subdivision method. The code collected and re-

vised as part of this thesis can be used for distributing material points across a domain of general shape, irrespective of the background grid used. A drawback was noted in the fact that the number of material points per element could not be controlled precisely, however this was found not to affect the solution quality for a sufficiently high average.

6.5. RECOMMENDATIONS

The Material Point Method holds great potential for adoption in the field of Geotechnical Engineering. Research efforts by a dedicated community in recent years have led to a number of new developments and applications for the method. As yet, there is no well-established commercial product utilising MPM and it remains predominantly a research tool.

6.5.1. FURTHER WORK

The implemented contact models permit the analysis of further classes of problems in addition to the capabilities of the MPM code present at the Geo-Engineering section. These have been investigated without significant changes in other areas of the implementation.

During the course of testing, issues related to cell-crossing noise and achieving a satisfactory stress response were encountered. These manifested as non-physical jumps in the stress field and sharp spikes in the model strain energy. Authors have argued that using a mass-weighted averaging from material points onto grid nodes can be used to produce a smooth stress field for presenting stresses [27], however such an approach has not been successfully used in calculations. As a next step in the development of the MPM code, further improvements to the stress calculation should be incorporated alongside the contact detection. These can include modifications to the shape function derivatives used in the stress calculation, using different element types or others. Suitable improvements in this area will benefit the stability and accuracy of contact simulations.

6.5.2. CODE DEVELOPMENT

The following discussion and recommendations are made based on observations of the code development process at the TU Delft Geo-Engineering section and in the context of a research program to develop MPM codes for applications in geotechnical engineering.

Use of the Fortran programming language allows the production of high-performance code and its modern revisions enable useful features such as defined types, inheritance, parallelisation etc. At the same time, it suffers from some limitations regarding the development environment support. Switching to a more modern object-oriented language (such as C++) may be beneficial in terms of the large number of open-source libraries for specific finite element applications, computational geometry, linear algebra, visualisation etc. These resources, coupled with the number of compilers available for C++ could improve the effectiveness of code development within the research group. Some examples of currently developed MPM software are Uintah (University of Utah) and MPM3Dpp [56]. Both of these codes are developed predominantly in the C++ programming language. Ma *et al.*, [56] argue that as MPM codes grow in complexity, aspects such as code reusability, extensibility and maintainability become increasingly important.

Scalability is another important aspect for future growth and development of the MPM code-base within the research group. In particular - the increases in computational and memory cost as the scale of the analyses increase. In this regard, FORTRAN is a language well suited to optimisation both from a user and compiler standpoint. Appropriate use of vectorisation, where possible, can greatly reduce the running time and memory required to conduct large analyses.

A trade-off exists between flexibility of the code (being easily adapted for different sizes of input, element types, material point density etc.) and its runtime performance. For example, avoiding dynamic memory allocation within the time stepping loops would reduce the overhead for determining variable sizes and allocating memory. This however, would involve initially allocating a maximum possible size for variables of unknown size, which could cause the code to use too much memory or encounter errors where the initially allocated size is found insufficient.

6.5.3. MESHING

An attempt to explore the material point generation procedure was made in this thesis, using a CVT-based approach. At present, the mesh generation features of the available code are limited and may hinder further research. This poses a challenge to faithfully representing the desired modelling scenario and ultimately hinders research progress. A survey of available Finite Element Meshing tools or libraries and their appropriate use could be very beneficial for future developments of the code.

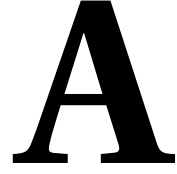
BIBLIOGRAPHY

- [1] J. Ma, D. Wang, and M. Randolph, *A new contact algorithm in the material point method for geotechnical simulations*, [International Journal for Numerical and Analytical Methods in Geomechanics](#) **38**, 1197 (2014), [arXiv:nag.2266 \[10.1002\]](#) .
- [2] M. A. Homel and E. B. Herbold, *Field-gradient partitioning for fracture and frictional contact in the material point method*, [International Journal for Numerical Methods in Engineering](#) , 1 (2016).
- [3] Y. Lian, X. Zhang, F. Zhang, and X. Cui, *Tied interface grid material point method for problems with localized extreme deformation*, [International Journal of Impact Engineering](#) **70**, 50 (2014).
- [4] F. Hamad, D. Stolle, and P. Vermeer, *Modelling of membranes in the material point method with applications*, [International Journal for Numerical and Analytical Methods in Geomechanics](#) **39**, 833 (2015).
- [5] W. Hu and Z. Chen, *A multi-mesh MPM for simulating the meshing process of spur gears*, [Computers & Structures](#) **81**, 1991 (2003).
- [6] Z. Chen, X. Qiu, X. Zhang, and Y. Lian, *Improved coupling of finite element method with material point method based on a particle-to-surface contact algorithm*, [Computer Methods in Applied Mechanics and Engineering](#) **293**, 1 (2015).
- [7] I. K. J. Al-Kafaji, *University of Stuttgart*, Ph.D. thesis, University of Stuttgart (2013).
- [8] I. Smith, D. Griffiths, and L. Margetts, *Programming the Finite Element Method*, 4th ed. (John Wiley & Sons, Ltd., Chichester, West Sussex, 2013) p. 628.
- [9] D. Sulsky, Z. Chen, and H. Schreyer, *A particle method for history-dependent materials*, [Computer Methods in Applied Mechanics and Engineering](#) **118**, 179 (1994).
- [10] N. Phuong, A. van Tol, A. Elkadi, and A. Rohe, *Numerical investigation of pile installation effects in sand using material point method*, [Computers and Geotechnics](#) **73**, 58 (2016).
- [11] A. Yerro, E. Alonso, and N. Pinyol, *The material point method for unsaturated soils*, [Géotechnique](#) **65**, 201 (2015).
- [12] B. Wang, P. J. Vardon, M. A. Hicks, and Z. Chen, *Development of an implicit material point method for geotechnical applications*, [Computers and Geotechnics](#) **71**, 159 (2016).
- [13] B. Wang, P. Vardon, and M. Hicks, *Investigation of retrogressive and progressive slope failure mechanisms using the material point method*, [Computers and Geotechnics](#) **78**, 88 (2016).
- [14] C. M. Mast, *University of Washington*, Ph.D. thesis (2013).
- [15] C. J. Coetzee, P. A. Vermeer, and A. H. Basson, *The modelling of anchors using the material point method*, [International Journal for Numerical and Analytical Methods in Geomechanics](#) **29**, 879 (2005).

- [16] N. Prime, F. Dufour, and F. Darve, *Solid-fluid transition modelling in geomaterials and application to a mudflow interacting with an obstacle*, [International Journal for Numerical and Analytical Methods in Geomechanics](#) **38**, 1341 (2014), arXiv:nag.2260 [10.1002] .
- [17] K. Soga, E. Alonso, A. Yerro, K. Kumar, and S. Bandara, *Trends in large-deformation analysis of landslide mass movements with particular emphasis on the material point method*, [Géotechnique](#) , 1 (2015).
- [18] T. Belytschko, W. Liu, and B. Moran, *Nonlinear Finite Elements for Continua and Structures* (John Wiley & Sons, Ltd., Chichester, West Sussex, 2000) p. 650.
- [19] B. Szabó and I. Babuška, *Introduction to Finite Element Analysis: Formulation, Verification and Validation* (John Wiley & Sons, Ltd., Chichester, West Sussex, 2011) p. 372.
- [20] O. Zienkiewicz and R. Taylor, *The finite element patch test revisited a computer test for convergence, validation and error estimates*, [Computer Methods in Applied Mechanics and Engineering](#) **149**, 223 (1997).
- [21] V. a. Yastrebov, *Numerical Methods in Contact Mechanics*, edited by V. A. Yastrebov and P. Breitkopf (John Wiley & Sons, Inc., Hoboken, NJ USA, 2013) p. 391.
- [22] H. K. Engin, R. B. J. Brinkgreve, and A. F. van Tol, *Simplified numerical modelling of pile penetration - the press-replace technique*, [International Journal for Numerical and Analytical Methods in Geomechanics](#) **39**, 1713 (2015).
- [23] P. Huang, X. Zhang, S. Ma, and X. Huang, *Contact algorithms for the material point method in impact and penetration simulation*, [International Journal for Numerical Methods in Engineering](#) **85**, 498 (2011), arXiv:1010.1724 .
- [24] Z. Chen and R. Brannon, *Sand2002-0482*, Tech. Rep. (2002).
- [25] E. Love and D. L. Sulsky, *An energy-consistent material-point method for dynamic finite deformation plasticity*, [International Journal for Numerical Methods in Engineering](#) **65**, 1608 (2006).
- [26] C. M. Mast, P. Mackenzie-Helnwein, P. Arduino, G. R. Miller, and W. Shin, *Mitigating kinematic locking in the material point method*, [Journal of Computational Physics](#) **231**, 5351 (2012).
- [27] S. Andersen and L. Andersen, *Analysis of spatial interpolation in the material-point method*, [Computers & Structures](#) **88**, 506 (2010).
- [28] S. G. Bardenhagen and E. M. Kober, *The generalized interpolation material point method*, [CMES - Computer Modeling in Engineering and Sciences](#) **5**, 477 (2004).
- [29] A. Sadeghirad, R. M. Brannon, and J. Burghardt, *A convected particle domain interpolation technique to extend applicability of the material point method for problems involving massive deformations*, [International Journal for Numerical Methods in Engineering](#) **86**, 1435 (2011).
- [30] D. Sulsky, S.-J. Zhou, and H. L. Schreyer, *Application of a particle-in-cell method to solid mechanics*, [Computer Physics Communications](#) **87**, 236 (1995).
- [31] D. Sulsky and H. L. Schreyer, *Axisymmetric form of the material point method with applications to upsetting and Taylor impact problems*, [Computer Methods in Applied Mechanics and Engineering](#) **139**, 409 (1996).
- [32] D. Burgess, D. Sulsky, and J. U. Brackbill, *Mass matrix formulation of the FLIP particle-in-cell method*, [Journal of Computational Physics](#) **103**, 1 (1992).

- [33] X. Ma, P. T. Giguere, B. Jayaraman, and D. Z. Zhang, *Distribution coefficient algorithm for small mass nodes in material point method*, [Journal of Computational Physics](#) **229**, 7819 (2010).
- [34] E. Love and D. Sulsky, *An unconditionally stable, energy–momentum consistent implementation of the material-point method*, [Computer Methods in Applied Mechanics and Engineering](#) **195**, 3903 (2006).
- [35] S. Bardenhagen, J. Brackbill, and D. Sulsky, *The material-point method for granular materials*, [Computer Methods in Applied Mechanics and Engineering](#) **187**, 529 (2000).
- [36] P. Wallstedt and J. Guilkey, *An evaluation of explicit time integration schemes for use with the generalized interpolation material point method*, [Journal of Computational Physics](#) **227**, 9628 (2008).
- [37] S. Bardenhagen, *Energy Conservation Error in the Material Point Method for Solid Mechanics*, [Journal of Computational Physics](#) **180**, 383 (2002).
- [38] J. E. Guilkey and J. a. Weiss, *Implicit time integration for the material point method: Quantitative and algorithmic comparisons with the finite element method*, [International Journal for Numerical Methods in Engineering](#) **57**, 1323 (2003).
- [39] D. Z. Zhang, X. Ma, and P. T. Giguere, *Material point method enhanced by modified gradient of shape function*, [Journal of Computational Physics](#) **230**, 6379 (2011).
- [40] A. R. York, D. Sulsky, and H. L. Schreyer, *The material point method for simulation of thin membranes*, [International Journal for Numerical Methods in Engineering](#) **44**, 1429 (1999).
- [41] Y. Lian, X. Zhang, and Y. Liu, *Coupling of finite element method with material point method by local multi-mesh contact method*, [Computer Methods in Applied Mechanics and Engineering](#) **200**, 3482 (2011).
- [42] Y. Lian, X. Zhang, and Y. Liu, *An adaptive finite element material point method and its application in extreme deformation problems*, [Computer Methods in Applied Mechanics and Engineering](#) **241-244**, 275 (2012).
- [43] A. Sadeghirad, R. Brannon, and J. Guilkey, *Second-order convected particle domain interpolation (CPDI2) with enrichment for weak discontinuities at material interfaces*, [International Journal for Numerical Methods in Engineering](#) **95**, 928 (2013), [arXiv:1010.1724](#).
- [44] L. J. Lim, *Pile Penetration Simulation using Material Point Method*, Tech. Rep. July (Delft University of Technology, 2012).
- [45] J. Bardenhagen, S.G.; Guilkey, J.E.; Roessig, K.M.; Brackbill, J.U.; Witzel, W.M.; Foster, *An Improved Contact Algorithm for the Material Point Method and Application to Stress Propagation in Granular Material*, [Computer Modeling in Engineering and Sciences](#) **2**, 509 (2001).
- [46] X.-F. Pan, A.-G. Xu, G.-C. Zhang, P. Zhang, J.-S. Zhu, S. Ma, and X. Zhang, *Three-Dimensional Multi-mesh Material Point Method for Solving Collision Problems*, [Communications in Theoretical Physics](#) **49**, 1129 (2008), [arXiv:0708.3532](#).
- [47] N. El-Abbasi and K. J. Bathe, *Stability and patch test performance of contact discretizations and a new solution algorithm*, [Computers and Structures](#) **79**, 1473 (2001).
- [48] V. P. Nguyen, T. Rabczuk, S. Bordas, and M. Duflot, *Meshless methods: A review and computer implementation aspects*, [Mathematics and Computers in Simulation](#) **79**, 763 (2008).

- [49] C. de Boor, *A practical guide to splines (Revised Edition)* (Springer-Verlag New York, Inc., 2001) p. 346.
- [50] J. Barlow, *Numerical Aspects of Solving Linear Least Squares Problems*, in *Handbook of Statistics*, Vol. 9, edited by C. Rao (Elsevier B.V., 1993) Chap. 9, pp. 303–376.
- [51] L. M. Shen and Z. Chen, *A silent boundary scheme with the material point method for dynamic analyses*, *Computer Modeling in Engineering & Sciences* **7**, 305 (2005).
- [52] Q. Du, V. Faber, and M. Gunzburger, *Centroidal Voronoi Tessellations: Applications and Algorithms*, *SIAM Review* **41**, 637 (1999).
- [53] Q. Du and M. Gunzburger, *Grid generation and optimization based on centroidal Voronoi tessellations*, *Applied Mathematics and Computation* **133**, 591 (2002).
- [54] J. C. Hateley, H. Wei, and L. Chen, *Fast Methods for Computing Centroidal Voronoi Tessellations*, *Journal of Scientific Computing* **63**, 185 (2015).
- [55] P. C. Wallstedt and J. E. Guilkey, *Improved velocity projection for the material point method*, *CMES - Computer Modeling in Engineering and Sciences* **19**, 223 (2007).
- [56] Z. T. Ma, X. Zhang, and P. Huang, *An object-oriented MPM framework for simulation of large deformation and contact of numerous grains*, *CMES - Computer Modeling in Engineering and Sciences* **55**, 61 (2010).
- [57] A. Andreykiv, F. Keulen, D. J. Rixen, and E. Valstar, *A level-set-based large sliding contact algorithm for easy analysis of implant positioning*, *International Journal for Numerical Methods in Engineering* **89**, 1317 (2012).
- [58] L. Wang, F. Hétroy-Wheeler, and E. Boyer, *A Hierarchical Approach for Regular Centroidal Voronoi Tessellations*, *Computer Graphics Forum* **35**, 152 (2016).
- [59] M. Oldenburg and L. Nilsson, *Position code algorithm for contact searching*, *International Journal for Numerical Methods in Engineering* **37**, 359 (1994).
- [60] J. C. Adams, W. S. Brainerd, R. A. Hendrickson, R. E. Maine, J. T. Martin, and B. T. Smith, *The Fortran 2003 Handbook* (Springer London, London, 2009).
- [61] S. Andersen and L. Andersen, *Analysis of stress updates in the material-point method*, in *Proceedings of the Twenty Second Nordic Seminar on Computational Mechanics* (2009) pp. 129–134.



CONTACT CONDITIONS

In the field of continuum mechanics the frictionless contact problem is formulated using the Hertz-Signorini-Moreau conditions. Discretised versions of these conditions are presented below in the context of their use after [21, 23, 59]. A good discussion on how these conditions are affected by the chosen discretisation and their implementation is presented by Yastrebov, [21] (Chapter 2).

A.1. IMPENETRABILITY CONDITION

Once contact occurs between two bodies, the outward unit normal vectors of the two bodies at the contact surface must remain in opposite directions.

$$\sum_{b=1}^2 n_i^b = 0 \quad (\text{A.1})$$

Similarly, the nodal velocities must satisfy the impenetrability condition on the background mesh:

$$\sum_{b=1}^2 n_i^b \cdot v_i^b = 0 \quad (\text{A.2})$$

N.B. Equations A.1 and A.2 above implicitly require the normal vectors to be collinear. If this condition is not satisfied, momentum conservation at the contact surface may not be guaranteed.

A.1.1. CONTACT FORCE CONDITION

The contact force is applied to the body when contact occurs. Based directly on Newton's Third Law, the contact forces applied to the two bodies follow:

$$\sum_{b=1}^2 f_{i,ct}^b = 0 \quad (\text{A.3})$$

The normal force on each body can then be determined as:

$$f_{i,ct}^{b,nor} = f_{i,ct}^b \cdot n_i^b \quad (\text{A.4})$$

N.B. The contact force between the bodies cannot be tensile, hence the restriction is applied: $f_i^{nor} \leq 0$.

A.1.2. MOMENTUM EQUATIONS

The weak form of the contact problem is obtained from the principle of virtual work by appending the Lagrange multiplier term. After imposing the contact constraints, the momentum equation of grid nodes at the contact surface can be written as:

$$m_i^b \cdot a_i^b = f_{i,ext}^b - f_{i,int}^b + f_{i,ct}^b \quad (\text{A.5})$$

A.2. DETECTION OF CONTACT NODES

For each body, the nodal velocity connecting to the body is found as the ratio of momentum to mass:

$$v_i^b = \frac{p_i^b}{m_i^b} \quad (\text{A.6})$$

N.B. In cases where the mass approaches zero, the resulting nodal velocity may become singular. In such cases, use of a formulation based directly on momentum is advised.

Contact occurs where the velocities of the two bodies are projected onto the same node. In that instance, use of the average mapping velocity of all material points on the grid node is termed the centre of mass velocity:

$$v_i^{cm} = \frac{\sum_b p_i^b}{\sum_b m_i^b} \quad (\text{A.7})$$

N.B. Momentum, mass and velocity from material points are mapped onto the nodes of current element of the MPs. This means contact is detected as soon as two adjacent elements receive material points from different bodies. The contact interface then has a thickness h_{con} ($0 \leq h_{con} \leq 2\sqrt{\Delta x^2 + \Delta y^2}$). This thickness also depends on the direction of velocities of the bodies with respect to the calculation grid. This may cause additional errors when bodies contact each other along different surfaces.

A.3. SURFACE NORMAL VECTORS

As discussed in previous sections, calculating the normal direction at the body surface has significant implications for the successful resolution of contact. To date, there is no standard way of calculating the normal at the body surface. In the first implementation of the "multi-velocity field-based algorithm" [35], an indicative procedure for evaluating the surface normal was proposed, despite a detailed example not being provided.

$$n_i^b = -\nabla \rho_e^b \Big|_i = -\nabla \left[\sum_{p=1}^{N_i^b} \frac{m_p^b S^{(2)}(x_c - x_p)}{\Omega_c} \right] \Big|_i \quad (\text{A.8})$$

Equation A.8 calculates the normal using a gradient of densities interpolated to the centres of elements in the support of the contact node, referred to as the "Density gradient" method in this text. Since then, several researchers have implemented variations on the approach. Computing the unit normal vector can be achieved using the gradient of nodal mass within the individual body [23].

$$\hat{n}_{bi} = \frac{1}{|\sum_p G_{ip} m_{bp}|} \sum_p G_{ip} m_{bp} \quad (\text{A.9})$$

The spatial derivatives of the shape functions are utilised in Equation A.9 and referred to as "Mass gradient" here. Alternatively, Pan *et al.*, [46] proposed an approach for calculating the normal using the coordinates of active element nodes within the nodal support (j) and the contact node (i). For bodies 1 and 2 then:

$$n_i = V_i^1 - V_i^2 = \sum_j (x_i^1 - x_{j,c}^1) - \sum_j (x_i^2 - x_{j,c}^2) \quad (\text{A.10})$$

Effectively, this represents a very simplified approach and also guarantees collinearity of normal vectors for the interaction. It does not account for the location of material points within the elements of the nodal support. Details are not provided explaining whether original or updated nodal positions are used.

The "Density gradient" and "Mass gradient" approaches shown *do not* guarantee the collinearity of contacting body normals hence may introduce errors in momentum conservation. The following solutions have been proposed for defining *modified* unit surface normal vectors [21, 23].

- If body 1 is stiffer than body 2, then let $n_{1i} = \hat{n}_{1i} = -n_{2i}$.
- If body 1 with flat/convex surface is expected to come into contact with body 2 with concave surface, let $n_{1i} = \hat{n}_{1i} = -n_{2i}$.
- The modified surface normal vector can be obtained as the average of the original vectors:

$$n_{1i} = -n_{2i} = \frac{1}{|\hat{n}_{1i} - \hat{n}_{2i}|} (\hat{n}_{1i} - \hat{n}_{2i})$$

B

ENERGY CONSERVATION AND 1-D VIBRATION

In the Material Point Method, mass is conserved by definition. A material domain is divided into a set of discrete points, the masses of which remain constant throughout the analysis. Momentum conservation is enforced by discretising and solving the equations of momentum in time and space over the calculation domain. Energy conservation is however, not explicitly enforced and has been found to depend on the procedure chosen for updating state variables, the choice of mass matrix (consistent or lumped) and the resolution of the background mesh.

Bardenhagen, [37] compared two of the widely used algorithms - Update-Stress-First (USF) and Update-Stress-Last (USL), by applying them to a simple harmonic oscillation scenario in 1-dimension. Both algorithms were found to perform well for vibration modes with wavelengths larger than the cellsize. Where the wavelength approached the cellsize (i.e. higher modes of vibration), errors in total system energy increased significantly (oscillating for USF and dissipative for USL).

B.1. 1-DIMENSIONAL VIBRATION

Following the example shown above, the energy conservation of the integration scheme used in this thesis was carried out. The axial vibration of a continuum bar was modelled to validate the explicit MPM algorithm. The test was carried out in 2-dimensions, however setting the Poisson's ratio to zero forced stresses and deformations to occur only in the axial direction. The behaviour of the material was perfectly elastic. The geometry and material points were initiated with no stresses, while the initial velocities for the problem were defined in order to obtain a certain vibration mode in the bar.

	# Elements	# MP/El.	E [Pa]	ν [-]	ρ [m/kg ³]	Δy	Δx	Δt	#Steps
Bar	25 (1x25)	4	100	0.0	1	1	1	0.001	9000

Table B.1: Configuration and material properties for axial bar vibration.

For the explicit time-stepping scheme, the maximum allowed step was estimated using the CFL conditions, which are necessary but not sufficient for stability [7]. An upper bound to the time step is found with the limiting ratio $C_{max} = \frac{c_p \Delta t}{\Delta x}$ is taken as $C_{max} \leq 1$, for a 1-dimensional analysis.

$$\Delta t \leq \frac{\Delta x}{c_p} \quad \text{where,} \quad c_p = \sqrt{\frac{E(1-\nu)}{\rho(1-2\nu)(1+\nu)}} \quad (\text{B.1})$$

The above can be extended to 2-dimensional scenarios. In the analyses used here, an isotropic material is assumed hence $E_x = E_y$, consequently $c_{p,x} = c_{p,y}$. For the quadrilateral elements $\Delta x = \theta \Delta y$, allowing the critical time step to be approximated.

$$C = \frac{c_{p,x} \Delta t}{\Delta x} + \frac{c_{p,y} \Delta t}{\Delta y} \leq C_{max} \quad (B.2)$$

$$\Delta t \leq \frac{\Delta y \theta}{c_p (\theta + 1)} \quad (B.3)$$

B.1.1. ANALYTICAL SOLUTION FOR SMALL DEFORMATIONS

A short overview of the analytical solution was added for completeness. Using separation of variables, expressions for the velocity and displacement along the length of the bar can be derived for a given mode of vibration.

$$u(x, t) = \frac{v_0}{\omega_n} \sin \omega_n t \sin \beta_n x \quad (B.4)$$

$$v(x, t) = v_0 \cos \omega_n t \sin \beta_n x \quad (B.5)$$

The phase $\phi_n(x)$, eigenvalues β_n , frequencies ω_n and mode of vibration n of the bar are related as follows:

$$\phi_n(x) = \sin(\beta_n x), \quad \beta_n = \frac{(2n-1)\pi}{2L}, \quad \omega_n = \beta_n c_p, \quad n = 1, 2, \dots \quad (B.6)$$

The centre-of-mass velocity for the bar can be found by taking the first mass-weighted moment of velocity [37].

$$v_{cm}(t) = \frac{\int_0^L \rho(x) v(x, t) dx}{\int_0^L \rho(x) dx} \quad (B.7)$$

B.1.2. INITIAL CONDITIONS IN NUMERICAL MODEL

From equations B.5 and B.6, the initial conditions for the bar can be set as vibration in the desired mode.

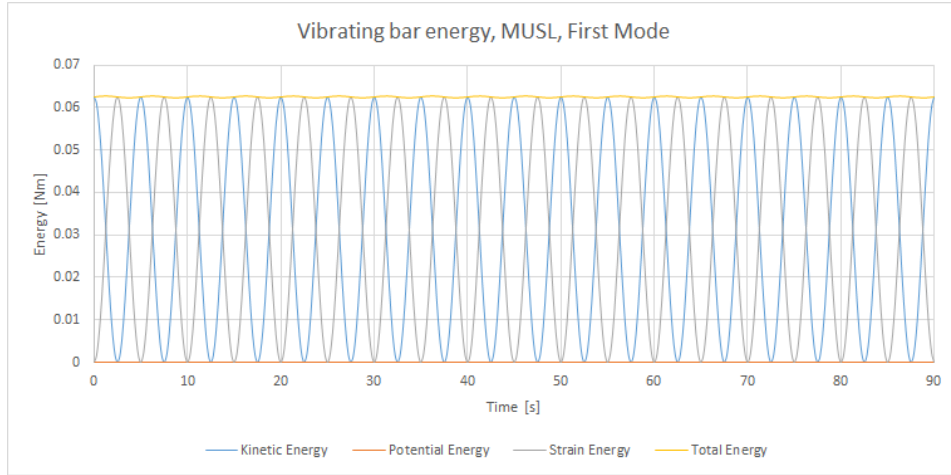
$$v(x, 0) = v_0 \sin \beta_n x = v_0 \sin \left(\frac{(2n-1)\pi}{2L} x \right) \quad (B.8)$$

First the model was tested with a small initial velocity amplitude, defined as $v_0 = 0.1 \text{ m/s}$. This ensured that no cell-crossing and corresponding noise was observed.

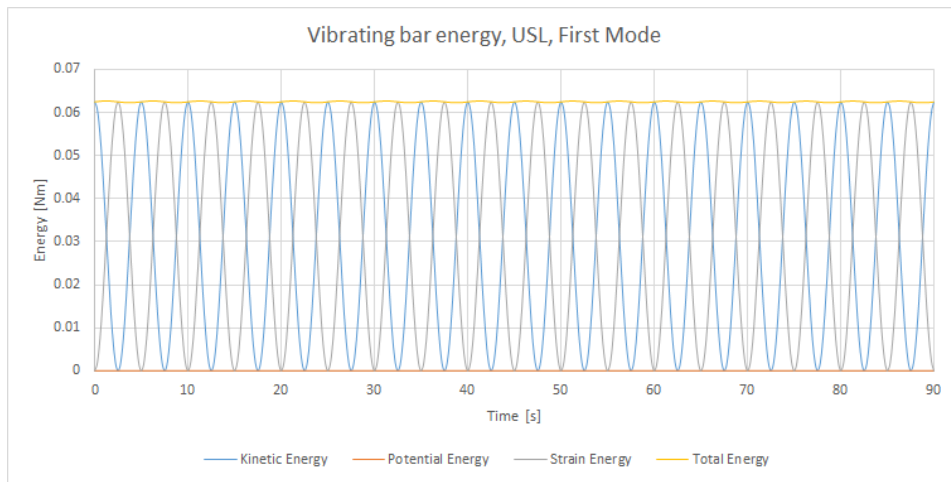
B.2. RESULTS

The USL algorithm implemented in this thesis has been reported by authors to be dissipative [7, 9, 37]. Examining the results of the axial bar vibration confirmed this, with higher vibration modes showing increased damping. That is, as the elastic wave length decreased and approached the size of the background cells, errors in the numerical solution increased significantly. The evolution of energy in the vibrating bar is presented below:

Figure B.1 above shows the variations of kinetic, strain and total energy. The total energy oscillates slightly and has a small downward trend with time. This can be seen more clearly in the motion of the bar centre of mass. Figure B.2 shows a comparison between the numerical and exact solutions for the velocity at the bar centre of mass. In the configuration tested, these showed very good agreement.



(a) MUSL algorithm.

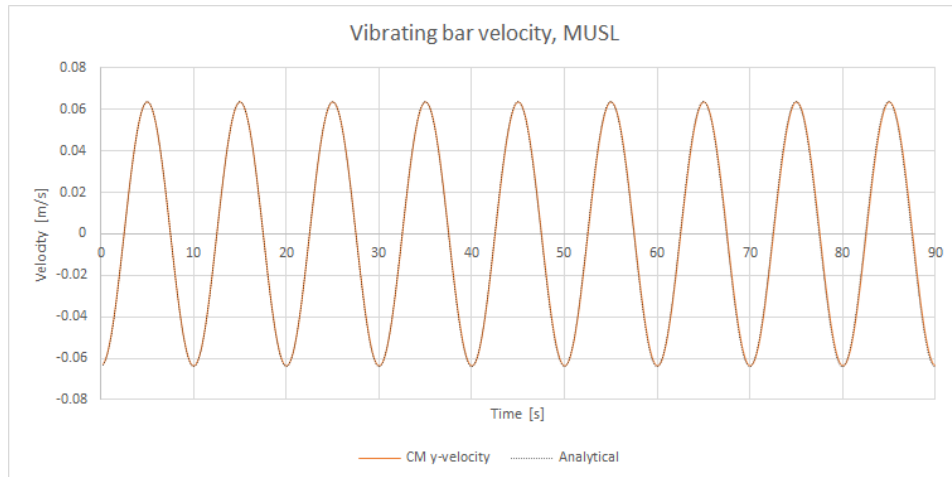


(b) USL algorithm.

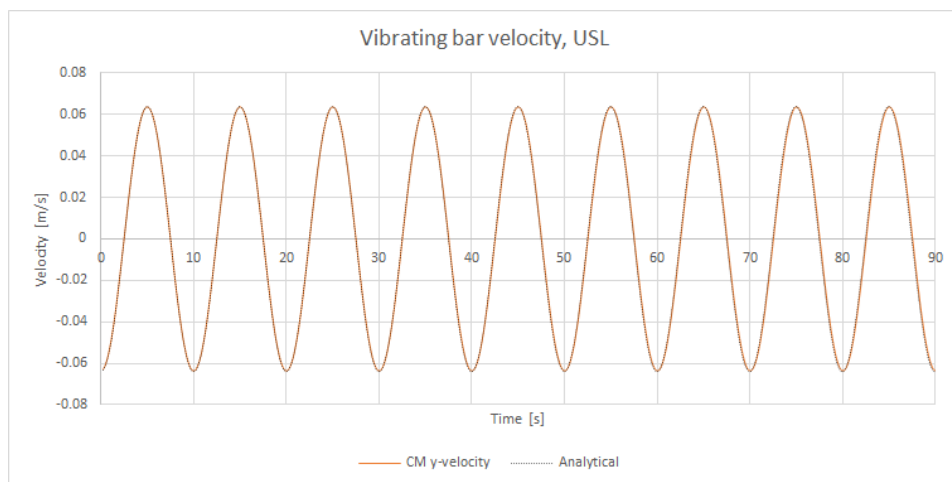
Figure B.1: Bar energy from first mode vibration.

The observed relationship between vibration mode, grid resolution and energy conservation was thought to have significant implications for benchmark contact problems such as the impact of elastic disks. In such a scenario, the collision represented an impulse with the bodies vibrating freely after separation. The energy behaviour of the system depended on the impact velocity, material properties and the accuracy of the contact resolution. Similar plots were produced for the fifth mode of oscillation. Interestingly, the MUSL approach manages to faithfully reproduce the energy and motion of the bar.

Visualising the errors between the different methods, the advantage of the MUSL becomes clear. It should be noted that these results differed from those obtained by [Bardenhagen, \[37\]](#).

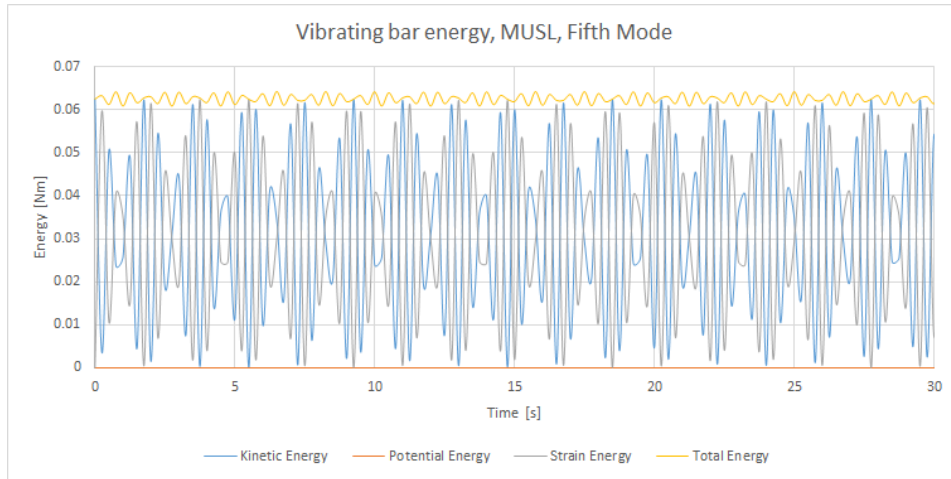


(a) MUSL algorithm.

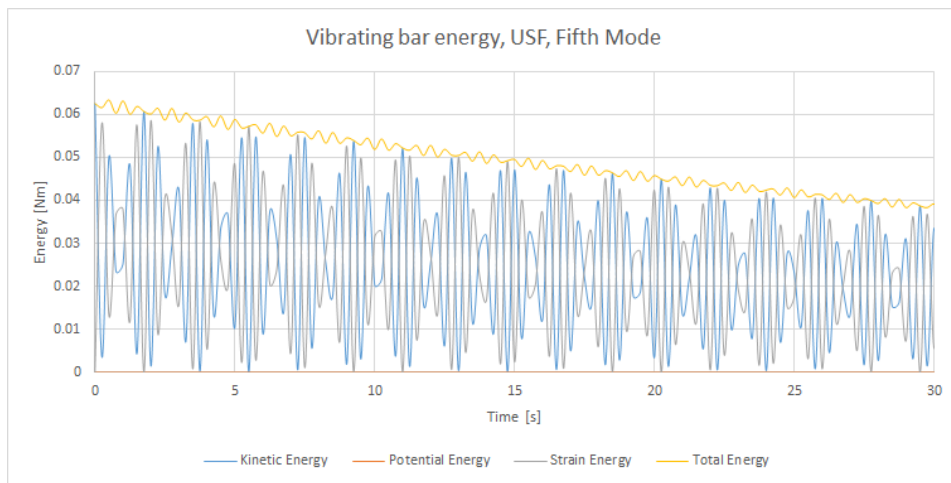


(b) USL algorithm.

Figure B.2: Centre of mass velocity for bar in first mode of vibration.



(a) MUSL algorithm.



(b) USF algorithm.

Figure B.3: Bar energy from fifth mode vibration.

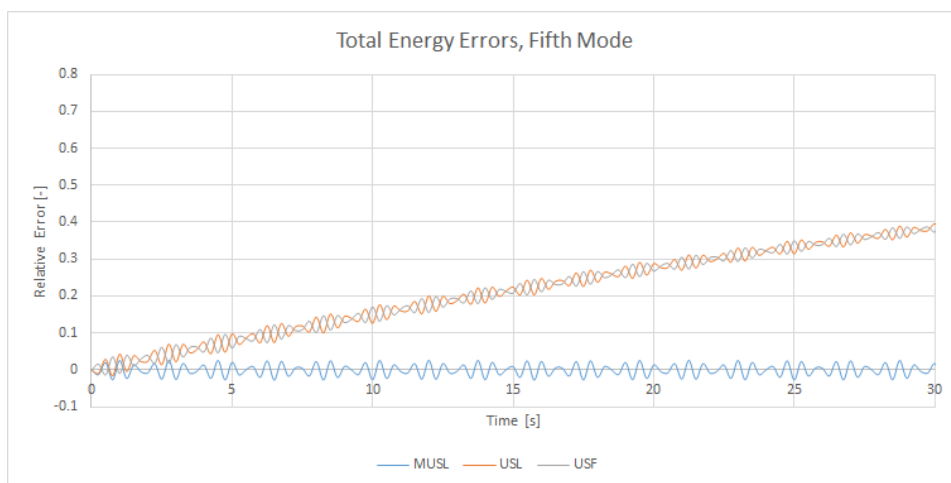


Figure B.4: Comparison of energy errors for fifth mode of vibration.

C

CODE DEVELOPMENT AND DATA STRUCTURES

As an initial step in the implementation and investigation in algorithms for frictional contact, a convenient, flexible programming framework had to be implemented. The following discussion presents and motivates some of the decision made in selecting the program structure. Further recommendations are made for future development of the MPM codebase.

C.1. DATA STRUCTURES

Contact between MPM-discretised bodies is established by independently integrating each body in time and comparing the results. A number of steps within the simulation are repeated for each body and hence can be streamlined by looping over a list of the distinct bodies. This also allows the straight forward definition of objects described by different constitutive models. In order to maintain clarity and a more transparent code structure, data for the bodies was organized using the structure presented in Figure C.1.

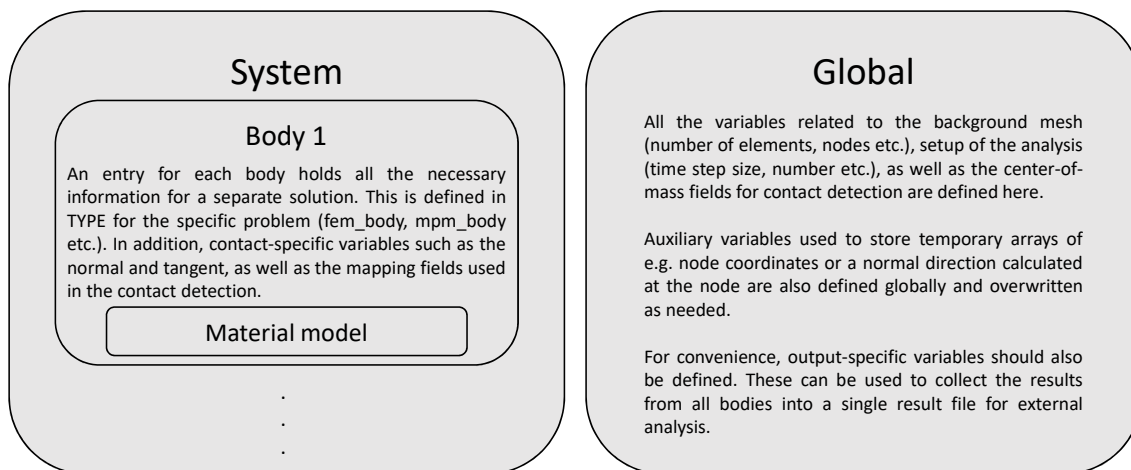


Figure C.1: Overview of data structures used in the program.

The 2003 revision of the Fortran standard allowed the use of Defined Types (similar to classes in C++ and other languages) with fields of allocatable length [60]. This permits the definition of a structure for each object in the simulation, ultimately enabling the straightforward scaling of modelling efforts with minimal changes to the program. The versatility of the code can be further enhanced

by adopting additional concepts of Object Oriented Programming. Class inheritance can be used to append desired features, e.g. different material models, to Defined Types for objects (bodies). While not the main focus of this work, timely implementation of these solutions can be greatly beneficial to the ease of use and subsequent development of the code.

The interaction between bodies is established by mapping the independent solutions onto the common background mesh. This defines the need for several *center-of-mass* variables. The *Multi-velocity field* algorithm as implemented uses the different mass, momentum and velocity fields of to detect contact. For this purpose, arrays of size (n, dim) , where n is the number of nodes in the background mesh and dim - the number of dimensions, are created. Clearly, the required memory will increase with increasing number of bodies and size of the background mesh. Additionally, objects in the simulation may occupy a small part of the full background mesh depending on definition. This means storing field values for the entire background mesh may become inefficient. A possible approach to mitigate increasing storage and processing requirements may be to initiate only a part of the global mesh in a given time step [56].

C.1.1. PROGRAMMING

All current code was developed using the FORTRAN programming language. This makes for high performance code and is well suited to solving problems involving linear algebra. The additional processes involved in modelling contact programs can have a significant impact on running times (e.g. contact detection) and storage (e.g. mapping of fields), hence potential improvements to program efficiency are worth considering at an earlier stage. The independent solutions for each body present good candidates for parallelization. These solutions can be handled in separate threads and executed simultaneously. Once completed these can be assembled into the “system” solution.

Inheritance is another feature that may be useful in implementing different material models. Types can be defined for the material model of choice, allowing bodies to inherit the material properties associated with the desired material model. These features are available within the 2003 revision to the Fortran standard and are supported by most widely used compilers - ifortran, gfortran, with the notable exception of g95. This being said, other programming languages such as C++ offer similar performance, while boasting a large number of available libraries for linear algebra, finite element meshing, computational geometry, plotting and visualisation and others.

C.2. ADDITIONAL LIBRARIES

A set of additional modules were produced in order to facilitate the reuse of the code. These include functions and subroutine handling the computational geometry procedures required in the contact detection and material point generation. Where needed, algorithms were sourced from external repositories which provide access under the GNU LGPL license. Table C.1 shows the procedures included in module **P_IN_POLYGON**, allowing the position of material points w.r.t. a finite element segment to be determined.

Table C.2 shows the procedures included in module **N_RAPHSON**, used for the iterative determination of material point orthogonal projections onto finite element segments. These utilise the complex-step derivative approach to find the derivative of the orthogonal projection vector. The Newton-Raphson method is used to find local coordinates on the of a point on the FE segment, which is the orthogonal projection of a material point. Included subroutines cover 2-dimensional quadrilateral element cases, however can be easily extended to general 3-d elements.

An additional library, called **essential_cvt_subroutines**, facilitating the generation of Centroidal Voronoi Tessellations for the discretisation of arbitrary shape domains was included. Of these, **cvt** and **user** are of particular importance. The former serves as the main driver for the generation,

Table C.1: Subroutines and functions included in module **P_IN_POLYGON**.

Name	Description	Source
<code>polygon_contains_point</code>	Returns TRUE if a point is inside a given polygon, FALSE if not.	Ext.
<code>polygon_point_dist</code>	Finds the smallest distance from a point to the segments of a polygon.	Ext.
<code>polygon_point_near</code>	Finds the closest point which lies on a given polygon.	Ext.
<code>segment_point_dist</code>	Finds the shortest distance from a point to a line segment.	Ext.
<code>segment_point_near</code>	Finds the closest point on a line segment.	Ext.
<code>envelope</code>	Finds a convex hull for a set of points (only extreme points!)	Ext.
<code>find_external_nodes</code>	Finds a convex hull for a set of points (all external points).	Own
<code>points_hull_2d</code>	Finds a convex hull for a set of points (all external points, CCW order).	Ext.
<code>angle_rad_2d</code>	Computes the angle between defined by three points in 2d (radians.)	Ext.
<code>segment_point_coords_2d</code>	Finds the distance from and relative location of a point to a line defined by two points.	Ext.

calling other needed functions. On the other hand, **user** allows the region in which material points and sampling points are generated. Using this in conjunction with the **polygon_contains_point** function listed in Table C.1 enables a domain to be described by points forming a convex hull.

Table C.2: Subroutines and functions included in module **N_RAPHSON**.

Name	Description	Source
check_pair	Returns TRUE if a material point and a segment may come into contact.	Own
find_contact_point	Computes the local and global coordinates of a material point orthogonal projection onto a finite element segment, as well as the normal to the FE segment.	Own
get_global	Computes the global coordinates of a point in/on a finite element from given local coordinates.	Own
get_global_complex	Computes global coordinates of a point in/on a finite element incremented by a complex step.	Own
ortho_project	Computes an approximate orthogonal projection of a material point on a FE segment.	Own
orhto_project_complex	Computes an approximate orthogonal projection of a material point on a FE segment, incremented by a complex step.	Own
ortho_project_linear	Computes the orthogonal projection of a material point on a FE segment (only for linear elements).	Own

D

BACKGROUND MESH

The calculation grid in the Material Point Method as described in this work is analogous to that used in FEM. In MPM however, material state variables are interpolated onto the grid nodes from the material points, rather than integration points. Different functions can be used to facilitate this mapping such as Lagrangian shape functions (as in FEM), B-splines or others. The gradients of these at the element boundaries are of particular importance for the quality of the stress calculation, hence a growing body of research exists exploring the choice of nodal basis functions [28, 29, 39, 43, 61].

D.1. ELEMENTS

A large number of finite elements have been developed by researchers for different applications. As the models presented in this work are 2-dimensional, of interest are the linear (Q4) and quadratic (Q8) quadrilateral elements. These are distinguished by the variation of state variable they can assume across the element. Linear elements are among the most widely used in the Material Point Method due to their relative simplicity and stability. Quadratic elements are better suited for problems where displacement and state variables may vary. In such scenarios, linear elements may show stiffer behaviour [8]. Triangular elements (T3, T6 etc.) may be employed as direct replacements of the above and may allow for accurate discretisation of domains with different shapes. The greatest advantage of using quadrilaterals however, was the seen as the ability to define a structured mesh. This can greatly simplify the procedure for locating material points, although it can introduce mesh-dependent directional effects on the solution.

D.2. SHAPE FUNCTIONS

Conversion between global and local coordinates is carried out using shape functions, nodal coordinates and local coordinates. For example, the global coordinates of any point within an element ($\tilde{x}$) can be found using the nodal coordinates ($x = (x, y, \dots, n_{dim})$) and shape functions (N_i).

$$\tilde{x} = \sum_{i=1}^{nod} N_i x_i \quad (D.1)$$

Converting derivative terms between local and global coordinates is also needed for evaluating strains and respectively stresses. Applying the chain rule, a quantity A in global coordinates can be differentiated with respect to the local coordinates.

$$\frac{\partial A}{\partial \xi} = \frac{\partial x}{\partial \xi} \frac{\partial A}{\partial x} + \frac{\partial y}{\partial \xi} \frac{\partial A}{\partial y} \quad (D.2)$$

Repeating the procedure along other local coordinates allows the assembly of the Jacobian matrix J which facilitates the conversion between global and local derivatives.

$$\begin{Bmatrix} \frac{\partial A}{\partial \xi} \\ \frac{\partial A}{\partial \eta} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{Bmatrix} \frac{\partial A}{\partial x} \\ \frac{\partial A}{\partial y} \end{Bmatrix} = \mathbf{J} \begin{Bmatrix} \frac{\partial A}{\partial x} \\ \frac{\partial A}{\partial y} \end{Bmatrix} \quad (\text{D.3})$$

D.2.1. BILINEAR QUADRILATERAL ELEMENTS

Shape functions are used to convert between global and local coordinates. In a the linear Q4 elements, these are bilinear functions of the local coordinates ξ and η .

$$\begin{aligned} N_1 &= \frac{1}{4}(1-\xi)(1-\eta) \\ N_2 &= \frac{1}{4}(1-\xi)(1+\eta) \\ N_3 &= \frac{1}{4}(1+\xi)(1+\eta) \\ N_4 &= \frac{1}{4}(1+\xi)(1-\eta) \end{aligned}$$

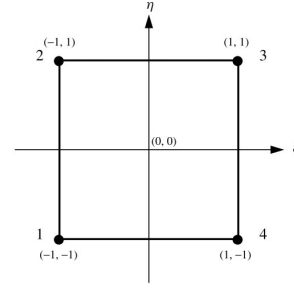


Figure D.1: Q4 element and local coordinate axes.

The derivatives of the shape functions are used in determining stress and strain rates. From the equations shown below it can be seen that the gradients are discontinuous across element boundaries. This has been related to oscillations in stresses calculated for material points crossing cell boundaries and this has been treated by proposing modifications to the shape functions to remedy the discontinuous shape function gradients [28].

$$\begin{aligned} G_{1,\xi} &= \frac{1}{4}(-1+\eta) & G_{1,\eta} &= \frac{1}{4}(-1+\xi) \\ G_{2,\xi} &= \frac{1}{4}(-1-\eta) & G_{2,\eta} &= \frac{1}{4}(1-\xi) \\ G_{3,\xi} &= \frac{1}{4}(1+\eta) & G_{3,\eta} &= \frac{1}{4}(1+\xi) \\ G_{4,\xi} &= \frac{1}{4}(1-\eta) & G_{4,\eta} &= \frac{1}{4}(-1-\xi) \end{aligned}$$

D.2.2. SHAPE FUNCTION GRADIENT DISCONTINUITY

A simple illustration of this problem is given by imagining a material point at the edge of the finite element. This is unlikely, but a useful simplification to remove the influence of other nodes. Without any influence on the discussion, let that edge be the leftmost as shown in Figure D.1, hence the material point will receive contributions from nodes 1 and 2 of the element shown. At the same time, if the material point is considered to be within the element on the left, nodes 3 and 4 of that element will be contributing. If the material point has local coordinates $(\xi_r, \eta_r) = (-1, 0)$ in the right element, corresponding to $(\xi_l, \eta_l) = (1, 0)$ in the left one then:

$$\begin{aligned} \varepsilon_p &= \sum_{i=1}^{nod} G_i v_i \Delta t \\ \varepsilon_{p,right} &= \frac{1}{4} [(-1+\eta) v_{1,right} + (-1-\eta) v_{2,right}] \Delta t \\ \varepsilon_{p,left} &= \frac{1}{4} [(1+\eta) v_{3,left} + (1-\eta) v_{4,left}] \Delta t \end{aligned}$$

Considering the node numbering used here, $v_{1,right} = v_{4,left}$ and $v_{2,right} = v_{3,left}$. The expressions above can be re-written:

$$\begin{aligned}\varepsilon_{p,right} &= \frac{1}{4} [(-1 + \eta_r) v_{1,right} + (-1 - \eta_r) v_{2,right}] \Delta t \\ \varepsilon_{p,left} &= \frac{1}{4} [(1 + \eta_l) v_{2,right} + (1 - \eta_l) v_{1,right}] \Delta t\end{aligned}$$

Substituting for $\eta_l = \eta_r = 0$ in both equations, we see the different strain evaluated at the material point from the two elements, despite receiving contributions from the same nodes.

$$\begin{aligned}\varepsilon_{p,right} &= \frac{1}{4} [(-1) v_{1,right} + (-1) v_{2,right}] \Delta t \\ \varepsilon_{p,left} &= \frac{1}{4} [v_{2,right} + v_{1,right}] \Delta t\end{aligned}$$

These errors are then propagated into the calculation of the material point stress, which can in turn affect the entire solution through the assembled internal forces. The highlighted issue holds for both explicit and implicit analyses, however in the latter equilibrium of the nodal forces is sought and this may help alleviate problems with stability.

D.2.3. QUADRATIC QUADRILATERAL ELEMENTS

In contrast, the shape functions of the Q8 element are continuous across the element boundaries. As in FEM, use of higher order shape functions could improve convergence and the quality of mapping from MPs to nodes.

$$\begin{aligned}N_1 &= \frac{1}{4}(1 - \xi)(1 - \eta)(-\xi - \eta - 1) \\ N_2 &= \frac{1}{2}(1 - \xi)(1 - \eta^2) \\ N_3 &= \frac{1}{4}(1 - \xi)(1 + \eta)(-\xi + \eta - 1) \\ N_4 &= \frac{1}{2}(1 - \xi^2)(1 + \eta) \\ N_5 &= \frac{1}{4}(1 + \xi)(1 + \eta)(\xi + \eta - 1) \\ N_6 &= \frac{1}{2}(1 + \xi)(1 - \eta^2) \\ N_7 &= \frac{1}{4}(1 + \xi)(1 - \eta)(\xi - \eta - 1) \\ N_8 &= \frac{1}{2}(1 - \xi^2)(1 - \eta)\end{aligned}$$

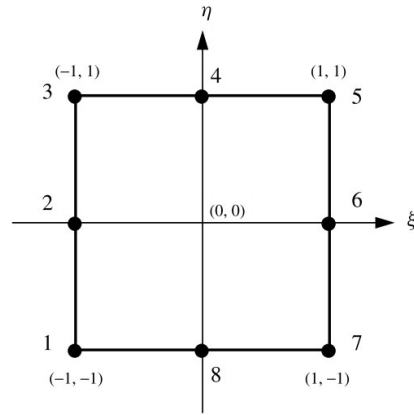


Figure D.2: Q8 element and local coordinate axes.

Shape functions for the corner nodes ($N_i, i = 1, 3, 5, 7$) can assume negative values within the element. This poses a problem when using these higher order elements in MPM, where sub-optimal locations of material points may result in a contribution with an inverted sign mapped onto such a node and cause significant errors in the overall solution. In contrast, the Finite Element Method integrates equations at the Gauss points within the element. These are selected at locations of optimal quadrature and stay at the same position with respect to the element nodes.

E

CONSTITUTIVE MODELS AND RELATIONS

E.1. LINEAR ELASTIC MODEL

The linear elastic model was used to relate stresses and strains in initial implementations. In a 2-dimensional plane-strain scenario, the elastic stress-strain matrix D^e can be expressed:

$$D^e = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{(1-2\nu)}{2} \end{bmatrix} \quad (\text{E.1})$$

This material model allowed elastic collisions to be modelled as cases where total energy remains unchanged.

E.2. LINEAR ELASTIC-PERFECTLY PLASTIC MODEL

Adding a failure criterion to the Linear Elastic Model allows for the onset of plastic deformations. Integration of the elastoplastic strain increments was done via the widely used *Initial stress* method [8]. This approach was chosen due to its relative simplicity as it describes an explicit relationship between stress and strain increments.

$$\Delta\sigma = D^{pl}\Delta\varepsilon = (D^e - D^p)\Delta\varepsilon \quad (\text{E.2})$$

Here, D^e , D^p and D^{pl} represent the elastic, plastic and elastoplastic stress-strain matrices. Assuming stress changes only occur due to elastic strain increments and allowing for non-associated flow, the plastic stress-strain matrix can be derived as:

$$D^p = \frac{D^e \left(\frac{\partial Q}{\partial \sigma} \right) \left(\frac{\partial F}{\partial \sigma} \right)^T D^e}{\left(\frac{\partial F}{\partial \sigma} \right)^T D^e \left(\frac{\partial Q}{\partial \sigma} \right)} \quad (\text{E.3})$$

Where F and Q are the yield and plastic potential surfaces respectively. Any strain increments incurred post-failure ($F=0$) do not produce a change in stress. This behaviour corresponds to perfect plasticity. Where hardening/softening behaviour is sought, a different method of integrating plastic strains would be required. If yield was found to occur ($F \geq 0$) a factor f was found relating the "overshoot" of the yield surface. This factor was then used to scale the new stress increment to ensure it lays on the failure surface.

$$\Delta\sigma_{ep}^{t+\Delta t} = \sigma^t + (1-f)\sigma_e^{t+\Delta t} \quad (\text{E.4})$$

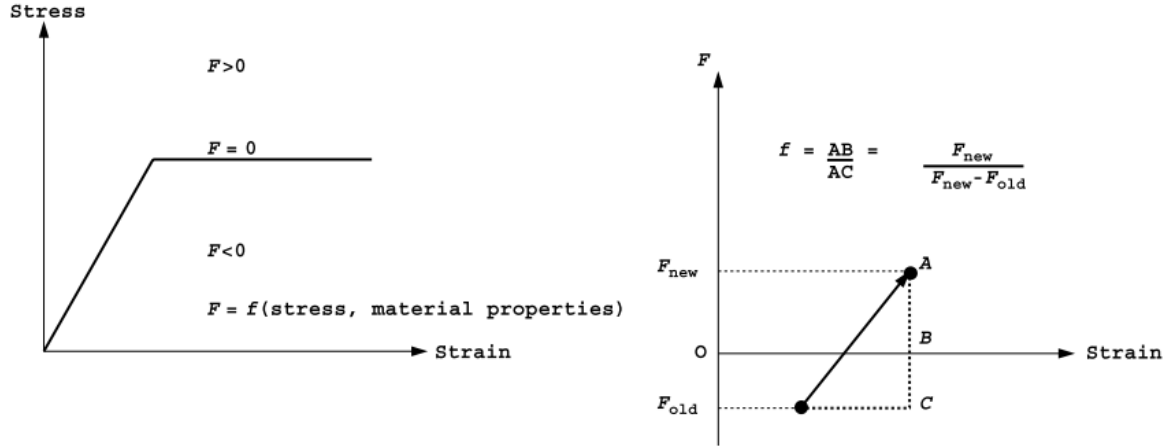


Figure E.1: Perfect plasticity and the initial stress method [8].

E.3. YIELD CRITERIA

The yield criteria used in the code are defined in accordance with [Smith et al., \[8\]](#). The stress state is defined in the stress invariants (s, t, θ) as follows:

$$s = \frac{1}{\sqrt{3}} (\sigma_x + \sigma_y + \sigma_z) \quad (\text{E.5})$$

$$t = \frac{1}{\sqrt{3}} \left[(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6\tau_{xy}^2 + 6\tau_{yz}^2 + 6\tau_{zx}^2 \right]^{1/2} \quad (\text{E.6})$$

$$\theta = \frac{1}{3} \arcsin \left(\frac{-3\sqrt{6}J_3}{t^3} \right) \quad (\text{E.7})$$

$$J_3 = s_x s_y s_z - s_x \tau_{yz}^2 - s_z \tau_{xz}^2 + 2\tau_{xy} \tau_{yz} \tau_{zx} \quad (\text{E.8})$$

$$s_x = \frac{(2\sigma_x - \sigma_y - \sigma_z)}{3} \quad (\text{E.9})$$

Based on these, the program calculates the σ_m (mean stress) and $\bar{\sigma}$ (deviator stress), based on triaxial loading. The failure criteria used are subsequently defined with respect to these:

$$\sigma_m = \frac{1}{\sqrt{3}} s \quad \text{and} \quad \bar{\sigma} = \sqrt{\frac{3}{2}} t \quad (\text{E.10})$$

E.3.1. VON MISES YIELD CRITERION

The Von Mises yield criterion is frequently used in saturated clays, which behave in a frictionless manner. It is pressure independent, hence the lack of σ_m in the definition. This approach was readily implemented in the code produced. Once strains at the material points were evaluated from the nodal velocities, an elastic stress increment could be found leading to a trial stress state. The stress invariants σ_m (mean stress), $\bar{\sigma}$ (deviator stress) and θ (Lode angle) were then evaluated and the yield function found as:

$$F_{vm} = \bar{\sigma} - \sigma_Y \quad (\text{E.11})$$

E.3.2. MOHR COULOMB YIELD CRITERION

The Mohr Coulomb yield criterion is one of the most widely used criteria for capturing the onset of plasticity in frictional materials. It is defined in principal stress terms as:

$$F_{mc} = \frac{\sigma_1 + \sigma_3}{2} \sin \phi - \frac{\sigma_1 - \sigma_3}{2} - c \cos \phi \quad (\text{E.12})$$

Following the discussion in the previous section, the MC failure criterion is expressed in the σ_m and $\bar{\sigma}$ invariants as:

$$F_{mc} = \sigma_m \sin \phi + \bar{\sigma} \left(\frac{\cos \theta}{\sqrt{3}} - \frac{\sin \theta \sin \phi}{3} \right) - c \cos \phi \quad (\text{E.13})$$

F

VARIABLE UPDATE ALGORITHMS

The procedures outlined in Algorithms [E1](#) below was used at each timestep for the duration of the analysis. It is based on the Update-Stress-Last algorithm originally proposed by [Sulsky *et al.*, \[30\]](#). It was modified to include contact detection and resolution. Similarly, the Modified-Update-Stress-Last procedure is shown in Algorithm [E2](#).

- **Forward Mapping** from material points to grid nodes
 - Map external body loads: $F_{i,grav}^t = \sum_{p=1}^{N_p} N_i(x_p^t) m_p g$
 - Map internal body loads: $F_{i,int}^t = \sum_{p=1}^{N_p} B^T(x_p^t) \sigma_p^t V_p^t$
 - Map momentum: $(mv)_i^t = \sum_{p=1}^{N_p} N_i(x_p^t) m_p v_p$
 - Map mass (Lumped mass matrix): $m_i^t = \sum_{p=1}^{N_p} N_i(x_p^t) m_p$
- **Explicit Integration** to find solutions at next time step
 - Resolve nodal forces: $m_i^t a_i^t = F_{i,ext}^t - F_{i,int}^t = F_i^t$
 - Find nodal acceleration: $a_i^t = F_i^t / m_i^t$
 - Update nodal velocity: $v_i^{t+\Delta t} = v_i^t + a_i^t \Delta t$
 - Update nodal momentum: $(mv)_i^{t+\Delta t} = (mv)_i^t + F_i^t \Delta t$
- **Assemble Center-of-Mass Fields** for contact detection
 - Momentum field: $(mv)_{cm,i}^{t+\Delta t} = \sum_{b=1}^{N_b} (mv)_{b,i}^{t+\Delta t}$
 - Mass field: $m_{cm,i}^t = \sum_{b=1}^{N_b} m_{b,i}^t$
 - Velocity field: $v_{cm,i}^{t+\Delta t} = (mv)_{cm,i}^{t+\Delta t} / m_{cm,i}^t$
- **Contact Resolution**
 - Detect potential contact: $v_{b,i}^{t+\Delta t} - v_{cm,i}^{t+\Delta t} > 0$
 - Find normal direction at node*: $n_{b,i} = \sum_{p=1}^{N_i^b} G_{ip}^b m_p^b / |G_{ip}^b m_p^b|$
 - Find tangential direction at node: $t_{b,i} = (v_{b,i}^{t+\Delta t} - v_{cm,i}^{t+\Delta t}) \times n_{b,i} / |v_{b,i}^{t+\Delta t} - v_{cm,i}^{t+\Delta t}|$
 - Relative normal velocity: $dv_{b,i}^n = (v_{b,i}^{t+\Delta t} - v_{cm,i}^{t+\Delta t}) \cdot n_{b,i} > 0$
 - Relative tangential velocity: $dv_{b,i}^t = (v_{b,i}^{t+\Delta t} - v_{cm,i}^{t+\Delta t}) \cdot t_{b,i} > 0$
 - Effective coefficient of friction: $\bar{\mu} = \text{sign}(dv_{b,i}^t) \min(\mu, dv_{b,i}^t / dv_{b,i}^n)$
 - Correct nodal velocity: $\bar{v}_{b,i}^{t+\Delta t} = v_{b,i}^{t+\Delta t} - dv_{b,i}^n (n_{b,i} + \bar{\mu} t_{b,i})$
 - Correct nodal acceleration: $\bar{a}_{b,i}^{t+\Delta t} = (\bar{v}_{b,i}^{t+\Delta t} - v_{b,i}^t) / \Delta t$
- **Reverse Mapping** from nodes to material points
 - Update particle velocity: $v_p^{t+\Delta t} = v_p^t + \sum_{i=1}^{N_n} N_i(x_p^t) \bar{a}_i^{t+\Delta t} \Delta t$
 - Update particle positions: $x_p^{t+\Delta t} = x_p^t + \sum_{i=1}^{N_n} N_i(x_p^t) \bar{v}_i^{t+\Delta t} \Delta t$
- **Stress Calculation**
 - Strain increment: $\Delta \epsilon_p = \sum_{i=1}^{N_n} \nabla N_i(x_p^t) \bar{v}_i^{t+\Delta t} \Delta t$
 - Stress increment: $\Delta \sigma_p = D \Delta \epsilon_p$
 - Update stress state: $\sigma_p^{t+\Delta t} = \sigma_p^t + \Delta \sigma_p$
 - Update volume: $V_p^{t+\Delta t} = V_p^t (1 + \sum_{j,k=1,dim} \delta_{jk} \Delta \epsilon_{p,jk})$

- **Forward Mapping** from material points to grid nodes
 - Map external body loads: $F_{i,grav}^t = \sum_{p=1}^{N_p} N_i(x_p^t) m_p g$
 - Map internal body loads: $F_{i,int}^t = \sum_{p=1}^{N_p} B^T(x_p^t) \sigma_p^t V_p^t$
 - Map momentum: $(mv)_i^t = \sum_{p=1}^{N_p} N_i(x_p^t) m_p v_p$
 - Map mass (Lumped mass matrix): $m_i^t = \sum_{p=1}^{N_p} N_i(x_p^t) m_p$
- **Explicit Integration** to find solutions at next time step
 - Resolve nodal forces: $m_i^t a_i^t = F_{i,ext}^t - F_{i,int}^t = F_i^t$
 - Find nodal acceleration: $a_i^t = F_i^t / m_i^t$
 - Update particle velocities*: $v_p^{t+\Delta t} = v_p^t + \sum_{i=1}^{N_e n} N_i(x_p^t) a_i^t \Delta t$
 - Update nodal velocities*: $v_i^{t+\Delta t} = m_i^{-1,t} \sum_{p=1}^{N_e p} m_p N_p(x_p^t) v_i^{t+\Delta t}$
- **Assemble Center-of-Mass Fields** for contact detection
 - Momentum field: $(mv)_{cm,i}^{t+\Delta t} = \sum_{b=1}^{N_b} (mv)_{b,i}^{t+\Delta t}$
 - Mass field: $m_{cm,i}^t = \sum_{b=1}^{N_b} m_{b,i}^t$
 - Velocity field: $v_{cm,i}^{t+\Delta t} = (mv)_{cm,i}^{t+\Delta t} / m_{cm,i}^t$
- **Contact Resolution**
 - Detect potential contact: $v_{b,i}^{t+\Delta t} - v_{cm,i}^{t+\Delta t} > 0$
 - Find normal direction at node**: $n_{b,i} = \sum_{p=1}^{N_i^b} G_{ip}^b m_p^b / \left| \sum_{p=1}^{N_i^b} G_{ip}^b m_p^b \right|$
 - Find tangential direction at node: $t_{b,i} = (v_{b,i}^{t+\Delta t} - v_{cm,i}^{t+\Delta t}) \times n_{b,i} / \left| v_{b,i}^{t+\Delta t} - v_{cm,i}^{t+\Delta t} \right|$
 - Relative normal velocity: $dv_{b,i}^n = (v_{b,i}^{t+\Delta t} - v_{cm,i}^{t+\Delta t}) \cdot n_{b,i} > 0$
 - Relative tangential velocity: $dv_{b,i}^t = (v_{b,i}^{t+\Delta t} - v_{cm,i}^{t+\Delta t}) \cdot t_{b,i} > 0$
 - Effective coefficient of friction: $\bar{\mu} = \text{sign}(dv_{b,i}^t) \min(\mu, dv_{b,i}^t / dv_{b,i}^n)$
 - Correct nodal velocity: $\bar{v}_{b,i}^{t+\Delta t} = v_{b,i}^{t+\Delta t} - dv_{b,i}^n (n_{b,i} + \bar{\mu} t_{b,i})$
 - Correct nodal acceleration: $\bar{a}_{b,i}^{t+\Delta t} = (\bar{v}_{b,i}^{t+\Delta t} - v_{b,i}^t) / \Delta t$
- **Reverse Mapping** from nodes to material points
 - Update particle velocity: $v_p^{t+\Delta t} = v_p^t + \sum_{i=1}^{N_n} N_i(x_p^t) \bar{a}_i^{t+\Delta t} \Delta t$
 - Update particle positions: $x_p^{t+\Delta t} = x_p^t + \sum_{i=1}^{N_n} N_i(x_p^t) \bar{v}_i^{t+\Delta t} \Delta t$
- **Stress Calculation**
 - Strain increment: $\Delta \epsilon_p^{t+\Delta t} = \sum_{i=1}^{N_n} \nabla N_i(x_p^t) \bar{v}_i^{t+\Delta t} \Delta t$
 - Stress increment: $\Delta \sigma_p^{t+\Delta t} = D \Delta \epsilon_p^{t+\Delta t}$
 - Update stress state: $\sigma_p^{t+\Delta t} = \sigma_p^t + \Delta \sigma_p^{t+\Delta t}$
 - Update volume: $V_p^{t+\Delta t} = V_p^t \left(1 + \sum_{j,k=1,dim} \delta_{jk} \Delta \epsilon_{p,jk}^{t+\Delta t} \right)$

G

ELASTOPLASTIC BLOCK COLLISION

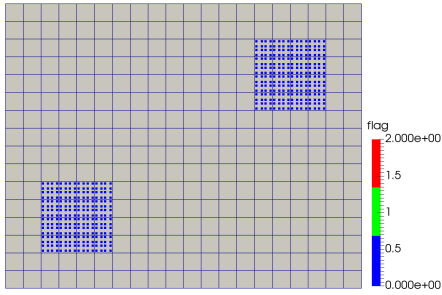
The collision of two blocks moving diagonally was modelled in order to demonstrate the effects of applying friction along the contact interface. *Standard* MPM corresponds to a stick case, whereas the slip scenario cannot be obtained without a contact algorithm. Further, the material was allowed to yield and demonstrated different patterns of deformation post-collision. Figure G.1 shows two elastoplastic blocks colliding at different times with different frictional regimes between the bodies. The two extreme cases - frictionless(a,c,e,g) and stick(b,d,f,h) are shown to illustrate the range of solutions attainable. Coloured flags indicate the loading mode of the material, with blue - elastic loading, green - plastic loading, red - elastic unload-reload.

The material properties used in the above analysis are shown in Table G.1 below. There were no specific restrictions for the units chosen other than the need for a consistent set, however working in SI units was found most convenient. A Von Mises yield criterion with a yield stress of $200Pa$ was used in the analysis with perfect plasticity.

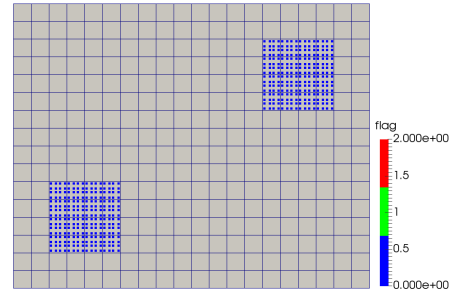
Body	# El.	# MP/El.	E [Pa]	ν [-]	ρ [kg/m ³]	v_0 [m/s]	Δy	Δx	Δt	#Steps
1	16 (4x4)	16	1000	0.2	1	(1.0,1.0)	0.25	0.25	$5E-4$	9000
2	16 (4x4)	16	1000	0.2	1	(-1.0,-1.0)	0.25	0.25	$5E-4$	9000

Table G.1: Configuration and material properties for elastoplastic block collision.

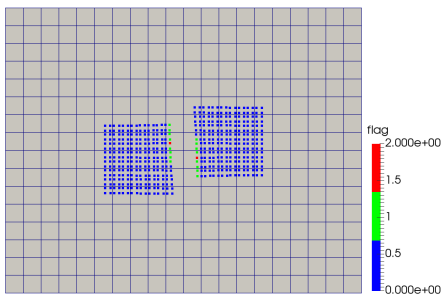
In the two analyses carried out, differences can be noted in the behaviour of the two block after collision. Notably, the rotation of the blocks seen in the stick case is not observed in the slip scenario.



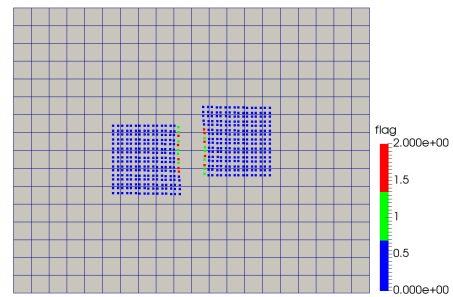
(a) Slip at Time 0.000s.



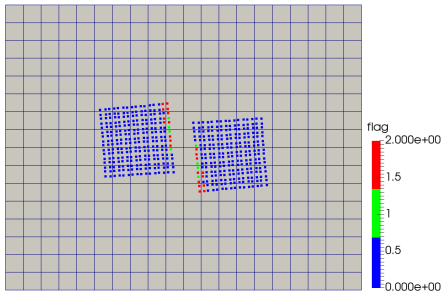
(b) Stick at Time 0.000s.



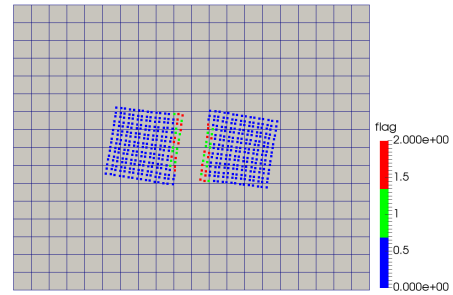
(c) Slip at Time 0.875s.



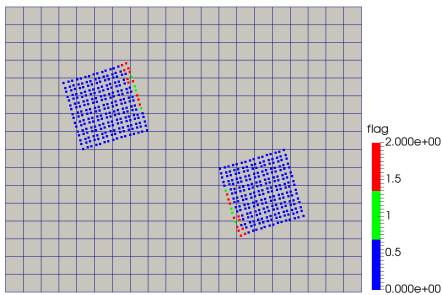
(d) Stick at Time 0.875s.



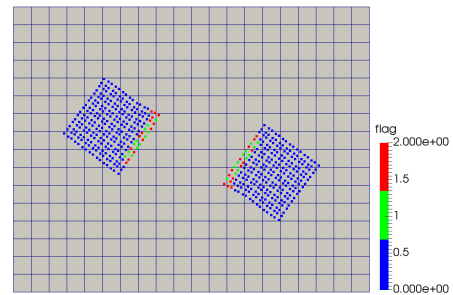
(e) Slip at Time 1.125s.



(f) Stick at Time 1.125s.



(g) Slip at Time 1.75s.



(h) Stick at Time 1.75s.

Figure G.1: Elastoplastic contact between blocks with different friction modes.