Non Conforming Boundary Conditions in Finite Element and Particles Methods

Nicola Comerci, 1237324

A.Y. 2021/2022

University of Padua
Civil, Environmental and Architectural Engineering.
Mathematical Engineering.

Techinal Univeristy of Munich (Host Institution)
Civil, Geo and Environmental Engineering.
Structural Analysis Chair.

Supervisor:
Antonia Larese

Supervisor Host Institution:
Veronika Singer

Abstract

Many computational engineering problems deal with the mechanical response of a body with unfitted boundary. With the term "unfitted" we mean that its boundary does not coincide with the underlying mesh nodes. This might happen if the body undergoes large displacements or large strain, just to make an example. Under these circumstances accurately imposing the boundary conditions can be a challenging task. This problem can be solved using methods derived from optimization theory such as the Penalty approach and Lagrange multipliers.

The purpose of this thesis is to revise the mathematical and numerical formulations that solve the problem of nonconforming boundary conditions in the classical Finite Element Method (FEM) and to extend these approaches to a continuum-based particle method such as the Material Point Method (MPM).

In fact, while the Finite Element Method is a widely accepted and well established techniques in many engineering fields, it shows some limitations when dealing with large strain regime. The Material Point Method, on the contrary, is an hybrid technique which uses two discretizations and it is especially suited for this kind of problems. MPM can be seen as a special Finite Element Method where the quadrature points are moving material points.

In this thesis we will focus in particular on appending boundary conditions by means of the Lagrange multipliers. Then, a comparison between the Penalty method and Lagrange multipliers in MPM is conducted. The convergence property is studied in details and compared with other existing techniques. For this purpose we simulate classical benchmark problems in computational mechanics. It was observed that the Lagrange multipliers method exhibits quadratic convergence even for coarse computational meshes. In contrast, this is not the case for Penalty method, which shows quadratic convergence only for fairly fine meshes.

Dedication

I dedicate this thesis work to all the women who have always been present in my life: my mother, my grandmother, my sister and last but not least my life partner. To them my most sincere thanks for always being an indispensable constant in my life. The end of the writing of this thesis work marks a new beginning.

Acknowledgements

First of all, I would like to thank the European Union, which through the Erasmus project allowed me to do this thesis work at the Technical University of Munich (TUM).

At TUM I met my supervisor, PhD Veronika Singer to whom goes my greatest gratitude for patiently mentoring me during my months in Munich. She followed me with dedication and extreme patience during the writing of this thesis.

Of course, all this would not have been possible if Professor Antonia Larese had not given me the opportunity to do this thesis work. Therefore, my sincere thanks also go to her for making this unforgettable experience possible.

Contents

List of Figures

Chapter 1

Introduction

1.1 General Introduction

The imposition of boundary or interface conditions in engineering problems turns out to not be an easy task. Often, the boundary of the considered body does not coincide with the boundary of the computational mesh. In this case we will speak of nonconforming boundary conditions. An engineering field where this situation is frequent is for instance geotechnics. Just think about a portion of slope that slides downstream: we will have a granular flow whose boundary will hardly coincide with that of the computational mesh. The case just described is therefore characterized by of extreme large deformations and displacements. Various proposals have been made for the imposition of nonconforming boundary conditions, many of which derive from optimization theory.
In this thesis we will mainly discuss the method of Lagrange Multipliers, already widely discussed in the literature with regard to its application in the Finite Element Method but little explored in the context of Particle methods. The latter, are to be preferred whenever the displacements and deformations occurring in the continuum are extremely large. This is due to the fact that the Finite Element Method reaches its limit when large strain occur due to the large mesh distortion. Particle methods, such as the Material Point Method (MPM) that we will discuss in this thesis, on the contrary, are particularly appropriate because the relevant information are not stored in the computational mesh but in the so-called material points. Therefore, the computational mesh can be reset to its original position whenever it is

excessively deformed. A common alternative to Lagrange Multipliers is represented by the Penalty. It is very popular and widely used in many engineering problems due to its simplicity. This method has been used by Chandra et al. [14] in order to append nonconforming boundary conditions within the particles mechanics framework. Thus, the main goal of this thesis is to formulate the problem of nonconforming boundary conditions treated by means of Lagrange Multipliers applied to the Finite Element Method and, mainly, to the Material Point Method. The Lagrange Multipliers method, applied to impose the boundary or interface conditions within the FEM framework, has been widely studied since the initial work by Brezzi [9] and Babuska [1]. They stated the famous and celebrated Ladyshenskaya-Brezzi-Babuska condition which is needed to be satisfied in order to have a well-posed problem as we will see in Chapter 5. Thus, this thesis wants to extend the knowledge gained from the Lagrange Multipliers method in FEM to MPM and compare the numerical results with respect other optimization method such as the Penalty. The structure of the document is presented in the following paragraphs. The second chapter deals with the continuum mechanics which is the starting point for the formulation of any mechanical problem. In this chapter we address the classical topics of mechanics such as motion, stress, balance equations and finally constitutive relations.

In the third chapter we introduce the concept of directional derivative and its use for linearization of the principle of virtual works. Engineering problems are often characterized by nonlinearities. The directional derivative provides an important tool for linearizing the governing equations. Once the principle of virtual work has been linearized its Finite Elements discretization can be easily obtained (both in FEM and MPM), and finally the Newton-Raphson scheme and Newmark integration are introduced.

Chapter 4 presents the state of the art of the Material Point Method in its explicit and implicit formulations. The differences with the Finite Element Method are highlighted.

Chapter 5 discusses Lagrange Multipliers applied to computational mechanics and their use in imposing conforming and nonconforming boundary conditions. The problems that arise from their use are then discussed. In fact, the use of Lagrange Multipliers leads to a saddle point problem whose solution is well-posed under certain conditions. We will talk about the famous Ladyshenskaya-Brezzi-Babuska condition mentioned before and how to respect it in order to have a well-posed problem. Finally, we will apply the Lagrange Multipliers both to the FEM and to the MPM and we will highlight the solving strategies used then at computational level.

Chapter 6 is dedicated to some engineering applications where the nonconforming boundary conditions have been appended by means of Lagrange Multiplier. In particular, we will estimate the order of convergence for cantilever beam under self weight and we will compare it with the one obtained with the Penalty method. Next, we will show another example where the clamped beam has been considered. In this case we will deal with conforming and nonconforming boundary conditions, one for each side of the clamped beam. Finally, we will take into account a collapsing granular column and we will compare the results with those obtained by Bui et al. [11]. The last chapter will summarize the main achievements of the thesis as well it will state the future research lines on this topic.

1.2 KratosMultiphysics

The numerical applications in chapter 6 were conducted within the Kratos Multiphysics framework. Kratos Multiphysics, or simply Kratos, is a engineering simulation softwere based on Finite Element Method. It is written mostly in C++ and Python by using an Object Oriented Programming (OOP). In this way modularity, generality, flexibility and reusability are guaranteed. Furthermore, in order to allow HPC (High Performance Computing), Kratos supports parallel computing mostly based on OpenMP directives. Moreover, Kratos is open source under the BSD (Berkeley Softwere Distribution) license. Thank to this license the existing code can be used and distributed without any restrictions in such way that developers, researchers and students can give their contributions. For any further information, refer

Figure 1.1: Kratos MultiPhysics logo.

to [16], [15] and [20]. Finally, most of the applications in Kratos are prepared to be used in GiD, a pre and post-processor. All the numerical applications present in this thesis were conducted using GiD as pre-processor in order to set the geometry problem and ad post-processor in such way to visualize the results.

Chapter 2

Continuum Mechanics

Real world phenomena can be described with a *microscopic* and *macroscopic* approach. The former is more appropriate if we want to study matters at the atomic and subatomic level while the latter is suitable for studying macroscopic quantities. Macroscopic system can be described by means of the *continuum mechanics*. Continuum mechanics is a discipline that allows us to give a macroscopic description of the bodies. With 'macroscopic' we mean variables such as velocity, density, temperature, displacement, deformations and so on. Therefore, continuum mechanics is an extremely important tool in many engineering fields. Let us think, for example, how important is to know the deformation value during the design phase. Examples could be: deformation of some components during an industrial process, deformation of a building undergoing some load, or again, the deformation of the vehicle chassis during a crash test. It is evident that continuum mechanics play a huge role in engineering and it is the starting point of many CAE (Computer-Aided Engineering) simulation such as FEA (Finite Element Analysis) and CFD (Computational Fluid Dynamics) and more in general of many numerical simulation.

The study of continuum mechanics can be divided into four categories [23]. The first one refers to the study of the *kinematics*, meaning the study of the body motion. The second category is the study of a fundamental concept that is the *stress*. The third category is related to the *balance principles* and the last one is inherent to *constitutive law* meaning the mathematical relationships connecting stress with deformation.

2.1 Kinematics

Kinematics is the study of motion and deformation of the body. Thus, in this section we are going to state some important and basic definitions that will turn out to be useful in order to describe mechanical phenomena.

2.1.1 Introduction

In continuum mechanics there are two main approach to describe physical phenomena

- Eulerian or Spatial : phenomena are described maintaining the attention on spatial position of euclidean space. In this approach we observe what happens to the various material points crossing the fixed euclidean spatial position.
- Lagrangian or Material: phenomena are described keeping the attention on the material points belonging to the body and we track its motion.

The next ingredient to describe the motion is a **reference frame** and coordinates. We can have two reference frame.

The **initial** reference frame is used when we want to adopt a Lagrangian description and the axes coordinates are formed from the orthonormal right-handed vector $\boldsymbol{e}_I$ with $I = 1, 2, 3$. Thus, the position vector will be $\boldsymbol{X} = X_I \boldsymbol{e}_I$.

The **deformed** reference frame is used when we use an Eulerian approach. The axes coordinates are formed from the orthonormal right-handed vector basis $\boldsymbol{e}_i$ with $i = 1, 2, 3$. The position vector will be $\boldsymbol{x} = x_i \boldsymbol{e}_i$.

These are typical notation in continuum mechanics and as we can see the capital letters will be used for describe the body in its initial configuration while the small letters will be used for describe the body in the current, meaning deformed, configuration. Moreover, we indicate the body with $\mathcal{B}$ in the current configuration, while we indicate the body in its initial configuration with letter $\mathcal{B}_0$.

2.1.2 Motion

Let us suppose that the body undergoes to some mechanical actions and occupies the position $\mathcal{B}_0$. Then, due to these mechanical actions, it will occupy a new configuration, $\mathcal{B}$. The function who described the change of position is called **motion** and it is defined as

$$\boldsymbol{\varphi} : \mathcal{B}_0 \times I \to \mathbb{R}^3 \tag{2.1}$$

where $I \subseteq \mathbb{R}^+$. Then, we can track the position of any material point. In fact at any time, thank to the motion function we will know that the material point $\boldsymbol{X}$ at time t will occupy position $\boldsymbol{x}$

$$\boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{X}, t) = \boldsymbol{x}(\boldsymbol{X}, t). \tag{2.2}$$

The motion must satisfy some regularity conditions in order to be invertible. In other words function $\boldsymbol{\varphi}$ has to be a bijective function. This is due because we are modeling in without taking into account any fracture or compenetration. With these conditions the inverse motion will be unique and it will allows as to know which was the initial configuration of any material point of the body:

$$\boldsymbol{X} = \varphi^{-1}(\boldsymbol{x}, t) = \boldsymbol{X}(\boldsymbol{x}, t). \tag{2.3}$$

The inverse motion is:

$$\boldsymbol{\varphi}^{-1} : \mathbb{R}^3 \times I \to \mathcal{B}_0. \tag{2.4}$$

Thanks to the motion function we can compute the **displacement field**. This field is given by the difference between the current position and the original one. In both description we have:

$$\boldsymbol{U}(\boldsymbol{X}, t) = \boldsymbol{x}(\boldsymbol{X}, t) - \boldsymbol{X} \tag{2.5}$$

$$\boldsymbol{u}(\boldsymbol{x}, t) = \boldsymbol{x} - \boldsymbol{X}(\boldsymbol{x}, t). \tag{2.6}$$

Fig[2.1] summarizes what has been explained so far.

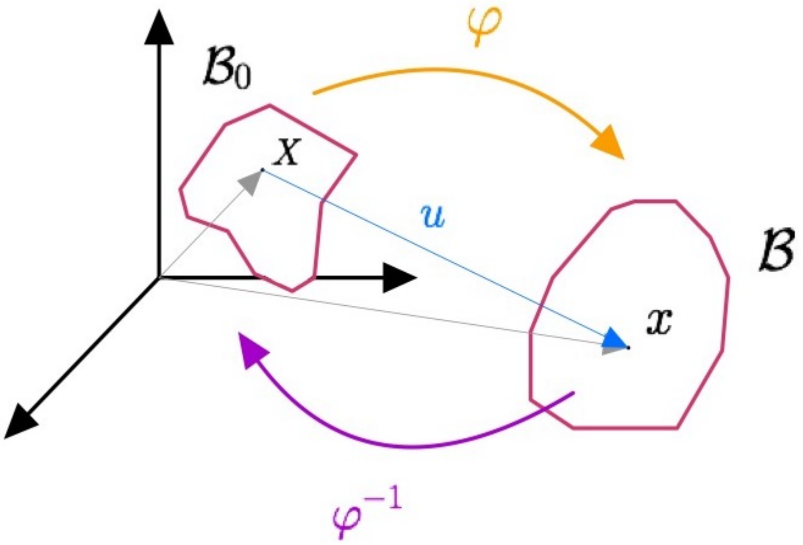

Figure 2.1: Schematization of the motion.

Given the motion of a material point we can compute its **velocity** and **acceleration**. They are defined as the first and second time derivative of the motion respectively:

$$\boldsymbol{V}(\boldsymbol{X},t) = \frac{\partial \boldsymbol{\varphi}(\boldsymbol{X},t)}{\partial t} = \boldsymbol{V}[\boldsymbol{\varphi}^{-1}(\boldsymbol{x},t)] = \boldsymbol{v}(\boldsymbol{x},t) \tag{2.7}$$

$$\boldsymbol{A}(\boldsymbol{X},t) = \frac{\partial^2 \boldsymbol{\varphi}(\boldsymbol{X},t)}{\partial t^2} = \boldsymbol{A}[\boldsymbol{\varphi}^{-1}(\boldsymbol{x},t)] = \boldsymbol{a}(\boldsymbol{x},t). \tag{2.8}$$

2.1.3 Deformation Gradient

An other fundamental quantity in continuum mechanics is the **deformation gradient**. It is defined as the following second order tensor:

$$\boldsymbol{F} = \frac{\partial \boldsymbol{\varphi}(\boldsymbol{X},t)}{\partial \boldsymbol{X}} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} \quad F_{i,I} = \frac{x_i}{X_I}. \tag{2.9}$$

that transforms linear spatial element in linear material element:

$$d\boldsymbol{x} = \boldsymbol{F}\, d\boldsymbol{X}. \tag{2.10}$$

This tensor is a *two-point tensor* because it involves coordinates from two different configuration (see index form of Equation[2.9]). It tells us how the material point $\boldsymbol{X}$ deforms during the motion.

Its determinant is indicate with J and gives information about the volume if the material point changes during the motion. In particular we have:

$$J = det(\boldsymbol{F}) = \frac{dv}{dV} \tag{2.11}$$

where dv is the elemental volume in current configuration and dV in initial configuration.

The deformation gradient is also involved in the **polar decomposition theorem**. The statement is:

> $\exists\ !\ \boldsymbol{R}$ orthogonal and $\exists\ !\ \boldsymbol{U}, \boldsymbol{V}$ symmetric and positive definite such that
>
> $$\begin{cases} \boldsymbol{F} = \boldsymbol{RU} \\ \\ \boldsymbol{F} = \boldsymbol{VR} \end{cases} \tag{2.12}$$

The orthogonal matrix $\boldsymbol{R}$ describes the *rigid deformation* while $\boldsymbol{U}$ and $\boldsymbol{V}$ describe the *effective deformation* (see FIG[2.2]). The first expression of Equation[2.12] is known as *right polar decomposition* while the second is called *left polar decomposition*.

2.1.4 Strain Tensors

Strain tensor are a useful tool for analyzing how the material point deforms during the motion. These tensors can be expressed in either initial (material) or deformed (spatial) configuration. Here, we list the main ones.

Material Strain Tensors

First, we have to state the **root square theorem**:

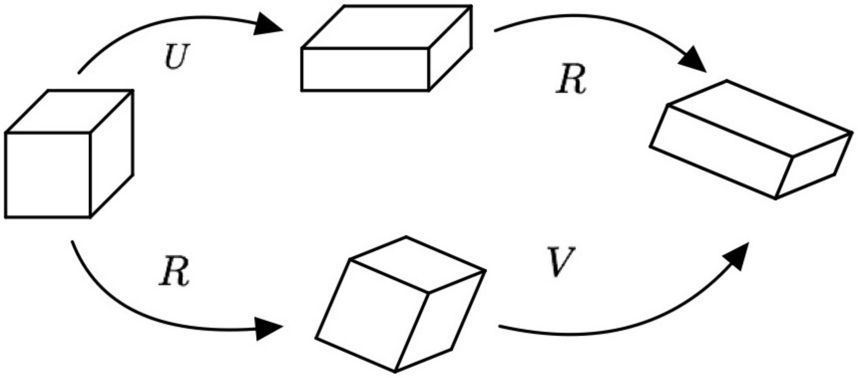

Figure 2.2: Scheme of polar decomposition for a material point.

$\forall \boldsymbol{U}$ symmetric and positive definite $\exists ! \boldsymbol{C}$ s.t. $\boldsymbol{C} = \boldsymbol{U}^2$.

The tensor $\boldsymbol{C}$ is known as *right Cauchy-Green tensor*. Using this definition and applying the right polar decomposition we end up with:

$$\boldsymbol{C} = \boldsymbol{F}^T \boldsymbol{F} \quad C_{I,J} = F_{I,i} F_{i,J}. \tag{2.13}$$

The diagonal of $\boldsymbol{C}$ represents the elongation or contraction of the material point in direction of the basis of the reference system, while the extra-diagonal therms are correlated to angular variation. The inverse of $\boldsymbol{C}$, $\boldsymbol{B} = \boldsymbol{C}^{-1}$, is called *Piola deformation tensor*.

Another very used material tensor is the so called *Green-Lagrange strain tensor*. It is define as

$$\boldsymbol{E} = \frac{1}{2}(\boldsymbol{C} - \boldsymbol{I}). \tag{2.14}$$

Spatial Strain Tensors

Using again the root square theorem and the polar decomposition theorem applied to the left decomposition we obtain a spatial tensor called *left Cauchy-Green tensor* given by:

$$\boldsymbol{b} = \boldsymbol{F} \boldsymbol{F}^T \quad b_{i,j} = F_{i,I} F_{I,j} \tag{2.15}$$

From this tensor we can obtain the well-known *Euler-Almansi strain tensor*:

$$\boldsymbol{e} = \frac{1}{2}(\boldsymbol{I} - \boldsymbol{b}^{-1}). \tag{2.16}$$

2.1.5 Rate of deformation Tensors

Sometimes it is useful to know how tensor fields change in time. We start by introducing the *spatial and material velocity gradient.* They are defined respectively as:

$$\boldsymbol{l}(\boldsymbol{x}, t) = \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}} \quad l_{i,j} = \frac{\partial v_i}{\partial x_j} \tag{2.17}$$

$$\dot{\boldsymbol{F}} = \frac{\partial}{\partial t}\left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}}\right) \quad \dot{F}_{i,I} = \frac{\partial v_i}{\partial X_I}. \tag{2.18}$$

These two tensors are connected with each other by following useful relation $\dot{\boldsymbol{F}} = \boldsymbol{l}\boldsymbol{F}$.
The spatial velocity gradient $\boldsymbol{l}$ can be additively decompose as

$$\boldsymbol{l}(\boldsymbol{x}, t) = \boldsymbol{d}(\boldsymbol{x}, t) + \boldsymbol{w}(\boldsymbol{x}, t) \tag{2.19}$$

where $\boldsymbol{d}$ is a symmetric tensor called *rate of strain tensor* and it is given by

$$\boldsymbol{d} = \frac{1}{2}(\boldsymbol{l} + \boldsymbol{l}^T) \tag{2.20}$$

and $\boldsymbol{w}$ is a antisymmetric tensor called *rate of rotation tensor* and it is defined as

$$\boldsymbol{w} = \frac{1}{2}(\boldsymbol{l} - \boldsymbol{l}^T). \tag{2.21}$$

2.2 Stress

One of the fundamental concepts in continuum mechanics is **stress**. The motion and various deformations that the continuum undergoes cause to the material points that compose it to interact with each other. One of the consequences of this interaction is the emergence of stress. Stresses have the dimension of a force per unit area. It is therefore postulated the existence of this stress acting on the material point. This stress acting on the material point, A, is the reaction to the tension that another material point, B, in contact with A, exerts. We report here the main concepts and definitions that will come in handy in the next chapters.

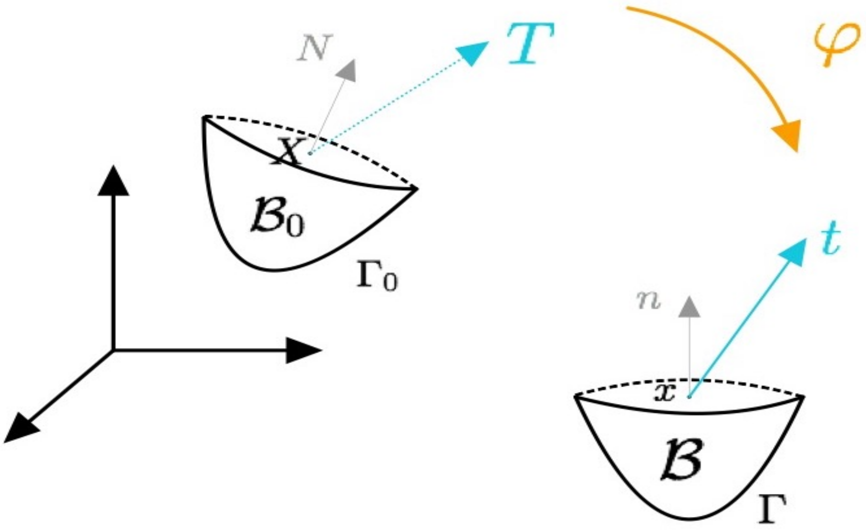

Figure 2.3: Stress representation in initial and deformed configuration.

Let's suppose we have a body in which external forces act to induce motion. Imagine to cut the body with a plane and focus the attention on a material point. In that material point a stress will be exerted FIG[2.3]. Then, the **Cauchy's Postulate** states the existence of this stress and for every surface element ds in the current configuration we have:

$$\boldsymbol{t}(\boldsymbol{x}, \boldsymbol{n}, t) = \lim_{ds \to 0} \frac{d\boldsymbol{f}}{ds} \tag{2.22}$$

where $d\boldsymbol{f}$ is any elementary force acting on a portion of the body. From this postulate we can also state the **Cauchy'stress theorem** (for the proof we refer to Cardin et al. [12]):

$$\boxed{\exists!\ \boldsymbol{\sigma},\boldsymbol{P} \in \mathbb{R}^{3\times3} \text{ s.t. } \boldsymbol{t}(\boldsymbol{x}, \boldsymbol{n}, t) = \boldsymbol{\sigma}(\boldsymbol{x}, t)\boldsymbol{n} \text{ or } \boldsymbol{T}(\boldsymbol{X}, \boldsymbol{N}, t) = \boldsymbol{P}(\boldsymbol{X}, t)\boldsymbol{N}}$$

where

- $\boldsymbol{\sigma}$ is the *Cauchy stress* or *true stress*
- $\boldsymbol{P}$ is called *first Piola-Kirchhoff stress tensor* (1PK)
- $\boldsymbol{n}$ is the normal vector to the cutting plane in current configuration (see FIG[2.3])
- $\boldsymbol{N}$ is the normal vector to the cutting plane in initial configuration
- $\boldsymbol{t}$ is the stress in the current configuration
- $\boldsymbol{T}$ is the stress in the initial configuration [1]

For the proof, please refer to [12]. The Cauchy tensor and the 1PK tensor are connected with the following formula

$$\boldsymbol{P} = J\boldsymbol{\sigma}\boldsymbol{F}^{\boldsymbol{-T}}. \tag{2.23}$$

Other alternative stress tensors can be defined. for example we could have

- Kirchhoff stress tensor $\boldsymbol{\tau} = J\boldsymbol{\sigma}$
- second Piola-Kirchhoff stress tensor (2PK) $\boldsymbol{P} = \boldsymbol{F}\boldsymbol{S}$

The 2PK tensor is particularly useful in computational mechanics due to its symmetry and it is used is various material formulation. In fact, the 1PK is not symmetric and this is very inconvenient in many situations.

2.3 Balance Principles

In this section we are going to define the most important balance principles such as the principle of conservation of mass, the principle of conservation of linear momentum, and the principle of conservation of angular momentum. The reader is encouraged to consult works such as [6],[12],[23] for a deeper understanding about the topic.

[1] We should specify that the vector $\boldsymbol{T}$ and the tensor $\boldsymbol{P}$ are somewhat fictitious and they are a mathematical artifice that often comes in handy for measuring stress. This is because in the initial configuration there are no forces acting on the body and consequently there is no stress inside the body.

2.3.1 Mass Balance Principle

Under the assumptions of regular motion, the body cannot undergo fracture phenomena and/or mass loss. Introducing the concept of mass density as the amount of mass per unit volume i.e.

$$\rho : \mathbb{R}^3 \times I \to \mathbb{R}^+ \quad \rho(\boldsymbol{x}, t) = \frac{dm}{dv}, \tag{2.24}$$

it follows therefore that the total mass of the body must be conserved:

$$\int_{\Omega_0} \rho_0(\boldsymbol{X}) \; dV = \int_{\Omega} \rho(\boldsymbol{x}, t) \; dv \tag{2.25}$$

or likewise

$$\frac{d}{dt} \int_{\Omega} \rho(\boldsymbol{x}, t) \; dv = 0. \tag{2.26}$$

where $\rho(\boldsymbol{X})$ is the initial mass density. At this point we apply Reynold's transport theorem (see for instance [23]) and we obtain

$$\begin{gathered} \frac{d}{dt} \int_{\Omega} \rho(\boldsymbol{x}, t) \; dv = \int_{\Omega} \left(\frac{d\rho}{dt} + \rho \mathrm{div}\, \boldsymbol{v} \right) \; dv = \\ \int_{\Omega} \left(\frac{\partial \rho}{\partial t} + \mathrm{div}(\rho \boldsymbol{v}) \right) \; dv = 0 \end{gathered} \tag{2.27}$$

which in its Eulerian[2] differential form becomes

$$\frac{d\rho}{dt} + \rho \mathrm{div} \boldsymbol{v} = 0 \qquad \frac{\partial \rho}{\partial t} + \mathrm{div}(\rho \boldsymbol{v}) = 0 \tag{2.28}$$

while in Lagrangian form is

$$\rho_0(\boldsymbol{X}) = \rho(\boldsymbol{x}, t) \; J. \tag{2.29}$$

For the sake of clarity of exposition we specify that $\boldsymbol{v}$ is the Eulerian velocity defined in Equation[2.7] while dv in the volume integral measure.

[2] Equations[2.28] are known as **Continuity Equations**.

2.3.2 Linear Momentum Balance Principle

Let us consider a continuum body $\mathcal{B}$ in the current configuration and a subset $\Omega \subseteq \mathcal{B}$ which is subjects to surface traction forces $\boldsymbol{t} : \Gamma \times I \to \mathbb{R}^3$ and volume forces $\boldsymbol{b} : \Omega \times I \to \mathbb{R}^3$. The force equilibrium read:

$$\int_\Omega \rho(\boldsymbol{x},t)\boldsymbol{b}(\boldsymbol{x},t)\ dv + \int_\Gamma \boldsymbol{t}(\boldsymbol{x},\boldsymbol{n},t)\ da = \int_\Omega \rho(\boldsymbol{x},t)\boldsymbol{a}(\boldsymbol{x},t)\ dv. \tag{2.30}$$

Next, we introduce the Cauchy's theorem and we evoke the divergence theorem. In this way Equation[2.30] becomes:

$$\int_\Omega \left(\rho(\boldsymbol{x},t)\boldsymbol{b}(\boldsymbol{x},t) + \mathrm{div}\boldsymbol{\sigma}(\boldsymbol{x},t) - \rho(\boldsymbol{x},t)\boldsymbol{a}(\boldsymbol{x},t)\right)\ dv = \boldsymbol{0} \tag{2.31}$$

which is known as **Cauchy's first equations of motion**. These set of equations can be written in their local spatial and material forms respectively as

$$\rho(\boldsymbol{x},t)\boldsymbol{b}(\boldsymbol{x},t) + \mathrm{div}\ \boldsymbol{\sigma}(\boldsymbol{x},t) = \rho(\boldsymbol{x},t)\boldsymbol{a}(\boldsymbol{x},t) \tag{2.32}$$

and

$$\rho_0(\boldsymbol{X})\boldsymbol{B}(\boldsymbol{X},\boldsymbol{t}) + \mathrm{DIV}\boldsymbol{P}(\boldsymbol{X},t) = \rho_0(\boldsymbol{X})\boldsymbol{A}(\boldsymbol{X},t). \tag{2.33}$$

These two equation are extremely important since they are basically the starting point for any solid and fluid numerical simulation.

2.3.3 Angular Momentum Balance Principle

The angular momentum balance is obtained by vector-multiplying Equation[2.30] by an arbitrary vector $\boldsymbol{r}$

$$\int_\Omega \boldsymbol{r} \times \rho(\boldsymbol{x},t)\boldsymbol{b}(\boldsymbol{x},t)\ dv + \int_\Gamma \boldsymbol{r} \times \boldsymbol{t}(\boldsymbol{x},\boldsymbol{n},t)\ da = \int_\Omega \boldsymbol{r} \times \rho(\boldsymbol{x},t)\boldsymbol{a}(\boldsymbol{x},t)\ dv. \tag{2.34}$$

Using Equation[2.32] and introducing the third-order permutation tensor $\mathcal{E}$, we have:

$$\int_\Omega \boldsymbol{r} \times \left(\rho(\boldsymbol{x},t)\boldsymbol{b}(\boldsymbol{x},t) + \mathrm{div}\boldsymbol{\sigma}(\boldsymbol{x},t) - \rho(\boldsymbol{x},t)\boldsymbol{a}(\boldsymbol{x},t)\right)\ dv = \int_\Omega \mathcal{E} : \boldsymbol{\sigma}^T\ dv \tag{2.35}$$

and using again Equation[2.32], we get the so called **Cauchy's second equation of motion**:

$$\mathcal{E} : \boldsymbol{\sigma}^T = \boldsymbol{0} \tag{2.36}$$

Equation[2.36] implies that the Cauchy tensor has to be symmetric $\boldsymbol{\sigma} = \boldsymbol{\sigma}^T$. In material coordinates we have

$$\boldsymbol{F}\boldsymbol{P} = \boldsymbol{P}^T\boldsymbol{F}^T \text{ or } \boldsymbol{S} = \boldsymbol{S}^T. \tag{2.37}$$

2.3.4 Energy Balance

The energy balance is expressed through the first law of thermodynamics of continuum body. First, we have to introduce some new quantities

- **source of heat**: $r : \Omega \times I \to \mathbb{R}$ or $R : \Omega_0 \times I \to \mathbb{R}$
- **internal energy** per unit mass: $e_c : \Omega \times I \to \mathbb{R}$ or $e : \Omega_0 \times I \to \mathbb{R}$
- **heat flux**: $\boldsymbol{q} : \Omega \times I \to \mathbb{R}^3$ or $\boldsymbol{Q} : \Omega_0 \times I \to \mathbb{R}^3$

The heat flux is needed to state the Stoke's heat flux theorem which is the counterpart of the Cauchy's theorem in thermodynamics:

$$q(\boldsymbol{x}, \boldsymbol{n}, t) = -\boldsymbol{q}(\boldsymbol{x}, t)\boldsymbol{n} \tag{2.38}$$

$$Q(\boldsymbol{X}, \boldsymbol{N}, t) = -\boldsymbol{Q}(\boldsymbol{X}, t)\boldsymbol{N}. \tag{2.39}$$

With all these concepts we can state the first principle of the continuum thermodynamics which is

$$\mathcal{P}^{int}(t) + \mathcal{Q}(t) = \dot{\mathcal{E}}(t) \tag{2.40}$$

where:

$$\mathcal{P}^{int}(t) = \int_{\Omega} \boldsymbol{\sigma} : \boldsymbol{d} \; dv = \int_{\Omega_0} \boldsymbol{P} : \dot{\boldsymbol{F}} \; dV$$

$$\mathcal{E}(t) = \int_{\Omega} e_c(\boldsymbol{x}, t) \; dv = \int_{\Omega_0} e(\boldsymbol{X}, t) \; dV$$

and

$$\mathcal{Q}(t) = \int_{\Gamma} q(\boldsymbol{x}, \boldsymbol{n}, t) \; da + \int_{\Omega} r(\boldsymbol{x}, t) \; dv =$$

$$= \int_{\Gamma} Q(\boldsymbol{X}, \boldsymbol{N}, t) \; dA + \int_{\Omega_0} R(\boldsymbol{X}, t) \; dV.$$

The first principle can be explicitly written as:

$$\int_\Omega \dot{e}_c \, dv = \int_\Omega \boldsymbol{\sigma} : \boldsymbol{d} - \mathrm{div}\boldsymbol{q} + r \, dv \tag{2.41}$$

which in local form spatial and material is respectively

$$\dot{e}_c = \boldsymbol{\sigma} : \boldsymbol{d} - \mathrm{div}\boldsymbol{q} + r \tag{2.42}$$

and

$$\dot{e} = \boldsymbol{P} : \dot{\boldsymbol{F}} - \mathrm{DIV}\boldsymbol{Q} + R. \tag{2.43}$$

2.3.5 Summary

Eulerian

$\frac{d}{dt}\rho + \rho \mathrm{div}\boldsymbol{v} = 0$	Mass Balance
$\rho\boldsymbol{b} + \mathrm{div}\ \boldsymbol{\sigma} = \rho\boldsymbol{a}$	Linear Momentum Balance
$\mathcal{E} : \boldsymbol{\sigma}^T = 0$	Angular Momentum Balance
$\dot{e}_c = \boldsymbol{\sigma} : \boldsymbol{d} - \mathrm{div}\boldsymbol{q} + r$	Energy Balance

Lagrangian

$\rho_0(\boldsymbol{X}) = \rho(\boldsymbol{x}, t)\ J$	Mass Balance
$\rho_0\boldsymbol{B} + \mathrm{DIV}\boldsymbol{P} = \rho_0\boldsymbol{A}$	Linear Momentum Balance
$\boldsymbol{F}\boldsymbol{P} = \boldsymbol{P}^T\boldsymbol{F}^T$	Angular Momentum Balance
$\dot{e} = \boldsymbol{P} : \dot{\boldsymbol{F}} - \mathrm{DIV}\boldsymbol{Q} + R.$	Energy Balance

2.4 Constitutive Laws

The constitutive laws are mathematical relationships that allow us to describe the mechanical behavior of the material based on the forces acting on it. The mechanical state of the material points that make up the continuum also depends on the mechanical "history". By mechanical history we mean the deformation history of the body, i.e. all those deformations occurred in the continuum. The first mathematical tool useful for the deformational description of the continuum is the deformation gradient ([2.9]). However, this does not prove to be sufficient for this purpose. What we need are the so-called **internal variables**, variables that describe any dissipative phenomena that are developed in the processes to which the material points of the continuum go against (examples are damage, permanent deformation, etc.). The internal variables generally can be considered as tensors of generic order are functions of both the position of the material point, $\boldsymbol{X}$, and of time and they are indicated as

$$\dot{\alpha}^{j}(\boldsymbol{X}, t) \text{ where } j = 1, 2, 3, ..., M$$

with M is the number of dissipative phenomena acting on the body. One of the fundamental tools for the formulation of constitutive laws is the **second principle of thermodynamics**. In the context of continuum mechanics, the second principle of thermodynamics is expressed through the **Clausius-Duhem inequality**:

$$\dot{\Psi}(\boldsymbol{X}, t) - \boldsymbol{P}(\boldsymbol{X}, t) : \dot{\boldsymbol{F}}(\boldsymbol{X}, t) \leq 0, \tag{2.44}$$

where Ψ is the **Helmholtz free energy** and it is defined as

$$\dot{\Psi}(\boldsymbol{X}, t) = \theta\, l(\boldsymbol{X}, t) \tag{2.45}$$

and $l(\boldsymbol{X}, t)$ is the work expended to deform the body while θ is a parameter that can take values between zero and one, $\theta \in [0, 1]$. When theta is one, $\theta = 1$, we will have that all the work spent to deform the body is accumulated in reversible form. This is the typical behavior of elastic and hyperelastic materials. When theta is less than one, $\theta < 1$ the material is dissipating energy. This means that the work spent to deform the material is dissipated both by inducing irreversible alterations on the microstructure and by internal friction.

This inequality tells us that the power developed by deformational forces can at most be completely accumulated in reversible form by the material. In other words, we can say that the energy accumulated by the material can never be greater than the work done to deform it.
In this thesis we will focus exclusively on the case where theta equals one. The engineering applications that will be seen will be inherent to elastic and hyperelastic materials.

2.4.1 Hyperelastic Material

Hyperelastic materials, as well as elastic materials, are nondissipative materials. These types of materials are characterized by reversible and time-independent deformation mechanisms. This implies that the Clausius-Duhem inequality can be rewritten as the following equality:

$$\frac{\partial \Psi}{\partial \dot{\alpha}^j} = 0. \tag{2.46}$$

This equality suggests that the Helmholtz free energy function depends only on the deformational state (described for example by deformation tensor $\boldsymbol{C}$) and not on dissipative phenomena:

$$\Psi = \Psi(\boldsymbol{C}). \tag{2.47}$$

We are used to denote free energy as **strain energy density** denoted by $W(\boldsymbol{C})$, $\Psi = \Psi(\boldsymbol{C}) = W(\boldsymbol{C})$. Thus, defining a constitutive law for a hyperelastic material means defining an expression of W that is a function solely of the deformational state $\boldsymbol{C}$.
In the context of hyperelastic materials, the elastic constitutive tensor is usually defined. This tensor (in its material description) is defined using the second Piola-Kirchhoff tensor and the right Cauchy-Green tensor:

$$\mathbb{C} = 2\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{C}} = 4\frac{\partial^2 \Psi}{\partial \boldsymbol{C} \partial \boldsymbol{C}} \quad \mathbb{C}_{ijkl} = 2\frac{\partial S_{ij}}{\partial C_{kl}} = 4\frac{\partial^2 \Psi}{\partial C_{ij} \partial C_{kl}} \tag{2.48}$$

Let us now focus our attention on a particular constitutive model that will be used in the chapter on engineering applications. The model in question is the **neo-Hookian** model. This type of constitutive law characterizes hyperelastic, isotropic and incompressible materials. For an isotropic material the work spent by the forces to deform the body does not depend on the direction of the deformation process but only on the intensity of it. Consequently, W will depend on $\boldsymbol{C}$ through its intensity (described by the eigenvalues, λ_i) that for the spectral and square root theorem is written as

$$\boldsymbol{C} = \lambda_i^2 \, \boldsymbol{v}_i \otimes \boldsymbol{v}_i \tag{2.49}$$

where $\boldsymbol{v}_i$ are the eigenvectors. The eigenvalues represent the principal stretches and the energy density is a function of them:

$$W(\boldsymbol{C}) = W(\lambda_1, \lambda_2, \lambda_3) \tag{2.50}$$

which it can be also expressed via the so called **principal scalar invariants**:

$$W(\boldsymbol{C}) = W(\lambda_1, \lambda_2, \lambda_3) = W(I_1, I_2, I_3) \tag{2.51}$$

where $I_1 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2$, $I_2 = \lambda_1^2\lambda_2^2 + \lambda_1^2\lambda_3^2 + \lambda_2^2\lambda_3^2$ and $I_3 = \lambda_1^2\lambda_2^2\lambda_3^2$. Thus, one way to write the energy density for neo-Hookian material is:

$$W(\boldsymbol{C}) = C_1(I_1 - 3) - p(J - 1) \tag{2.52}$$

where C_1 is en experimental parameter, I_1 is the first invariant of $\boldsymbol{C}$, J is the determinant of the deformation gradient[3] and p is the hydrostatic pressure[4]. From Equation[2.52] we can find the stress expressions. After few steps we have:

$$\boldsymbol{S} = 2C_1\boldsymbol{I} - pJ\boldsymbol{C}^T \tag{2.53}$$

or

$$\boldsymbol{\sigma} = 2C_1\boldsymbol{b} - pJ \tag{2.54}$$

where $\boldsymbol{b}$ is defined by Equation[2.15].

2.4.2 Elastic Material

Linear elasticity is a special case of hyperelasticity. In the case of linear elasticity (linear by geometry) the displacements and deformations are small, i.e. we can confuse the deformed configuration with the undeformed ones $\Omega \simeq \Omega_0$. As an immediate consequence of this we have that the mechanical

[3] $J = 1$ for incompressible material

[4] The hydrostatic pressure plays the role of a Lagrange Multiplier. This means that if J is different from 1, it take value $+\infty$ Physically, we can interprete this as an infinite amount of work to change the volume of the material.

problem can be defined on a known domain since the undeformed configuration turns out to be known. Another consequence lies in the fact that real stress, $\boldsymbol{\sigma}$, and nominal stress, $\boldsymbol{P}$, coincide. Again this is due to the fact that the infinitesimal area elements in the two configurations are the same.
The deformation measure that best fits the mechanical problem of small deformations is given by the engineering tensor, $\boldsymbol{\epsilon}$, which coincides with the symmetrical part of the displacement gradient

$$\boldsymbol{\epsilon} = \frac{1}{2}(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T). \tag{2.55}$$

Even in the case of linear elasticity, it is possible to define a strain energy density function that will be a function of the tensor ϵ only. Similarly to what we have seen for the case of hyperleasticity we define the elastic constitutive tensor as

$$\mathbb{C} = \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\epsilon}} = \frac{\partial^2 W}{\partial \boldsymbol{\epsilon} \partial \boldsymbol{\epsilon}} \quad \mathbb{C}_{ijkl} = \frac{\partial \sigma_{ij}}{\partial \epsilon_{kl}} = \frac{\partial^2 W}{\partial \epsilon_{ij} \partial \epsilon_{kl}}. \tag{2.56}$$

In the context of computational mechanics, tensor notation is inconvenient for implementation purposes. Therefore, very often the so called Voigt notation is adopted. With this notation it is possible to express higher order tensors by means of vectors or second order tensors. In the case of linear elasticity we will have that the tensor $\mathbb{C}$ can be written as a matrix, $\boldsymbol{D}$, often called a stiffness matrix. This matrix connect the stress tensor, $\boldsymbol{\sigma}$, with small strain tensor, $\boldsymbol{\epsilon}$, via the following expression:

$$\boldsymbol{\sigma} = \boldsymbol{D}\boldsymbol{\epsilon}. \tag{2.57}$$

The components of the $\boldsymbol{D}$ tensor are well known for an isotropic material. See for example [4].

Chapter 3

Linearized Updated Lagrangian Formulation

Continuum mechanics is mostly characterized by nonlinearity due to geometrical and material properties. Geometrical nonlinearity occurs when variations of displacement or strain affect the response of loads in a nonlinear way. Then, the dependence between displacements and loads is a nonlinear relation. Material nonlinearity occurs when the stress-strain relation, meaning the constitutive law, is nonlinear. In order to develop the finite element scheme we have to linearize the equation of virtual works that are the weak formulation of the Cauchy equations. This is achieved by performing a Taylor's series expansion that will naturally lead to the Newton-Raphson iterative process which is one of the most used tools to solve nonlinear equations. Therefore, we will introduce the concept of linearization and then the directional derivative. In this way, we will be ready to linearize the PVW and once done, we will discretize it with the FEM formulation. At that point, since we will consider the dynamics equilibrium, we will have to use some time integrators such as the Newmark time integration scheme. Following this procedure we will obtain a linear system whose solution will be in terms of displacement or acceleration increment. These increments will be used interactively to update the position and acceleration until the dynamical equilibrium will be satisfied. Some great references for the topic are provided by Bonet et al. [6],[5].

3.1 Linearization and Directional Derivative

First of all we have to introduce the linearization procedure and the directional derivative concept. Let us consider the following set of nonlinear equations

$$\boldsymbol{F}(\boldsymbol{x}) = \mathbf{0}, \tag{3.1}$$

where $\boldsymbol{F}$ could represent a system of both nonlinear algebraic or differential equations and $\boldsymbol{x}$ could represent scalar or vector unknown function or simply a scalar variable. This notation is adopted in order to keep general the discussion and then eventually it will be specified the particular quantity involved.

Consider now an initial guess, $\boldsymbol{x}_0$, and an increment, $\boldsymbol{u}$, in such way that $\bar{\boldsymbol{x}} = \boldsymbol{x}_0 + \boldsymbol{u}$ is close to the solution, $\mathbf{x}$, of Equation[3.1]. In other words if $\bar{\boldsymbol{x}}$ tends to $\boldsymbol{x}$ then Equation[3.1] is satisfied, meaning

$$\lim_{\bar{\boldsymbol{x}} \to \boldsymbol{x}} \boldsymbol{F}(\bar{\boldsymbol{x}}) = \mathbf{0}. \tag{3.2}$$

As introduced previously we are going to use the Taylor's series expansion to linearize nonlinear equations. Our goal is to linearize $\boldsymbol{F}(\bar{\boldsymbol{x}})$ and this will lead to some derivatives of $\boldsymbol{F}$ with respect to $\bar{\boldsymbol{x}}$. These derivatives may not be easy to compute. An artificial scalar parameter, ϵ, is introduced to simplify them. In this way we could define a new function $\boldsymbol{G}$ given by

$$\boldsymbol{G}(\epsilon) = \boldsymbol{F}(\bar{\boldsymbol{x}}) = \boldsymbol{F}(\boldsymbol{x}_0 + \epsilon \boldsymbol{u}), \tag{3.3}$$

which allows us to perform the Taylor's series expansion in a simpler way. Therefore, if we expand Equation[3.3] around $\epsilon = 0$ we have what follows

$$\boldsymbol{G}(\epsilon) = \boldsymbol{G}(0) + \left.\frac{d\boldsymbol{G}(\epsilon)}{d\epsilon}\right|_{\epsilon=0} \epsilon + \frac{1}{2} \left.\frac{d^2\boldsymbol{G}(\epsilon)}{d\epsilon^2}\right|_{\epsilon=0} \epsilon^2 + o(\epsilon^2), \tag{3.4}$$

which, by making use of $\boldsymbol{G}$ definition in Equation[3.3], can be written also as

$$\boldsymbol{F}(\mathrm{x}_0 + \epsilon \boldsymbol{u}) = \boldsymbol{F}(\boldsymbol{x}_0) + \left.\frac{d\boldsymbol{F}(\boldsymbol{x}_0 + \epsilon \boldsymbol{u})}{d\epsilon}\right|_{\epsilon=0} \epsilon + \frac{1}{2} \left.\frac{d^2\boldsymbol{F}(\boldsymbol{x}_0 + \epsilon \boldsymbol{u})}{d\epsilon^2}\right|_{\epsilon=0} \epsilon^2 + o(\epsilon^2). \tag{3.5}$$

At this point if we neglect the higher order terms and set $\epsilon = 1$ Equation[3.5] gives

$$\boldsymbol{F}(\mathbf{x}_0 + \epsilon \boldsymbol{u}) \simeq \boldsymbol{F}(\mathbf{x}_0) + \left.\frac{d\boldsymbol{F}(\boldsymbol{x}_0 + \epsilon \boldsymbol{u})}{d\epsilon}\right|_{\epsilon=0}. \tag{3.6}$$

The term

$$\left.\frac{d\boldsymbol{F}(\mathbf{x}_0 + \epsilon \boldsymbol{u})}{d\epsilon}\right|_{\epsilon=0} := D\boldsymbol{F}(\boldsymbol{x}_0)[\boldsymbol{u}] \tag{3.7}$$

is the *directional derivative* of $\boldsymbol{F}(\bar{\boldsymbol{x}})$ at $\boldsymbol{x}_0$ in the direction of $\boldsymbol{u}$. Supposing that $\boldsymbol{F}(\bar{\boldsymbol{x}}) = 0$, Equation[3.6] becomes

$$\boldsymbol{F}(\boldsymbol{x}_0) + D\boldsymbol{F}(\boldsymbol{x}_0)[\boldsymbol{u}] = \mathbf{0}. \tag{3.8}$$

Now, we can start the Newton-Raphson procedure. In fact, Equation[3.8] can be written for a generic iteration step, k, as

$$\boxed{D\boldsymbol{F}(\boldsymbol{x}^{(k)})[\boldsymbol{u}] = -\boldsymbol{F}(\boldsymbol{x}^{(k)}); \qquad \boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{u}. \qquad (3.9)}$$

Equations[3.9] tell us that once we solve $D\boldsymbol{F}(\boldsymbol{x}^{(k)})[\boldsymbol{u}] = -\boldsymbol{F}(\boldsymbol{x}^{(k)})$ for $\boldsymbol{u}$, we update the solution at step $(k+1)$ as $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{u}$ and we repeat it iterativelly until convergence is achived.

We conclude this section by listing some useful properties of directional derivatives that we will use later on (see [6]). They are:

- Linearity $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{F}_1(\boldsymbol{x}) + \boldsymbol{F}_2(\boldsymbol{x})$

$$D\boldsymbol{F}(\boldsymbol{x}_0)[\boldsymbol{u}] = D\boldsymbol{F}_1(\boldsymbol{x}_0)[\boldsymbol{u}] + D\boldsymbol{F}_2(\boldsymbol{x}_0)[\boldsymbol{u}] \tag{3.10}$$

- Product Rule $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{F}_1(\boldsymbol{x}) \cdot \boldsymbol{F}_2(\mathbf{x})$

$$D\boldsymbol{F}(\boldsymbol{x}_0)[\boldsymbol{u}] = D\boldsymbol{F}_1(\boldsymbol{x}_0)[\boldsymbol{u}] \cdot \boldsymbol{F}_2(\mathbf{x}_0) + \boldsymbol{F}_1(\boldsymbol{x}_0) \cdot D\boldsymbol{F}_2(\boldsymbol{x}_0)[\boldsymbol{u}] \tag{3.11}$$

- Chain Rule $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{F}_1(\boldsymbol{F}_2(\boldsymbol{x}))$

$$D\boldsymbol{F}(\boldsymbol{x}_0)[\boldsymbol{u}] = D\boldsymbol{F}_1(\boldsymbol{F}_2(\boldsymbol{x}_0))[D\boldsymbol{F}_2(\boldsymbol{x}_0)[\boldsymbol{u}]] \tag{3.12}$$

- Time Derivative $\boldsymbol{F}(\boldsymbol{\phi}(\boldsymbol{X}, t))$

$$\frac{d}{dt}\boldsymbol{F}(\boldsymbol{\phi}(\boldsymbol{X}, t)) = D\boldsymbol{F}[\boldsymbol{v}] \tag{3.13}$$

  where $\boldsymbol{v}$ is the material velocity.

3.2 Linearized Principle of Virtual Works

In this section we are going to linearize the principle of virtual works. The linearization of the various quantities involved can be found in Appendix A. First of all, we recall that the Principle of Virtual Works is obtained by multiplying Equation[2.32] by a test function $\delta\boldsymbol{v}$ and then integrating over the current or initial configuration. Basically, it is the weak form of the Cauchy's Equations mentioned in Equation[2.32]:

$$\delta W(\boldsymbol{\phi}, \delta\boldsymbol{v}) = \int_v \rho\boldsymbol{a}\cdot\delta\boldsymbol{v}\ dv + \int_v \sigma : \delta\boldsymbol{d}\ dv - \int_v \boldsymbol{f}\cdot\delta\boldsymbol{v}\ dv - \int_a \boldsymbol{t}\cdot\delta\boldsymbol{v}\ da = 0, \tag{3.14}$$

while in material form is written as

$$\begin{aligned} \delta W(\boldsymbol{\phi}, \delta\boldsymbol{v}) = & \int_V \rho_0\boldsymbol{A}\cdot\delta\boldsymbol{V}\ dV + \\ + & \int_V \boldsymbol{S} : \delta\dot{\boldsymbol{E}}\ dV - \int_V \boldsymbol{F}_0\cdot\delta\boldsymbol{V}\ dV - \int_A \boldsymbol{T}\cdot\delta\boldsymbol{V}\ dA = 0, \end{aligned} \tag{3.15}$$

where $\delta\boldsymbol{v}$ and $\delta\boldsymbol{V}$ are virtual velocities test function. We could have also used virtual displacement. The space of test function is the space of continuous functions vanishing on the boundary Γ_{v_i} where the velocities are prescribed (see [4]):

$$\delta\boldsymbol{v} \in [U_0]^d \qquad U_0 = \left\{\delta v_i : \delta v_i \in C^0,\ \ \delta v_i = 0 \text{ on } \Gamma_{v_i}\right\}.$$

Alternatively, we could use a more general test space meaning the space of function with square integrable derivatives and vanishing wherever the velocities are prescribed (see [10]):

$$\delta\boldsymbol{v} \in \left[H_0^1\right]^d \qquad H_0^1 = \left\{\delta v_i : \delta v_i \in L^2,\ \frac{\partial\delta v_i}{\partial x_j} \in L^2,\ \delta v_i = 0 \text{ on } \Gamma_{v_i}\right\}.$$

More generally, the PVW is often written in the following compact way

$$\delta W(\boldsymbol{\phi}, \delta\boldsymbol{v}) = \delta W_m(\boldsymbol{\phi}, \delta\boldsymbol{v}) + \delta W_{int}(\boldsymbol{\phi}, \delta\boldsymbol{v}) - \delta W_{ext}(\boldsymbol{\phi}, \delta\boldsymbol{v}) = 0, \tag{3.16}$$

where $\delta W_m(\boldsymbol{\phi}, \delta\boldsymbol{v})$ is the inertial virtual work

$$\delta W_m(\boldsymbol{\phi}, \delta\boldsymbol{v}) = \int_v \rho\boldsymbol{a} \cdot \delta\boldsymbol{v} \; dv = \int_V \rho_0\boldsymbol{a} \cdot \delta\boldsymbol{v} \; dV, \tag{3.17}$$

$\delta W_{int}(\boldsymbol{\phi}, \delta\boldsymbol{v})$ is the internal virtual work

$$\delta W_{int}(\boldsymbol{\phi}, \delta\boldsymbol{v}) = \int_v \sigma : \delta\boldsymbol{d} \; dv = \int_V \boldsymbol{S} : \delta\dot{\boldsymbol{E}} \; dV, \tag{3.18}$$

and $\delta W_{ext}(\boldsymbol{\phi}, \delta\boldsymbol{v})$ is the external virtual work

$$\delta W_{ext}(\boldsymbol{\phi}, \delta\boldsymbol{v}) = \int_v \boldsymbol{f} \cdot \delta\boldsymbol{v} \; dv + \int_a \boldsymbol{t} \cdot \delta\boldsymbol{v} \; da = \int_V \boldsymbol{f}_0 \cdot \delta\boldsymbol{v} \; dV + \int_A \boldsymbol{t}_0 \cdot \delta\boldsymbol{v} \; dA. \tag{3.19}$$

The update Lagrangian formulation (ULF) is characterized by the fact that integrals and various quantities refer to the current configuration. Instead, in the total Lagrangian formulation (TLF) they refer to the initial configuration. Since the linearization in easier to perform in the initial configuration due to the fact the elementary volume dV is constant, we will linearize Equation[3.15] first and then through push-forward operations we will obtain the linearization of Equation[3.14]. This is particularly useful especially when we will linearize the internal virtual work in Equation[3.18]. Regardless which formulation we use, evoking Equation[3.8], the linearized principle of virtual works is in the form

$$\boxed{\delta W(\boldsymbol{\phi}, \delta\boldsymbol{v}) + D\delta W(\boldsymbol{\phi}, \delta\boldsymbol{v})[\boldsymbol{u}] = 0. \qquad (3.20)}$$

Thanks to property [3.10] the directional derivative of the virtual works in Equation[3.16] is

$$D\delta W(\boldsymbol{\phi}, \delta\boldsymbol{v})[\boldsymbol{u}] = D\delta W_m(\boldsymbol{\phi}, \delta\boldsymbol{v})[\boldsymbol{u}] + D\delta W_{int}(\boldsymbol{\phi}, \delta\boldsymbol{v})[\boldsymbol{u}] - D\delta W_{ext}(\boldsymbol{\phi}, \delta\boldsymbol{v})[\boldsymbol{u}]. \tag{3.21}$$

So what we are going to do next is to linearize each terms of Equation[3.21].

3.2.1 Linearization of the Inertial virtual work

The first term we linearize is the inertial virtual works. Following the usual definition of directional derivative given in Equation[3.7], the directional derivative of the inertial virtual works is computed as follows

$$
\begin{aligned}
D\delta W_m(\boldsymbol{\phi}, \delta \boldsymbol{v})[\boldsymbol{u}] &= \int_v D\left(\rho \boldsymbol{a} \cdot \delta \boldsymbol{v}\right)[\boldsymbol{u}]\ dv \\
&= \int_v \rho D\boldsymbol{a}[\boldsymbol{u}] \cdot \delta \boldsymbol{v}\ dv \\
&= \int_v \rho \left(\left. \frac{d}{d\epsilon} \right|_{\epsilon=0} \frac{\partial^2}{\partial t^2}(\boldsymbol{x} + \epsilon \boldsymbol{u}) \right) \cdot \delta \boldsymbol{v}\ dv \\
&= \int_v \rho \left(\left. \frac{d}{d\epsilon} \right|_{\epsilon=0} \frac{\partial^2}{\partial t^2}\boldsymbol{x} + \epsilon \frac{\partial^2}{\partial t^2}\boldsymbol{u} \right) \cdot \delta \boldsymbol{v}\ dv \\
&= \int_v \rho \left(\left. \frac{d}{d\epsilon} \right|_{\epsilon=0} \epsilon \frac{\partial^2}{\partial t^2}\boldsymbol{u} \right) \cdot \delta \boldsymbol{v}\ dv \\
&= \int_v \rho \frac{\partial^2}{\partial t^2}\boldsymbol{u} \cdot \delta \boldsymbol{v}\ dv = \int_v \rho \boldsymbol{a} \cdot \delta \boldsymbol{v}\ dv
\end{aligned}
\tag{3.22}
$$

or alternatively, knowing that $D\boldsymbol{F}(\boldsymbol{x})[\boldsymbol{u}] = \boldsymbol{u} \cdot \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}$ we have

$$
\int_v \rho D\boldsymbol{a}(\boldsymbol{x})[\boldsymbol{u}] \cdot \delta \boldsymbol{v}\ dv = \int_v \rho \boldsymbol{u}^T \cdot \frac{\partial \boldsymbol{a}}{\partial \boldsymbol{x}} \delta \boldsymbol{v}\ dv. \tag{3.23}
$$

3.2.2 Linearization of the Internal Virtual Works

From continuum mechanics we know that $\boldsymbol{S} = \mathbb{C} : \boldsymbol{E}$, where $\mathbb{C}$ is the *material elasticity tensor*. Recalling Equation[3.13] and using the symmetry of $\boldsymbol{S}$ the directional derivatives of the internal virtual works is given by the following steps

$$\begin{aligned}
&D\delta W_{int}(\boldsymbol{\phi}, \delta \boldsymbol{v})[\boldsymbol{u}] = \int_V D\left(\boldsymbol{S} : \delta\dot{\boldsymbol{E}}\right)[\boldsymbol{u}]\ dV \\
&= \int_V D\boldsymbol{S}[\boldsymbol{u}] : \delta\dot{\boldsymbol{E}}\ dV + \int_V \boldsymbol{S} : D\delta\dot{\boldsymbol{E}}[\boldsymbol{u}]\ dV \\
&= \int_V \delta\dot{\boldsymbol{E}} : \mathbb{C} : D\boldsymbol{E}[\boldsymbol{u}]\ dV + \int_V \boldsymbol{S} : \frac{1}{2}\left(\nabla_0 \delta\boldsymbol{v}^T \nabla_0 \boldsymbol{u} + \nabla_0 \boldsymbol{u}^T \nabla_0 \delta\boldsymbol{v}\right)\ dV \\
&= \int_V D\boldsymbol{E}[\delta\boldsymbol{v}] : \mathbb{C} : D\boldsymbol{E}[\boldsymbol{u}]\ dV + \int_V \boldsymbol{S} : \left(\nabla_0 \boldsymbol{u}^T \nabla_0 \delta\boldsymbol{v}\right)\ dV.
\end{aligned} \tag{3.24}$$

Equation[3.24] is the material or Lagrangian linearized internal virtual works. As said before, we have to obtain the spatial or eElerian linearized virtual works to achieve the ULF.
First of all we recall that the second Piola-Kirchhoff tensor can be seen as the pull-back operation applied to the Kirchhoff stress tensor $\boldsymbol{\tau} = J\boldsymbol{\sigma}$ meaning $\boldsymbol{S} = \boldsymbol{\phi}_*^{-1}(\tau) = J\boldsymbol{F}^{-1}\boldsymbol{\sigma}\boldsymbol{F}^{-T}$. Secondly, we evok the property of the double dot product i.e. $\boldsymbol{A} : \boldsymbol{B}\boldsymbol{C} = \boldsymbol{B}^T\boldsymbol{A} : \boldsymbol{C} = \boldsymbol{A}\boldsymbol{C}^T : \boldsymbol{B}$ and, finally, we consider the identity $\nabla(\cdot)\boldsymbol{F} = \nabla_0(\cdot)$. Keeping these relations in mind the second integral of Equation[3.24] becomes

$$\begin{aligned}
&\int_V \boldsymbol{S} : \left(\nabla_0 \boldsymbol{u}^T \nabla_0 \delta \boldsymbol{v}\right)\ dV \\
&= \int_v J\boldsymbol{F}^{-1}\boldsymbol{\sigma}\boldsymbol{F}^{-T} : \left(\boldsymbol{F}^T \nabla \boldsymbol{u}^T \nabla \delta \boldsymbol{v}\ \boldsymbol{F}\right)\ J^{-1} dv \\
&= \int_v \nabla \boldsymbol{u}\ \boldsymbol{F}\boldsymbol{F}^{-1}\boldsymbol{\sigma}\boldsymbol{F}^{-T} : \nabla \delta \boldsymbol{v}\ \boldsymbol{F} dv \\
&= \int_v \boldsymbol{\sigma}\boldsymbol{F}^{-T} : \nabla \boldsymbol{u}^T \nabla \delta \boldsymbol{v}\ \boldsymbol{F} dv \\
&= \int_v \boldsymbol{\sigma} : \nabla \boldsymbol{u}^T \nabla \delta \boldsymbol{v}\ \boldsymbol{F}(\boldsymbol{F}^{-T})^T dv \\
&= \int_v \boldsymbol{\sigma} : \nabla \boldsymbol{u}^T \nabla \delta \boldsymbol{v}\ dv.
\end{aligned} \tag{3.25}$$

Now, we focus on the first integral of Equation[3.24]. Using again previuos the double dot property and recalling that $D\boldsymbol{E}[\boldsymbol{u}] = \boldsymbol{F}^T \boldsymbol{\epsilon}\ \boldsymbol{F}$ and $D\boldsymbol{E}[\delta \boldsymbol{v}] = \boldsymbol{F}^T \delta \boldsymbol{d}\ \boldsymbol{F}$ we obtain

$$\begin{aligned}
&\int_V D\boldsymbol{E}[\delta \boldsymbol{v}] : \mathbb{C} : D\boldsymbol{E}[\boldsymbol{u}]\ dV \\
&= \int_v \boldsymbol{F}^T \delta \boldsymbol{d}\ \boldsymbol{F} : \mathbb{C} : \boldsymbol{F}^T \boldsymbol{\epsilon}\ \boldsymbol{F}\ J^{-1} dv \\
&= \int_v \boldsymbol{F}^T \delta \boldsymbol{d}\ \boldsymbol{F} : \boldsymbol{F}\mathbb{C}\boldsymbol{F}^T : \boldsymbol{\epsilon}\ J^{-1} dv \\
&= \int_v \delta \boldsymbol{d} : \boldsymbol{F}\boldsymbol{F}\mathbb{C}\boldsymbol{F}^T\boldsymbol{F}^T : \boldsymbol{\epsilon}\ J^{-1} dv \\
&= \int_v \delta \boldsymbol{d} : \hat{\mathbb{C}} : \boldsymbol{\epsilon}\, dv
\end{aligned} \tag{3.26}$$

where $\hat{C}_{ijkl} = F_{iI}F_{jJ}F_{kK}F_{lL}C_{IJKL}J^{-1}$ is the *spatial elasticity tensor*. Hence, the directional derivative of the internal virtual works is

$$D\delta W_{int}(\boldsymbol{\phi}, \delta\boldsymbol{v})[\boldsymbol{u}] = \int_v \delta\boldsymbol{d} : \hat{\mathbb{C}} : \boldsymbol{\epsilon}\, dv + \int_v \boldsymbol{\sigma} : \nabla\boldsymbol{u}^T \nabla\delta\boldsymbol{v}\, dv = \tag{3.27}$$
$$= D\delta W_{int}^{mat}(\boldsymbol{\phi}, \delta\boldsymbol{v})[\boldsymbol{u}] + D\delta W_{int}^{geo}(\boldsymbol{\phi}, \delta\boldsymbol{v})[\boldsymbol{u}]$$

where $D\delta W_{int}^{mat}(\boldsymbol{\phi}, \delta\boldsymbol{v})[\boldsymbol{u}]$ and $D\delta W_{int}^{geo}(\boldsymbol{\phi}, \delta\boldsymbol{v})[\boldsymbol{u}]$ are the *material* and *geometrical* contribution to the resulting tangent stiffness matrix.

3.3 Finite Element Discretization of the Linearized Principle of Virtual Works

Once, we have linearized the PVW we are ready to perform the FEM discretization. It is convenient to discretize the PVW for each element (the superscript $^{(e)}$ refers to the finite element) and then assembly each local contribution to get the global FEM formulation.
The main idea of the finite element method is to discretize a generic functional space V in a finite dimension space V_h. This discretized space is spanned by a finite set of functions, called *basis functions*, $N_i = N_i(\boldsymbol{x})$, meaning

$$V_h = \text{span}\{N_1, N_2, ..., N_N\}.$$

where N is the total number of nodes of the computational grid. Let us consider a generic quantity $\boldsymbol{u}(\boldsymbol{X}, t)$. Its finite element discretization is

$$\boldsymbol{u}_h(\boldsymbol{x}, t) = \sum_{i=1}^{n} N_i(\boldsymbol{X})\boldsymbol{u}_i(t), \tag{3.28}$$

If we apply Equation[3.28] to Equation[3.29], the local or elemental PVW the finite element problem is:

Find $\boldsymbol{u}_h \in V_h$ such that

$$D\delta W^{(e)}(\boldsymbol{\phi}, \delta\boldsymbol{v}_h)[\boldsymbol{u}_h] = -\delta W^{(e)}(\boldsymbol{\phi}, \delta\boldsymbol{v}_h). \tag{3.29}$$

At this point we need to make the FEM discretization of Equation[3.29] explicit, i.e., find the expression for the left and right hand terms. We will denote hereafter by LHS the left hand side and by RHS the right hand side term. For details see Appendix A.

3.3.1 Finite Element Discretization of the RHS

The right hand side is given by

$$\delta W^{(e)}(\boldsymbol{\phi}, \delta \boldsymbol{v}_h) =$$

$$\underbrace{\int_{v^{(e)}} \rho \boldsymbol{a}_h \cdot \delta \boldsymbol{v}_h \, dv}_{\delta W_m^{(e)}(\boldsymbol{\phi}, \delta \boldsymbol{v}_h)} + \underbrace{\int_{v^{(e)}} \boldsymbol{\sigma} : \delta \boldsymbol{d}_h \, dv}_{\delta W_{int}^{(e)}(\boldsymbol{\phi}, \delta \boldsymbol{v}_h)} \underbrace{- \int_{v^{(e)}} \boldsymbol{f} \cdot \delta \boldsymbol{v}_h \, dv - \int_{a^{(e)}} \boldsymbol{t} \cdot \delta \boldsymbol{v}_h \, da}_{\delta W_{ext}^{(e)}(\boldsymbol{\phi}, \delta \boldsymbol{v}_h)}$$

Then, in order to obtain the FEM discretization, we have to deal with these three terms meaning the inertial, internal and external terms.
The FEM discretization **inertial term** $\delta W_m(\boldsymbol{\phi}, \delta \boldsymbol{v}_h)$ gives

$$\delta W_m^{(e)}(\boldsymbol{\phi}, \delta \boldsymbol{v}_h) = \delta \boldsymbol{v}^{T^{(e)}} \boldsymbol{M}^{(e)} \boldsymbol{a}^{(e)} \tag{3.30}$$

where the term $\boldsymbol{M}_{ij}^{(e)}$ can be expressed by using the identity matrix $\boldsymbol{I}$ as follows

$$\boldsymbol{M}_{ij}^{(e)} = m_{ij}^{(e)} \boldsymbol{I}, \tag{3.31}$$

and

$$m_{ij}^{(e)} = \int_{v^{(e)}} \rho N_i(\boldsymbol{X}) N_j(\boldsymbol{X}) \, dv. \tag{3.32}$$

If we now assembly each element contribute we obtain the discrete form of the global inertial virtual works

$$\delta W_m(\boldsymbol{\phi}, \delta \boldsymbol{v}_h) = \delta \boldsymbol{v}^T \boldsymbol{M} \boldsymbol{a}, \tag{3.33}$$

$$\delta W_m(\boldsymbol{\phi}, \delta \boldsymbol{v}_h) = \begin{bmatrix} \delta \boldsymbol{v}_1 \\ \delta \boldsymbol{v}_2 \\ \vdots \\ \delta \boldsymbol{v}_N \end{bmatrix} \begin{bmatrix} m_{11}\boldsymbol{I} & m_{12}\boldsymbol{I} & \cdots & m_{1N}\boldsymbol{I} \\ m_{21}\boldsymbol{I} & m_{22}\boldsymbol{I} & \cdots & m_{2N}\boldsymbol{I} \\ \vdots & \vdots & \ddots & \vdots \\ m_{N1}\boldsymbol{I} & m_{N2}\boldsymbol{I} & \cdots & m_{NN}\boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{a}_1(t) \\ \boldsymbol{a}_2(t) \\ \vdots \\ \boldsymbol{a}_N(t) \end{bmatrix} \tag{3.34}$$

What we could observe is that the global mass matrix $\boldsymbol{M}$ is a symmetric and sparse matrix. In fact the entries m_{ij} are different from zero only if node i and node j are connected via a finite element.
Often matrix $\boldsymbol{M}$ is written in an alternative form $\boldsymbol{M}^L$ called *lumped mass*

matrix. This matrix is a diagonal matrix and is obtained by replacing the diagonal term with the sum of the coefficient row by row. This is called *row sum* technique. In this way the mass matrix of Equation[3.34] is

$$\boldsymbol{M}^L = \begin{bmatrix} m_{11}^L\boldsymbol{I} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & m_{22}^L\boldsymbol{I} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & m_{NN}^L\boldsymbol{I} \end{bmatrix} \tag{3.35}$$

where

$$m_{ij}^L = \sum_{j=1}^{n} m_{ij}. \tag{3.36}$$

The FEM discretization of the **internal virtual works**, $\delta W_{int}^{(e)}(\boldsymbol{\phi}, \delta\boldsymbol{v}_h)$, gives

$$\delta W_{int}^{(e)}(\boldsymbol{\phi}, \delta\boldsymbol{v}_h) = \delta\boldsymbol{v}^{T^{(e)}}\boldsymbol{T}^{(e)} \tag{3.37}$$

where

$$\boldsymbol{T}_i^{(e)} = \int_{v^{(e)}} \boldsymbol{\sigma}\nabla N_i(\boldsymbol{X})\, dv \;\text{ or }\; \boldsymbol{T}_i^{(e)} = \int_{v^{(e)}} \boldsymbol{B}_i^T\boldsymbol{\sigma}\, dv \tag{3.38}$$

where $\boldsymbol{B}$ is the usual shape function derivative matrix adopting to Voigt notation. Now, if we assembly the contributes of each element we have the global discretized internal virtual works

$$\delta W_{int}(\boldsymbol{\phi}, \delta\boldsymbol{v}_h) = \delta\boldsymbol{v}^T\boldsymbol{T} \tag{3.39}$$

Finally, the FEM discretization of the **external term**, $\delta W_{ext}(\boldsymbol{\phi}, \delta\boldsymbol{v}_h)$, gives

$$\delta W_{ext}^{(e)}(\boldsymbol{\phi}, \delta\boldsymbol{v}_h) = \delta\boldsymbol{v}^{T^{(e)}}\boldsymbol{F}^{(e)} \tag{3.40}$$

where

$$\boldsymbol{F}_i^{(e)} = \int_{v^{(e)}} \boldsymbol{f} N_i(\boldsymbol{X})\, dv + \int_{a^{(e)}} \boldsymbol{t} N_i(\boldsymbol{X})\, da. \tag{3.41}$$

Equation[3.40] in global form, after the usual assembling procedure, becomes

$$\delta W_{ext}(\boldsymbol{\phi}, \delta \boldsymbol{v}_h) = \delta \boldsymbol{v}^T \boldsymbol{F} \tag{3.42}$$

Therefore, the right hand side of Equation[3.29] in the global form is:

$$\delta W(\boldsymbol{\phi}, \delta \boldsymbol{v}_h)[\boldsymbol{u}_h] = \delta \boldsymbol{v}^T \ (\boldsymbol{Ma} + \boldsymbol{T} - \boldsymbol{F}) = \delta \boldsymbol{v}^T \boldsymbol{R}, \tag{3.43}$$

where in literature $\boldsymbol{R}$ is called *residual vector*.

3.3.2 Finite Element Discretization of the LHS

At this point we have to discretize the left hand side of Equation[3.29], meaning the directional derivative of the virtual works. The left hand side of Equation[3.29] is given by the sum of the three contributions, meaning the directional derivatives of the inertial, internal and external virtual works:

$$\begin{aligned} &D\delta W^{(e)}(\boldsymbol{\phi}, \delta \boldsymbol{v}_h)[\boldsymbol{u}_h] = \\ &D\delta W_m^{(e)}(\boldsymbol{\phi}, \delta \boldsymbol{v}_h)[\boldsymbol{u}_h] + D\delta W_{int}^{(e)}(\boldsymbol{\phi}, \delta \boldsymbol{v}_h)[\boldsymbol{u}_h] - D\delta W_{ext}^{(e)}(\boldsymbol{\phi}, \delta \boldsymbol{v}_h)[\boldsymbol{u}_h]. \end{aligned} \tag{3.44}$$

The FEM discretization of the directional derivative of the **Inertial Term** $D\delta W_m^{(e)}(\boldsymbol{\phi}, \delta \boldsymbol{v}_h)[\boldsymbol{u}_h]$ is simply given by

$$D\delta W_m^{(e)}(\boldsymbol{\phi}, \delta \boldsymbol{v}_h)[\boldsymbol{u}_h] = \delta \boldsymbol{v}^{T^{(e)}} \boldsymbol{M}^{(e)} \boldsymbol{a}^{(e)}, \tag{3.45}$$

or after performing the usual assembling procedure by

$$D\delta W_m(\boldsymbol{\phi}, \delta \boldsymbol{v}_h)[\boldsymbol{u}_h] = \delta \boldsymbol{v}^T \boldsymbol{Ma}. \tag{3.46}$$

The FEM discretization of the directional derivative of the **Internal Term** $D\delta W_{int}^{(e)}(\boldsymbol{\phi}, \delta \boldsymbol{v}_h)[\boldsymbol{u}_h]$ is made by two terms as saw in Equation[3.27]. The discrete and local version of that equation is

$$D\delta W_{int}^{(e)}(\boldsymbol{\phi}, \delta \boldsymbol{v}_h)[\boldsymbol{u}_h] = D\delta W_{int}^{mat,(e)}(\boldsymbol{\phi}, \delta \boldsymbol{v}_h)[\boldsymbol{u}_h] + D\delta W_{int}^{geo,(e)}(\boldsymbol{\phi}, \delta \boldsymbol{v}_h)[\boldsymbol{u}_h]. \tag{3.47}$$

The material contribution is given by

$$D\delta W_{int}^{mat,(e)}(\boldsymbol{\phi}, \delta\boldsymbol{v}_h)[\boldsymbol{u}_h] = \delta\boldsymbol{v}^{T^{(e)}}\boldsymbol{K}^{mat,(e)}\boldsymbol{u}^{(e)} \tag{3.48}$$

while the geometric contribution is

$$D\delta W_{int}^{geo,(e)}(\boldsymbol{\phi}, \delta\boldsymbol{v}_h)[\boldsymbol{u}_h] = \delta\boldsymbol{v}^{T^{(e)}}\boldsymbol{K}^{geo,(e)}\boldsymbol{u}^{(e)} \tag{3.49}$$

where

$$\boldsymbol{K}_{ij}^{geo,(e)} = \int_{v^{(e)}} \nabla N_i \cdot \boldsymbol{\sigma}\nabla N_j \boldsymbol{I}\, dv \quad \text{or} \quad \boldsymbol{K}_{ij}^{geo,(e)} = \int_{v^{(e)}} \boldsymbol{B}_i^T\boldsymbol{\sigma}\boldsymbol{B}_j\boldsymbol{I}\, dv$$

Concerning the FEM discretization of the directional derivative of the **External Term** ,$D\delta W_{ext}^{(e)}(\boldsymbol{\phi}, \delta\boldsymbol{v}_h)[\boldsymbol{u}_h]$, depending on which kind of volume and surface forces are acting on the continuum body the directional derivative will change. As consequence also the FEM discretization will change. In order to keep general the discussion we will call the resulting matrix from the discretization as $\boldsymbol{K}^{ext,(e)}$. In general we will have

$$D\delta W_{ext}^{(e)}(\boldsymbol{\phi}, \delta\boldsymbol{v}_h)[\boldsymbol{u}_h] = \delta\boldsymbol{v}^{T,(e)} \cdot \boldsymbol{K}^{ext,(e)}\boldsymbol{u}^{(e)}. \tag{3.50}$$

The FEM discretization of the left hand side of Equation[3.29] is

$$\begin{aligned} &D\delta W^{(e)}(\boldsymbol{\phi}, \delta\boldsymbol{v}_h)[\boldsymbol{u}_h] = \\ &\delta\boldsymbol{v}^{T,(e)} \cdot \boldsymbol{M}^{(e)}\boldsymbol{a}^{(e)} + \delta\boldsymbol{v}^{T,(e)} \cdot \left(\boldsymbol{K}_{int}^{mat,(e)} + \boldsymbol{K}_{int}^{geo,(e)} - \boldsymbol{K}^{ext,(e)}\right)\boldsymbol{u}^{(e)}. \end{aligned} \tag{3.51}$$

where $\boldsymbol{K}_{int}^{mat,(e)} + \boldsymbol{K}_{int}^{geo,(e)} - \boldsymbol{K}^{ext,(e)} := \boldsymbol{K}^{tan,(e)}$ is called tangent stiffness matrix.

Finally, Equation[3.29] becomes

$$\delta\boldsymbol{v}^{T,(e)} \cdot \boldsymbol{M}^{(e)}\boldsymbol{a}^{(e)} + \delta\boldsymbol{v}^{T,(e)} \cdot \boldsymbol{K}^{tan,(e)}\boldsymbol{u}^{(e)} = -\delta\boldsymbol{v}^{T,(e)} \cdot \boldsymbol{R}^{(e)}. \tag{3.52}$$

or equivalently in the global form

$$\delta\boldsymbol{v}^T \cdot \boldsymbol{M}\boldsymbol{a} + \delta\boldsymbol{v}^T \cdot \boldsymbol{K}^{tan}\boldsymbol{u} = -\delta\boldsymbol{v}^T \cdot \boldsymbol{R}. \tag{3.53}$$

Before going any further, let us observe that residual $\boldsymbol{R}$ depends on $\boldsymbol{u}$ and $\boldsymbol{a}$ while the tangent stiffness matrix $\boldsymbol{K}^{tan}$ depends only, in general, on the displacement $\boldsymbol{u}$:

$$\boldsymbol{R} = \boldsymbol{R}(\boldsymbol{u}, \boldsymbol{a})$$

$$\boldsymbol{K}^{tan} = \boldsymbol{K}^{tan}(\boldsymbol{u})$$

Given the arbitrariness of the virtual velocity $\delta\boldsymbol{v}$, Equation[3.53] gives

$$\boldsymbol{M}\boldsymbol{a} + \boldsymbol{K}^{tan}(\boldsymbol{u})\,\boldsymbol{u} = -\boldsymbol{R}(\boldsymbol{u},\boldsymbol{a}) \tag{3.54}$$

3.4 Newton-Raphson procedure and Newmark time integration scheme

Equation[3.54] can be interpreted as the linearization of the discretized virtual works $\delta W(\phi_{n+1}^{(k+1)}, \delta\boldsymbol{v}_h)$ at the time step $(n+1)$ and at the $(k+1)$ Newton-Raphson iteration where the Taylor's series has been truncated at the first order terms (same of Equation[3.9])

$$\delta W(\phi_{n+1}^{(k+1)}, \delta\boldsymbol{v}_h) \simeq \delta W(\phi_{n+1}^{(k)}, \delta\boldsymbol{v}_h) + D\delta W(\phi_{n+1}^{(k)}, \delta\boldsymbol{v}_h)[\Delta\boldsymbol{u}_{h,n+1}] = 0 \tag{3.55}$$

or

$$\boldsymbol{R}(\boldsymbol{a}_{n+1}^{(k+1)}, \boldsymbol{u}_{n+1}^{(k+1)}) \simeq \boldsymbol{R}(\boldsymbol{a}_{n+1}^{(k)}, \boldsymbol{u}_{n+1}^{(k)}) + \boldsymbol{M}\Delta\boldsymbol{a}_{n+1} + \boldsymbol{K}^{tan}(\boldsymbol{u}_{n+1}^{(k)})\Delta\boldsymbol{u}_{n+1} = 0 \tag{3.56}$$

where $\Delta\boldsymbol{u}_{n+1}$ plays the role of the increment. At this point we can introduce the Newmark time integration scheme. The general Newmark time-stepping scheme is given by the following relationships

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + \Delta t\boldsymbol{v}_n + \frac{\Delta t^2}{2}\left[(1-2\beta)\boldsymbol{a}_n + 2\beta\boldsymbol{a}_{n+1}\right], \tag{3.57}$$

$$\boldsymbol{v}_{n+1} = \boldsymbol{v}_n + \Delta t\left[(1-\gamma)\boldsymbol{a}_n + \gamma\boldsymbol{a}_{n+1}\right]. \tag{3.58}$$

What we can observe is that for both Equations[3.57],[3.58] we need the previous time step value. For instance Equation[3.57] needs $\boldsymbol{u}_n$. Therefore at each time step $(n+1)$ an initial guess, a prediction, is made and then corrected until the dynamics equilibrium is satisfied. This is achieved by using the Newton-Raphson solution strategy. We start from $k = 0$ and we set $\boldsymbol{a}_{n+1}^{(0)} = \boldsymbol{a}_n$. In this way Equation[3.57] and Equation[3.58] becomes

$$\boldsymbol{u}_{n+1}^{(0)} = \boldsymbol{u}_n + \Delta t \boldsymbol{v}_n + \frac{\Delta t^2}{2}, \boldsymbol{a}_n \tag{3.59}$$

$$\boldsymbol{v}_{n+1}^{(0)} = \boldsymbol{v}_n + \Delta t \boldsymbol{a}_n. \tag{3.60}$$

where $\boldsymbol{u}_n$, $\boldsymbol{v}_n$ and $\boldsymbol{a}_n$ are the solutions of the previous time-step. For a generic Newton-Raphson iteration we can use Equations[3.57],[3.58] in order to obtain the following corrections

$$\boldsymbol{a}_{n+1}^{(k+1)} = \boldsymbol{a}_{n+1}^{(k)} + \Delta \boldsymbol{a}_{n+1} \tag{3.61}$$

$$\begin{aligned}
\boldsymbol{u}_{n+1}^{(k+1)} &= \boldsymbol{u}_n + \Delta t \boldsymbol{v}_n + \tfrac{\Delta t^2}{2} \left[(1-2\beta)\boldsymbol{a}_n + 2\beta \boldsymbol{a}_{n+1}^{(k+1)}\right] \\
&= \boldsymbol{u}_n + \Delta t \boldsymbol{v}_n + \tfrac{\Delta t^2}{2} \left[(1-2\beta)\boldsymbol{a}_n)\right] + \Delta t^2 \beta \boldsymbol{a}_{n+1}^{(k)} + \Delta t^2 \beta \Delta \boldsymbol{a}_{n+1} \\
&= \boldsymbol{u}_n + \Delta t \boldsymbol{v}_n + \tfrac{\Delta t^2}{2} \left[(1-2\beta)\boldsymbol{a}_n) + 2\beta \boldsymbol{a}_{n+1}^{(k)}\right] + \Delta t^2 \beta \Delta \boldsymbol{a}_{n+1} \\
&= \boldsymbol{u}_{n+1}^{(k)} + \Delta \boldsymbol{u}_{n+1}
\end{aligned} \tag{3.62}$$

$$\begin{aligned}
\boldsymbol{v}_{n+1}^{(k+1)} &= \boldsymbol{v}_n + \Delta t \left[(1-\gamma)\boldsymbol{a}_n + \gamma \boldsymbol{a}_{n+1}^{(k+1)}\right] \\
&= \boldsymbol{v}_n + \Delta t \left[(1-\gamma)\boldsymbol{a}_n + \gamma \boldsymbol{a}_{n+1}^{(k)}\right] + \Delta t \gamma \Delta \boldsymbol{a}_{n+1} \\
&= \boldsymbol{v}_n^{(k)} + \Delta \boldsymbol{v}_{n+1}
\end{aligned} \tag{3.63}$$

where in Equations[3.62],[3.63] we used Equation[3.61]. Moreover, in these equations we have defined the following important quatities

$$\Delta \boldsymbol{u}_{n+1} = \Delta t^2 \beta \Delta \boldsymbol{a}_{n+1} \tag{3.64}$$

$$\Delta \boldsymbol{v}_{n+1} = \Delta t \gamma \Delta \boldsymbol{a}_{n+1} \tag{3.65}$$

which are used to update the solutions.

Two ways are now available in order to solve Equation[3.56] and obtain $\Delta \boldsymbol{u}_{n+1}$

or $\Delta\boldsymbol{a}_{n+1}$. The first one consists to write $\Delta\boldsymbol{u}_{n+1}$ in terms of $\Delta\boldsymbol{a}_{n+1}$ and in this case we will talk about *Newmark acceleration-corrector method*. The second way consists to write $\Delta\boldsymbol{a}_{n+1}$ in terms of $\Delta\boldsymbol{u}_{n+1}$ and we will call it *Newmark displacement-corrector method*. In order to do that, in both displacement-based and acceleration-based methods we make use of Equation[3.64]. For instance if we use the acceleration-based method Equation[3.56] gives

$$\boxed{\left[\boldsymbol{M} + \boldsymbol{K}^{tan}(\boldsymbol{u}_{n+1}^{(k)})\Delta t^2 \beta\right] \Delta\boldsymbol{a}_{n+1} = -\boldsymbol{R}(\boldsymbol{a}_{n+1}^{(k)}, \boldsymbol{u}_{n+1}^{(k)}) \qquad (3.66)}$$

while if we use the displacement-based method Equation[3.56] becomes

$$\left[\frac{1}{\Delta t^2 \beta}\boldsymbol{M} + \boldsymbol{K}^{tan}(\boldsymbol{u}_{n+1}^{(k)})\right] \Delta\boldsymbol{u}_{n+1} = -\boldsymbol{R}(\boldsymbol{a}_{n+1}^{(k)}, \boldsymbol{u}_{n+1}^{(k)}). \tag{3.67}$$

Chapter 4

Material Point Method

4.1 Introduction

The Material Point Method is a particles-based method and its origin dates back to the Particle-In-Cell (PIC) method. The PIC method was developed at Los Alamos National Laboratory by Herlow et al. [21] and it was used only for fluid mechanics application. The method consists to discretized the fluid with Lagrangian particle while the mesh is an Eulerian one meaning it is fixed in space and it does not move with the particles. The computation is divided into two phases, a Lagrangian phase and an Eulerian phase. In the former the variable are advanced i.e are updated. In the latter the mesh is mapped back to the original position. In PIC method only the material position and mass are carried by the particles while all the fluid's variable are stored in the mesh. This leads to numerical diffusion and in order to overcome this issue it has been proposed by Brackbill [7] a fully Lagrangian particle method known as Fluid Implicit Particle (FLIP).Thanks to Suslky [29] the method started to be used also for solid mechanics and the FLIP method started to be called as MPM.

In the MPM method the continuum body is discretized by a finite set of material points. This points carry the relevant quantities and are tracked through the deformation process. The continuum body is nested in a Lagrangian mesh, the background grid, where the balance momentum equations are solved for each node. Once we solve the momentum equations the main properties are mapped to the material point and the background grid in reset. No permanent information are stored on the nodal grid. Then, the grid information are updated again by mapping the information from particles to grid nodes and the momentum equations can be solved again and the procedure can restart. In the classical MPM we can identify three main steps FIG[4.1]:

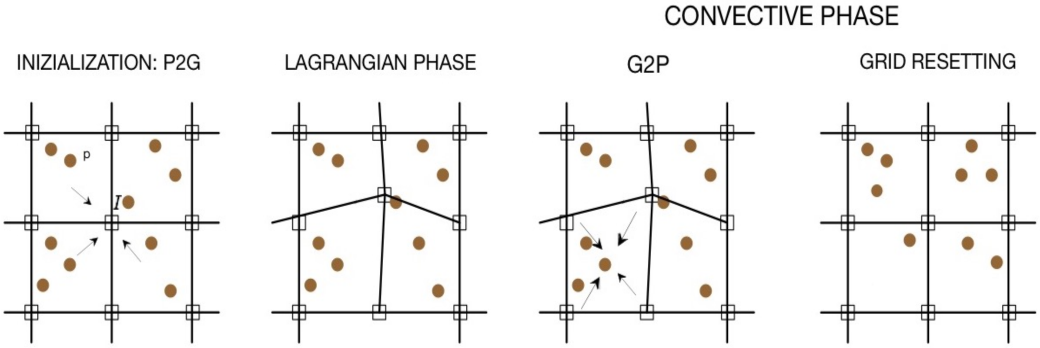

Figure 4.1: MPM consists in three main steps: (1) Particle to Grid (P2G) the information are mapped from the particles to the nodes; this is useful to initialize the nodal values. (2) Lagrangian phase is where the balance equations are solved and the grid is updating. (3) In Convective phase the information are mapped back to the particles and the grid is resetted.

- **Inizialization phase**: The initialization phase consists to map the information previously obtained at time step t^n from the particles to the nodal grid by using the basis function interpolation. This phase in also called *Particle to Grid* or P2G.

- **Lagrangian phase**: In this phase momentum equations are solved for each node of the background grid. If we use an Update Lagrangian MPM, the solution procedure is exactly the same of the Update Lagrangian FEM. Then, position and velocity of the grid are update.

- **Convective phase**: Once position and velocity of the nodes of the grid have been updated we have to map back these information the the particle point. Thus, the relevant quantities such as position and velocity are mapped back from grid nodes to the material particles by means of basis functions interpolation (G2P). Then, if the grid is distorted we can reset it. The grid resetting is not mandatory. We can also replace the distorted grid with any other suitable grid.

4.2 Weak Formulation and FEM Discretization

Let us consider the weak formulation of the linear momentum balance (Equation[2.32]), i.e. the principle of virtual works (Equation[3.14])

$$\int_v \rho \boldsymbol{a} \cdot \delta \boldsymbol{v} \; dv + \int_v \rho \hat{\boldsymbol{\sigma}} : \delta \boldsymbol{d} \; dv - \int_v \rho \boldsymbol{f} \cdot \delta \boldsymbol{v} \; dv - \int_a \rho \boldsymbol{t} \cdot \delta \boldsymbol{v} \; da = 0, \qquad (4.1)$$

where $\hat{\boldsymbol{\sigma}} = \dfrac{\boldsymbol{\sigma}}{\rho}$ is the specific stress.

In MPM the physical domain, Ω, is discretized by sub-domains, Ω_p, see

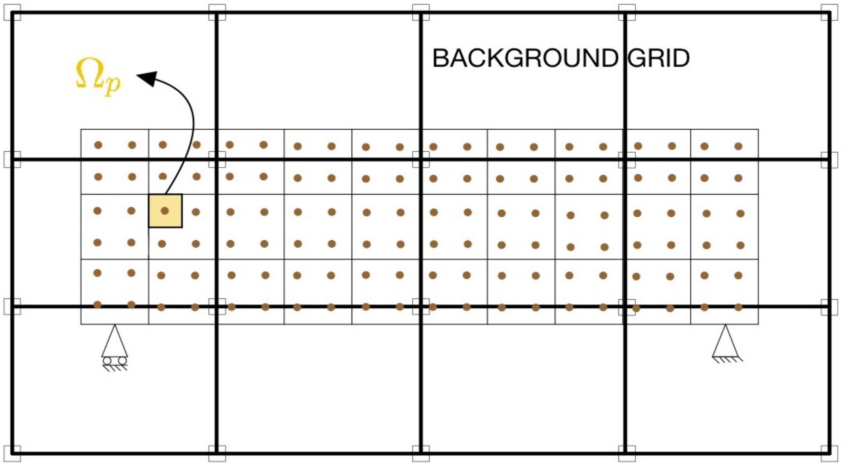

Figure 4.2: MPM discretization for a beam. The region marked in yellow is an example of sub-domain Ω_p

FIG[4.2], in such way that

$$\Omega = \cup_{p=1}^{n_p} \Omega_p.$$

For each sub-domain is associated a material point p, where the entire mass of the sub-domain is concentrated. In this way, from a mathematical point of view, the mass density can be expressed in terms of the Dirac delta as follows

$$\boxed{\rho(\boldsymbol{x}, t) = \sum_{p=1}^{n_p} \rho_p = \sum_{p=1}^{n_p} m_p \delta(\boldsymbol{x} - \boldsymbol{x}_p). \qquad (4.2)}$$

In order to be consistent with the unit measure the Dirac delta has the dimension of the inverse of volume. Substituting Equation[4.2] into Equation[4.1], introducing in the surface integral of the external forces the thickness h and using the identity

$$\int f(x)\delta(x - x_p)dx = f(x_p)$$

Equation[4.1] gives

$$\sum_{p=1}^{n_p} m_p \delta\boldsymbol{v}(\boldsymbol{x}_p)\boldsymbol{a}(\boldsymbol{x}_p) + \sum_{p=1}^{n_p} m_p \delta\boldsymbol{d}(\boldsymbol{x}_p) : \hat{\boldsymbol{\sigma}}(\boldsymbol{x}_p) - \sum_{p=1}^{n_p} m_p \delta\boldsymbol{v}(\boldsymbol{x}_p)\boldsymbol{f} - \sum_{p=1}^{n_p} m_p \delta\boldsymbol{v}(\boldsymbol{x}_p)h^{-1}\boldsymbol{t}(\boldsymbol{x}_p) = 0. \tag{4.3}$$

At this point we can proceed with the FEM discretization of the various quantities involved:

$$\boldsymbol{a}(\boldsymbol{x}_p, t) = \sum_{I=1}^{n_n} N_I(\boldsymbol{x}_p)\boldsymbol{a}_I(t), \tag{4.4}$$

$$\delta\boldsymbol{v}(\boldsymbol{x}_p) = \sum_{I=1}^{n_n} N_I(\boldsymbol{x}_p)\delta\boldsymbol{v}_I, \tag{4.5}$$

$$\delta\boldsymbol{d} = \frac{1}{2}\sum_{I=1}^{n_n} (\ \delta\boldsymbol{v}_I \otimes \nabla N_I(\boldsymbol{x}_p) + \nabla N_I(\boldsymbol{x}_p) \otimes \delta\boldsymbol{v}_I\ ) \tag{4.6}$$

where I and n_n are the generic node and the number of nodes of the background grid respectively. Substituting Equations[4.4], [4.5] and [4.6] and evoking the symmetry of the specific Cauchy tensor we have

$$\begin{gathered}\sum_{p=1}^{n_p} m_p \sum_{I=1}^{n_n} N_I(\boldsymbol{x}_p)\delta\boldsymbol{v}_I \sum_{J=1}^{n_n} N_J(\boldsymbol{x}_p)\boldsymbol{a}_J(t) + \sum_{p=1}^{n_p} m_p \sum_{I=1}^{n_n} \delta\boldsymbol{v}_I \cdot \hat{\boldsymbol{\sigma}}(\boldsymbol{x}_p)\nabla N_I(\boldsymbol{x}_p) - \\ -\sum_{p=1}^{n_p} m_p \sum_{I=1}^{n_n} N_I(\boldsymbol{x}_p)\delta\boldsymbol{v}_I\boldsymbol{f} - \sum_{p=1}^{n_p} m_p \sum_{I=1}^{n_n} N_I(\boldsymbol{x}_p)\delta\boldsymbol{v}_I h^{-1}\boldsymbol{t}(\boldsymbol{x}_p) = 0.\end{gathered} \tag{4.7}$$

Due to the arbitrariness of the test function $\delta\boldsymbol{v}$, for each grid node $I = 1, 2, ..., n_n$ we have that Equation[4.7] gives

$$\begin{aligned}&\sum_{p=1}^{n_p} m_p N_I(\boldsymbol{x}_p) \sum_{J=1}^{n_n} N_J(\boldsymbol{x}_p)\boldsymbol{a}_J + \sum_{p=1}^{n_p} m_p\hat{\boldsymbol{\sigma}}(\boldsymbol{x}_p)\nabla N_I(\boldsymbol{x}_p) = \\ &= \sum_{p=1}^{n_p} N_I(\boldsymbol{x}_p)\boldsymbol{f}(\boldsymbol{x}_p) + \sum_{p=1}^{n_p} N_I(\boldsymbol{x}_p)h^{-1}\boldsymbol{t}(\boldsymbol{x}_p)\end{aligned} \tag{4.8}$$

that in compact form gives

$$\boxed{m_{IJ}\boldsymbol{a}_J = \boldsymbol{f}_I^{int} + \boldsymbol{f}_I^{ext}, \qquad I = 1, 2, ..., n_n \qquad (4.9)}$$

where m_{IJ} are the components of the *consistent mass matrix*

$$m_{IJ} = \sum_{p=1}^{n_p} m_p N_I(\boldsymbol{x}_p) N_J(\boldsymbol{x}_p), \tag{4.10}$$

$\boldsymbol{f}_I^{int}$ is the *internal force vector*

$$\boldsymbol{f}_I^{int} = -\sum_{p=1}^{n_p} m_p\hat{\boldsymbol{\sigma}}(\boldsymbol{x}_p)\nabla N_I(\boldsymbol{x}_p) = -\sum_{p=1}^{n_p} V_p\boldsymbol{B}_I^T(\boldsymbol{x}_p)\boldsymbol{\sigma}(x_p), \tag{4.11}$$

and $\boldsymbol{f}_I^{ext}$ is the *external force vector*

$$\boldsymbol{f}_I^{ext} = \sum_{p=1}^{n_p} N_I(\boldsymbol{x}_p)\boldsymbol{f}(\boldsymbol{x}_p) + \sum_{p=1}^{n_p} N_I(\boldsymbol{x}_p)h^{-1}\boldsymbol{t}(\boldsymbol{x}_p), \tag{4.12}$$

where matrix $\boldsymbol{B}_I^T(\boldsymbol{x}_p)$ in three dimension is

$$\boldsymbol{B}_I^T(\boldsymbol{x}_p) = \left[\frac{\partial N_I}{\partial x}(\boldsymbol{x}_p), \frac{\partial N_I}{\partial y}(\boldsymbol{x}_p), \frac{\partial N_I}{\partial z}(\boldsymbol{x}_p)\right].$$

These expressions can be also written for the entire grid by assembling each local contributes

$$\boldsymbol{M}\boldsymbol{a} = \boldsymbol{f}, \tag{4.13}$$

meaning

$$\begin{bmatrix} m_{11}\boldsymbol{I} & m_{12}\boldsymbol{I} & \cdots & m_{1n_n}\boldsymbol{I} \\ m_{21}\boldsymbol{I} & m_{22}\boldsymbol{I} & \cdots & m_{2n_n}\boldsymbol{I} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n_n1}\boldsymbol{I} & m_{n_n2}\boldsymbol{I} & \cdots & m_{n_nn_n}\boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{a}_1(t) \\ \boldsymbol{a}_2(t) \\ \vdots \\ \boldsymbol{a}_{n_n}(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_1^{int} + \boldsymbol{f}_1^{ext} \\ \boldsymbol{f}_2^{int} + \boldsymbol{f}_2^{ext} \\ \vdots \\ \boldsymbol{f}_{n_n}^{int} + \boldsymbol{f}_{n_n}^{ext} \end{bmatrix}. \tag{4.14}$$

Often the consistent mass matrix is turned into a *lumped mass matrix* as we saw in Equation[3.35]

$$m_{II}^L = \sum_{J=1}^{n_n} m_{IJ} = \sum_{p=1}^{n_p} m_p N_I(\boldsymbol{x}_p) \tag{4.15}$$

in this way the mass matrix of Equation[4.13] becomes

$$\boldsymbol{M}^L = \begin{bmatrix} m_{11}^L\boldsymbol{I} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & m_{22}^L\boldsymbol{I} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & m_{n_nn_n}^L\boldsymbol{I} \end{bmatrix}. \tag{4.16}$$

In order to solve Equation[4.9], which is a second order differential equation in time, we can use an explicit integration scheme or an implicit integration scheme. The former scheme involves quantities referred to the previous time step. As consequence an explicit scheme does not require to solve a linear system. While, the latter scheme involves quantities referred to the current time step. In general explicit scheme are *conditionally stable* meaning that the time step has to be sufficiently small to give a consistent solution. On the other hand explicit time schemes are cheaper than implicit time schemes computationally speaking but the latter are *unconditionally stable* meaning that the time step does not affect stability.

4.3 Explicit Formulation

Explicit schemes have to be preferred when we deal with transient problems such as hypervelocity impacts or blasts and a small time step is needed by the nature of the phenomena. Here below, we report two explicit methods.

Forward Euler. The simplest and straightforward explicit method is the *Explicit or Forward Euler*. In this scheme position and velocity are updated as follows

$$\boldsymbol{x}_I^{n+1} = \boldsymbol{x}_I^n + \Delta t^n \, \boldsymbol{v}_I^{n+1} \tag{4.17}$$

$$\boldsymbol{v}_I^{n+1} = \boldsymbol{v}_I^n + \Delta t^n \, \boldsymbol{a}_I^n \tag{4.18}$$

where $\boldsymbol{a}_I^n = \dfrac{\boldsymbol{f}_I^n}{m_{II}^n}$ and $\Delta t^n = t^{n+1} - t^n$. Once we computed the nodal grid velocity and position we can use them to update the particle velocity and position. Two approaches are possible, the *Particle In Cell* (PIC) and the *Fluid Implicit Particle method* (FLIP)

$$\boldsymbol{v}_p^{n+1} = \sum_{I=1}^{N_{pI}} N_I(\boldsymbol{x}_p^n) \boldsymbol{v}_I^{n+1} \qquad \text{(PIC)} \tag{4.19}$$

$$\boldsymbol{v}_p^{n+1} = \boldsymbol{v}_p^n + \sum_{I=1}^{N_{pI}} N_I(\boldsymbol{x}_p^n) \left[\boldsymbol{v}_I^{n+1} - \boldsymbol{v}_I^n\right] \qquad \text{(FLIP)} \tag{4.20}$$

where N_{pI} is the number of node of the element to which particle p belongs to. If we add an artificial parameter we can take into account both methods. Let's call this parameter α and let it takes value 0 or 1. If $\alpha = 1$ we have the FLIP scheme while if we set $\alpha = 0$ we have the PIC method.

$$\boldsymbol{v}_p^{n+1} = \alpha \left(\boldsymbol{v}_p^n + \sum_{I=1}^{N_{pI}} N_I(\boldsymbol{x}_p^n) \left[\boldsymbol{v}_I^{n+1} - \boldsymbol{v}_I^n\right] \right) + (1-\alpha) \sum_{I=1}^{N_{pI}} N_I(\boldsymbol{x}_p^n) \boldsymbol{v}_I^{n+1}. \tag{4.21}$$

At this point we can also update the particle p position

$$\boldsymbol{x}_p^{n+1} = \boldsymbol{x}_p^n + \Delta t^n \boldsymbol{v}_p^{n+1}. \tag{4.22}$$

Finally, we can update deformation gradient $\boldsymbol{F}$, velocity gradient $\boldsymbol{l}$, rate of deformation gradient $\boldsymbol{d}$ and other kinematic quantities

$$\boldsymbol{l}_p^{n+1} = \nabla \boldsymbol{v}_p^{n+1} = \sum_{I=1}^{N_{pI}} \nabla N_I(\boldsymbol{x}_p^n) \boldsymbol{v}_I^{n+1} \tag{4.23}$$

$$\boldsymbol{F}_p^{n+1} = \left(\boldsymbol{I} + \boldsymbol{l}_p^{n+1} \Delta t^n\right) \boldsymbol{F}_p^n \tag{4.24}$$

$$J = det \boldsymbol{F}_p^{n+1} \tag{4.25}$$

$$V_p^{n+1} = J V_p^0 \quad \rho_p^{n+1} = J^{-1} \rho_0 \tag{4.26}$$

$$\boldsymbol{d}_p^{n+1} = \frac{1}{2} \left[\boldsymbol{l}_p^{n+1} + (\boldsymbol{l}_p^{n+1})^T\right] \tag{4.27}$$

$$\Delta \boldsymbol{\epsilon}_p^{n+1} = \Delta t^n \boldsymbol{d}_p^{n+1} \tag{4.28}$$

$$\boldsymbol{\sigma}_p^{n+1} = \boldsymbol{\sigma}_p^n + \Delta \boldsymbol{\sigma}_p \left(\Delta \boldsymbol{\epsilon}_p^{n+1}\right) \text{ or } \boldsymbol{\sigma}_p^{n+1} = \boldsymbol{\sigma}_p^{n+1}(\boldsymbol{F}_p^{n+1}) \tag{4.29}$$

Two possible choices are available when we have to updated the stress, the *Update Stress Last* (USL) and *Update Stress First* (USF). If we follow the above procedure it means we adopted the USL since the stress is updated at the end.

Central Differences. The central differences method is one of the most popular explicit scheme in computational mechanics. In this method we assume known displacement, velocity and acceleration at time $t \leq t^n$. Moreover, in this contest we denote the time increment as

$$\Delta t^{n+\frac{1}{2}} = t^{n+1} - t^n; \quad \Delta t^n = t^{n+\frac{1}{2}} - t^{n-\frac{1}{2}}.$$

The velocity $\boldsymbol{v}_I^{n+\frac{1}{2}}$ at time $t^{n+\frac{1}{2}}$ and the acceleration $\boldsymbol{a}_I^n$ at time t^n are given respectively by

$$\boldsymbol{v}_I^{n+\frac{1}{2}} = \frac{\boldsymbol{x}_I^{n+1} - \boldsymbol{x}_I^n}{t^{n+1} - t^n} = \frac{1}{\Delta t^{n+\frac{1}{2}}} \left(\boldsymbol{u}_I^{n+1} - \boldsymbol{u}_I^n\right), \tag{4.30}$$

$$\boldsymbol{a}_I^n = \frac{\boldsymbol{v}_I^{n+\frac{1}{2}} - \boldsymbol{v}_I^{n-\frac{1}{2}}}{t^{n+\frac{1}{2}} - t^{n-\frac{1}{2}}} = \frac{1}{\Delta t^n}\left(\boldsymbol{v}_I^{n+\frac{1}{2}} - \boldsymbol{v}_I^{n-\frac{1}{2}}\right). \tag{4.31}$$

At this point if we solve Equation (4.31) for $\boldsymbol{v}_I^{n+\frac{1}{2}}$ and Equation (4.30) for $\boldsymbol{u}_I^{n+1}$ we obtain

$$\boldsymbol{v}_I^{n+\frac{1}{2}} = \boldsymbol{v}_I^{n-\frac{1}{2}} + \Delta t^n \boldsymbol{a}_I^n, \tag{4.32}$$

$$\boldsymbol{x}_I^{n+1} = \boldsymbol{x}_I^n + \Delta t^{n+\frac{1}{2}} \boldsymbol{v}_I^{n+\frac{1}{2}}, \tag{4.33}$$

where $\boldsymbol{a}_I^n = \dfrac{\boldsymbol{f}_I^n}{m_{II}^n}$. Now, we can update particle velocity and position. For instance, if we adopt a FLIP approach the updated particle velocity is

$$\boldsymbol{v}_p^{n+\frac{1}{2}} = \boldsymbol{v}_p^{n-\frac{1}{2}} + \sum_{I=1}^{N_{pI}} N_I(\boldsymbol{x}_p^n)\left(\boldsymbol{v}_I^{n+\frac{1}{2}} - \boldsymbol{v}_I^{n-\frac{1}{2}}\right), \tag{4.34}$$

and the updated particle position is

$$\boldsymbol{x}_p^{n+1} = \boldsymbol{x}_p^n + \Delta t^{n+\frac{1}{2}} \sum_{I=1}^{N_{pI}} N_I(\boldsymbol{x}_p^n) \boldsymbol{v}_I^{n+\frac{1}{2}}. \tag{4.35}$$

Here below we report the update of other kinematic and stress quantities

$$\boldsymbol{l}_p^{n+\frac{1}{2}} = \sum_{I=1}^{N_{pI}} \nabla N_I(\boldsymbol{x}_p^n) \boldsymbol{v}_I^{n+\frac{1}{2}} \tag{4.36}$$

$$\boldsymbol{F}_p^{n+1} = \left(\boldsymbol{I} + \boldsymbol{l}_p^{n+\frac{1}{2}} \Delta t^{n+\frac{1}{2}}\right) \boldsymbol{F}_p^n \tag{4.37}$$

$$\boldsymbol{d}_p^{n+\frac{1}{2}} = \frac{1}{2}\left[\boldsymbol{l}_p^{n+\frac{1}{2}} + (\boldsymbol{l}_p^{n+\frac{1}{2}})^T\right] \tag{4.38}$$

$$\Delta \boldsymbol{\epsilon}_p^{n+\frac{1}{2}} = \Delta t^n \boldsymbol{d}_p^{n+\frac{1}{2}} \tag{4.39}$$

$$\boldsymbol{\sigma}_p^{n+1} = \boldsymbol{\sigma}_p^{n+1}(\boldsymbol{F}_p^{n+1}) \tag{4.40}$$

4.4 Implicit Formulation

The most used implicit schemes are the Newmark methods already seen in the previous chapter. We recall that this family of methods consist in computing velocity and displacement at time t^{n+1} as

$$\boldsymbol{u}_I^{n+1} = \boldsymbol{u}_I^n + \Delta t \boldsymbol{v}_I^n + \frac{\Delta t^2}{2}\left[(1-2\beta)\boldsymbol{a}_I^n + 2\beta \boldsymbol{a}_I^{n+1}\right] \quad 0 \le 2\beta \le 1, \tag{4.41}$$

$$\boldsymbol{v}_I^{n+1} = \boldsymbol{v}_I^n + \Delta t\left[(1-\gamma)\boldsymbol{a}_I^n + \gamma \boldsymbol{a}_I^{n+1}\right] \quad 0 \le \gamma \le 1. \tag{4.42}$$

Observe that if $\beta = 0$ we have an explicit scheme. The grid node acceleration is obtained by solving Equation[4.41] for $\boldsymbol{a}_I^n$

$$\boldsymbol{a}_I^{n+1} = \frac{1}{\beta \Delta t^2}\left(\boldsymbol{x}_I^{n+1} - \underbrace{\boldsymbol{x}_I^n + \Delta t\, \boldsymbol{v}_I^n + \frac{\Delta t^2}{2}\,(1-2\beta)\, \boldsymbol{a}_I^n}_{\tilde{\boldsymbol{x}}_I^n}\right). \tag{4.43}$$

Observe that $\tilde{\boldsymbol{x}}_I^n$ depends on only quantities at the previous time step. An implicit scheme requires also to linearized the equations of motion. By performing the same linearization procedure done in the Update Lagrangian Formulation we end up with the acceleration-based method (see Equation[3.66])

$$\boxed{\left[\boldsymbol{M} + \boldsymbol{K}(\boldsymbol{x}_{n+1}^{(k)})\Delta t^2 \beta\right] \Delta \boldsymbol{a}_{n+1} = -\boldsymbol{R}(\boldsymbol{a}_{n+1}^{(k)}, \boldsymbol{x}_{n+1}^{(k)}) \qquad (4.44)}$$

where:

- $\boldsymbol{R}(\boldsymbol{a}_{n+1}^{(k)}, \boldsymbol{x}_{n+1}^{(k)})$ is the residual vector given by
$$-\boldsymbol{R}(\boldsymbol{a}_{n+1}^{(k)}, \boldsymbol{x}_{n+1}^{(k)}) = \boldsymbol{M}\boldsymbol{a}_{n+1}^{(k)} + \boldsymbol{K}(\boldsymbol{x}_{n+1}^{(k)})\boldsymbol{x}_{n+1}^{(k)},$$
- $\boldsymbol{M}$ is the lumped (or consistent) mass matrix of Equation (11)
- $\boldsymbol{K}$ is the *tangent matrix* whose component in current configuration are
$$\boldsymbol{K}_{IJ} = \boldsymbol{K}_{IJ}^{mat} + \boldsymbol{K}_{IJ}^{geo} =$$
$$\sum_p V_p \boldsymbol{B}_I^T(\boldsymbol{x}_p)\boldsymbol{D}(\boldsymbol{x}_p)\boldsymbol{B}_J(\boldsymbol{x}_p) + \sum_p V_p \boldsymbol{B}_I^T(\boldsymbol{x}_p)\boldsymbol{\sigma}(\boldsymbol{x}_p)\boldsymbol{B}_J(\boldsymbol{x}_p)\boldsymbol{I}$$

- $\Delta \boldsymbol{a}_{n+1}$ is the increment solution that we use to update the acceleration

$$\boldsymbol{a}_{n+1}^{(k+1)} = \boldsymbol{a}_{n+1}^{(k)} + \Delta \boldsymbol{a}_{n+1}$$

The updating of the various kinematic and stress quantities is the same of the classical Update Lagrangian FEM formulation.

4.5 Comparison between MPM and FEM

One might wonder about the convenience of using MPM rather than other computational methods such as FEM. The convenience of using one method rather than another depends on the problem being simulated. It is well known that for extremely large deformations the accuracy of FEM suffers particularly because the elements can be excessively distorted. As far as MPM is concerned, this does not occur because at each computational cycle the calculation mesh is reset and the information is stored in the material points. It follows that MPM is particularly efficient for large deformations. Vice versa, the latter mentioned method shows difficulties in the area of small deformations because of the particle numerical quadrature. In fact, in FEM, the numerical quadrature takes place in Gauss points, which ensure an accurate integration, while MPM uses the particles as integration points (particle integration) so we lose this accuracy. We said that at each computational cycle the computational grid is reset. It could also be replaced with another computational mesh. This is for the usual reason that the information is not stored in the grid. Reason for which the fractures can be simulated effectively.

Although MPM is more robust than FEM when we are dealing with large deformations, the accuracy of numerical quadrature remains higher in FEM. In order to obtain the best of these two methods, several MPM-FEM coupling algorithms have been developed. In coupled problems the body that undergoes greater deformation is discretized with MPM while the body with less deformation with FEM.

FEM

- Guass Quadrature
- Small deformations ed displacements
- Information stored on computational mesh nodes

MPM

- Particles Quadrature
- Very large deformations and displacements
- No relevant information is stored on computational mesh nodes just on particles

Chapter 5

Lagrange Multipliers Method in Computational Mechanics

In order to append boundary conditions (both conforming, non-conforming, first type or second type) we can treat them as constraints and different approaches are available from the optimization theory. For example, *penalty*, *Lagrange Multipliers*, *Augmented Lagrangian* and *Perturbed Lagrangian*. In this thesis we are going to speak mainly about the Lagrange Multipliers method and which problems involves this approach. The penalty approach is briefly introduced int he following sections.

5.1 Overview of the Penalty Method

The theory of the Penalty method provides that the constraint is imposed by adding a quadratic term. Calling $\Pi(\boldsymbol{x})$ the function to constraint and $\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0}$ the constraint, the Penalty formulation is obtained by adding to the potential the square of the constraint:

$$\Pi(\boldsymbol{x})^{tot} = \Pi(\boldsymbol{x}) + \Pi(\boldsymbol{x})^{penalty} = \Pi(\boldsymbol{x}) + \frac{1}{2}\beta \boldsymbol{h}(\boldsymbol{x})^T \boldsymbol{h}(\boldsymbol{x}) \tag{5.1}$$

where β is the so called *Penalty factor*. In computational mechanics, if we want for example to use the Penalty method to append Dirichlet boundary conditions, the quantity $\Pi(\boldsymbol{x})^{penalty}$ can be written as:

$$\Pi(\boldsymbol{x})^{penalty} = \beta \int_\Gamma (\boldsymbol{u} - \bar{\boldsymbol{u}})^T (\boldsymbol{u} - \bar{\boldsymbol{u}}) \, d\Gamma. \tag{5.2}$$

The idea is that if β is large enough, the minimum of $\Pi(\boldsymbol{x})^{tot}$ cannot be attained without satisfying the constraints [4].
By using the Penalty approach the linearized principle of virtual works gives:

$$(\boldsymbol{K}^{tan} + \boldsymbol{K}^{penalty})\Delta\boldsymbol{u} = -(\boldsymbol{R} + \boldsymbol{R}^{penalty}) \tag{5.3}$$

where

$$\boldsymbol{K}^{penalty} = \beta \int_{\Gamma} \boldsymbol{H}^T \boldsymbol{H} \; d\Gamma$$

and

$$\boldsymbol{R}^{penalty} = \boldsymbol{K}^{tan}(\boldsymbol{u} - \bar{\boldsymbol{u}}).$$

Matrix $\boldsymbol{H}$ is the shape function matrix (see next section).
One of the main drawbacks of this method is the choice of the penalty factor. Sometimes is not possible to tune the parameter in such way to enforce correctly the constraint without affect the conditioning of the matrix. Moreover, due to the nature of the formulation, we are not solving the exact problem but a modified version. For these reasons the Lagrange Multipliers approach is preferred since we are solving exactly the problem. However, as we will see, the Lagrange Multipliers approach enlarge the system of equation and therefore is computationally speaking more expensive. In addition the Lagrange Multipliers method suffers some stability issue as reported in the next section.

5.2 Introduction to Lagrange Multipliers

In this section we are going to introduce the Lagrange Multipliers theory. The Lagrange Multipliers method is one of the main methods for *constrained optimization*. The standard form for a constrained optimization problem is:

find $\boldsymbol{x}^* \in \mathbb{R}^d$ such that $\boldsymbol{x}^* = \text{argmin}(\Pi(\boldsymbol{x}))$ where $\Pi(\boldsymbol{x})$ is subjected to $\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0}$.

In the optimization theory the function $\Pi : \mathbb{R}^d \to \mathbb{R}$ is called **objective function** while $\boldsymbol{h} : \mathbb{R}^d \to \mathbb{R}^m$ is called **constraint function**. Now, we introduce the Lagrange Multipliers λ_i which are as many as the are constraints. Following the previous formalism we can collect the Lagrange Multipliers into a vector $\boldsymbol{\lambda} \in \mathbb{R}^m$ and we are going to call it simply Lagrange Multiplier.

At this point we can define a new function $L : \mathbb{R}^d \times \mathbb{R}^m \rightarrow \mathbb{R}$ called **Lagrangian function** as:

$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = \Pi(\boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{h}(\boldsymbol{x}) = \Pi(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i h_i(\boldsymbol{x}) \tag{5.4}$$

In order to find the value of $\boldsymbol{x}$ minimizing $\Pi(\boldsymbol{x})$ subjected to $\boldsymbol{h}(\boldsymbol{x})$, we have to derive Equation[5.4] with respect to $\boldsymbol{x}$ (also called primal variable) and $\boldsymbol{\lambda}$ and set the result equal to zero. Basically, we have to compute the gradient of the Lagrangian function:

$$\nabla L(\boldsymbol{x}, \boldsymbol{\lambda}) = \left(\frac{\partial L}{\partial \boldsymbol{x}}, \frac{\partial L}{\partial \boldsymbol{\lambda}} \right)^T = \mathbf{0}.$$

Particularly interesting from the point of view of applications is the case of quadratic objective function and linear constraints. In that case we have:

$$\begin{aligned} \Pi(\boldsymbol{x}) &= \tfrac{1}{2}\boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^T \boldsymbol{x} + c \\ \boldsymbol{h}(\boldsymbol{x}) &= \boldsymbol{B}\boldsymbol{x} - \boldsymbol{d} \end{aligned} \tag{5.5}$$

and consequently the Lagrangian function is

$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^T \boldsymbol{x} + c + \boldsymbol{\lambda}^T (\boldsymbol{B}\boldsymbol{x} - \boldsymbol{d}). \tag{5.6}$$

Computing the first derivatives with respect to $\boldsymbol{x}$ and $\boldsymbol{\lambda}$ we obtain

$$\begin{aligned} \frac{\partial L}{\partial \boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} + \boldsymbol{B}^T \boldsymbol{\lambda} = \mathbf{0} \\ \frac{\partial L}{\partial \boldsymbol{\lambda}} &= \boldsymbol{B}\boldsymbol{x} - \boldsymbol{d} = \mathbf{0} \end{aligned}$$

which give the following linear system:

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}^T \\ \boldsymbol{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{b} \\ \boldsymbol{d} \end{bmatrix}. \tag{5.7}$$

As we will see in section 3 some problems arise from the previous linear system.

5.3 Lagrange Multipliers Method in Computational Mechanics

In computational mechanics we can use the Lagrange Multipliers method in order to constrain the system. What we are interested in is to use the Lagrange Multipliers to append boundary conditions. The following discussion is going to be general and it holds both for FEM, MPM and other numerical methods. First of all we need to define the objective and constraint function in order to obtain the Lagrangian and compute its gradient. Our objective function will be the Potential Energy $\Pi(\boldsymbol{u})$ while the constraint function is:

$$\boldsymbol{C}(\boldsymbol{u}) = \boldsymbol{0} \text{ in } \Omega \text{ or on } \Gamma. \tag{5.8}$$

In order to constraint the system we introduce another potential, $\Pi^{LM}(\boldsymbol{u}, \boldsymbol{\lambda})$, defined as

$$\Pi^{LM}(\boldsymbol{u}, \boldsymbol{\lambda}) = \int_\Omega \lambda^T \boldsymbol{C}(\boldsymbol{u}) \, d\Omega \tag{5.9}$$

or

$$\Pi^{LM}(\boldsymbol{u}, \boldsymbol{\lambda}) = \int_\Gamma \lambda^T \boldsymbol{C}(\boldsymbol{u}) \, d\Gamma. \tag{5.10}$$

In our case, since we want to append boundary conditions, the constraint function is

$$\boldsymbol{C}(\boldsymbol{u}) = \boldsymbol{u} - \bar{\boldsymbol{u}} = \boldsymbol{0} \tag{5.11}$$

where $\bar{\boldsymbol{u}}$ is the prescribed displacement. Consequently, the Constraint potential becomes

$$\boxed{\Pi^{LM}(\boldsymbol{u}, \boldsymbol{\lambda}) = \int_\Gamma \boldsymbol{\lambda}^T (\boldsymbol{u} - \bar{\boldsymbol{u}}) \, d\Gamma} \tag{5.12}$$

where $\boldsymbol{\lambda}$ plays the role of the Lagrange Multipliers[1]. We can now define the Lagrangian function by summing the potential energy and the Constraint potential and we obtain

[1] From a physical point of view $\boldsymbol{\lambda}$ has the dimension of a force. This is due to the fact the potential is expressed in terms of energy and since inside the integrand there are also displacements, so a length, $\boldsymbol{\lambda}$ has to be a force. [E]=[F][L].

$$\Pi^{tot}(\boldsymbol{u}, \boldsymbol{\lambda}) = \Pi(\boldsymbol{u}) + \Pi^{LM}(\boldsymbol{u}, \boldsymbol{\lambda}). \tag{5.13}$$

Since our goal is to solve numerically the Cauchy Equations[2.32] we have to compute the first variation of $\Pi^{tot}(\boldsymbol{u}, \boldsymbol{\lambda})$ in direction $\delta\boldsymbol{u}$ and $\delta\boldsymbol{\lambda}$. Hence, we are going to obtain its weak formulation meaning the Principle of Virtual Works. If we do so for the potential energy $\Pi(\boldsymbol{u})$ we end up with the classical virtual works term, $\delta W(\boldsymbol{u}, \delta\boldsymbol{u})$, while if we compute the first variations for the Constraints potential $\Pi^{LM}(\boldsymbol{u}, \boldsymbol{\lambda})$ we obtain the virtual works associated to:

$$D\Pi^{LM}(\boldsymbol{u}, \boldsymbol{\lambda})[\delta\boldsymbol{u}] = \delta W^{LM}(\boldsymbol{u}, \boldsymbol{\lambda}, \delta\boldsymbol{u}), \tag{5.14}$$

$$D\Pi^{LM}(\boldsymbol{u}, \boldsymbol{\lambda})[\delta\boldsymbol{\lambda}] = \delta W^{LM}(\boldsymbol{u}, \boldsymbol{\lambda}, \delta\boldsymbol{\lambda}), \tag{5.15}$$

where

$$\begin{aligned} D\Pi^{LM}(\boldsymbol{u}, \boldsymbol{\lambda})[\delta\boldsymbol{\lambda}] &= \left.\frac{d}{d\epsilon}\right|_{\epsilon=0} \left(\int_\Gamma (\boldsymbol{\lambda} + \epsilon\delta\boldsymbol{\lambda})^T (\boldsymbol{u} - \bar{\boldsymbol{u}})\, d\Gamma \right) = \\ &= \int_\Gamma \delta\boldsymbol{\lambda}^T\, (\boldsymbol{u} - \bar{\boldsymbol{u}})\ d\Gamma, \end{aligned} \tag{5.16}$$

$$D\Pi^{LM}(\boldsymbol{u}, \boldsymbol{\lambda})[\delta\boldsymbol{u}] = \left.\frac{d}{d\epsilon}\right|_{\epsilon=0} \left(\int_\Gamma \boldsymbol{\lambda}(\boldsymbol{u} + \epsilon\delta\boldsymbol{u} - \bar{\boldsymbol{u}})\, d\Gamma \right) = \int_\Gamma \boldsymbol{\lambda}^T \delta\boldsymbol{u}\, d\Gamma. \tag{5.17}$$

The **modified principle of virtual works**, which is the first variation of the Lagrangian function in Equation[5.13], can be written as

$$\delta W^{tot}(\boldsymbol{u}, \boldsymbol{\lambda}, \delta\boldsymbol{u}, \delta\boldsymbol{\lambda}) = \delta W(\boldsymbol{u}, \delta\boldsymbol{u}) + \delta W^{LM}(\boldsymbol{u}, \boldsymbol{\lambda}, \delta\boldsymbol{u}) + \delta W^{LM}(\boldsymbol{u}, \boldsymbol{\lambda}, \delta\boldsymbol{\lambda}). \tag{5.18}$$

Since we want to keep the discussion as general as possible we are going to consider the nonlinear case and next, as specific case, the linear static case. Therefore, the principle of virtual work $\delta W(\boldsymbol{u}, \delta\boldsymbol{u})$ is given by:

$$\delta W(\boldsymbol{u}, \delta\boldsymbol{u}) = \delta W^{dyn}(\boldsymbol{u}, \delta\boldsymbol{u}) + \delta W^{int}(\boldsymbol{u}, \delta\boldsymbol{u}) - \delta W^{ext}(\boldsymbol{u}, \delta\boldsymbol{u}).$$

In the contest of Newton-Raphson procedure if we want to obtain a linear model of the previous equations we have to linearized them. In order to do that we have to compute the directional derivative in directions $\Delta\boldsymbol{u}$ and $\Delta\boldsymbol{\lambda}$. Let us focus our attention on Constraint terms, meaning $D\Pi^{LM}(\boldsymbol{u},\boldsymbol{\lambda})[\delta\boldsymbol{u}]$ and $D\Pi^{LM}(\boldsymbol{u},\boldsymbol{\lambda})[\delta\boldsymbol{\lambda}]$. Under the hypothesis of linear constraints[2], the directional derivatives (that are the second directional derivative of the Constraint potential) are:

$$D^2\Pi^{LM}(\boldsymbol{u},\boldsymbol{\lambda})[\delta\boldsymbol{\lambda},\Delta\boldsymbol{\lambda}] = 0,$$

$$D^2\Pi^{LM}(\boldsymbol{u},\boldsymbol{\lambda})[\delta\boldsymbol{\lambda},\Delta\boldsymbol{u}] = \int_\Gamma \delta\boldsymbol{\lambda}^T \left.\frac{d}{d\epsilon}\right|_{\epsilon=0} (\boldsymbol{u}+\epsilon\Delta\boldsymbol{u}-\bar{\boldsymbol{u}})\, d\,\Gamma = \int_\Gamma \delta\boldsymbol{\lambda}^T\Delta\boldsymbol{u}\; d\Gamma,$$

$$D^2\Pi^{LM}(\boldsymbol{u},\boldsymbol{\lambda})[\delta\boldsymbol{u},\Delta\boldsymbol{\lambda}] = \int_\Gamma \left.\frac{d}{d\epsilon}\right|_{\epsilon=0} (\boldsymbol{\lambda}+\epsilon\Delta\boldsymbol{\lambda})^T\,\delta\boldsymbol{u}\; d\Gamma = \int_\Gamma \Delta\boldsymbol{\lambda}^T\delta\boldsymbol{u}\; d\Gamma,$$

$$D^2\Pi^{LM}(\boldsymbol{u},\boldsymbol{\lambda})[\delta\boldsymbol{u},\Delta\boldsymbol{u}] = 0.$$

The last equation, $D^2\Pi^{LM}(\boldsymbol{u},\boldsymbol{\lambda})[\delta\boldsymbol{u},\Delta\boldsymbol{u}]$, is zero because the constraints are supposed to be linear in $\boldsymbol{u}$.
At this point we can proceed with the FEM discretization of the first and second directional derivatives of the Constraint potential. Let's indicate with $\boldsymbol{H}$ the shape function matrix

$$\boldsymbol{H} = [\boldsymbol{N}_1, \boldsymbol{N}_2, ..., \boldsymbol{N}_N]$$

where in the three dimensional case

$$\boldsymbol{N}_i = \begin{bmatrix} N_i & 0 & 0 \\ 0 & N_i & 0 \\ 0 & 0 & N_i \end{bmatrix}$$

then, the discretized quantities are in the form $\boldsymbol{v}_h = \boldsymbol{H}_v\boldsymbol{v}$. Following this notation we have

[2] Linear constraints means that the function $\boldsymbol{C}(\boldsymbol{u}) = \boldsymbol{0}$ is linear with respect to the displacement $\boldsymbol{u}$

$$D\Pi^{LM}(\boldsymbol{u}_h,\boldsymbol{\lambda}_h)[\delta\boldsymbol{\lambda}_h] = \delta\boldsymbol{\lambda}^T \int_\Gamma \boldsymbol{H}_\lambda^T \boldsymbol{H}_u \; d\Gamma \;\; (\boldsymbol{u} - \bar{\boldsymbol{u}})\,,$$

$$D\Pi^{LM}(\boldsymbol{u}_h,\boldsymbol{\lambda}_h)[\delta\boldsymbol{u}_h] = \delta\boldsymbol{u}^T \int_\Gamma \boldsymbol{H}_u^T \boldsymbol{H}_\lambda \; d\Gamma \; \boldsymbol{\lambda},$$

$$D^2\Pi^{LM}(\boldsymbol{u}_h,\boldsymbol{\lambda}_h)[\delta\boldsymbol{\lambda}_h,\Delta\boldsymbol{\lambda}_h] = 0,$$

$$D^2\Pi^{LM}(\boldsymbol{u}_h,\boldsymbol{\lambda}_h)[\delta\boldsymbol{\lambda}_h,\Delta\boldsymbol{u}_h] = \delta\boldsymbol{\lambda}^T \int_\Gamma \boldsymbol{H}_\lambda^T \boldsymbol{H}_u \; d\Gamma \; \Delta u,$$

$$D^2\Pi^{LM}(\boldsymbol{u}_h,\boldsymbol{\lambda}_h)[\delta\boldsymbol{u}_h,\Delta\boldsymbol{\lambda}_h] = \delta\boldsymbol{u}^T \int_\Gamma \boldsymbol{H}_u^T \boldsymbol{H}_\lambda \; d\Gamma \; \Delta \lambda,$$

$$D^2\Pi^{LM}(\boldsymbol{u}_h,\boldsymbol{\lambda}_h)[\delta\boldsymbol{u}_h,\Delta\boldsymbol{u}_h] = 0.$$

Concerning the standard discretized virtual works $\delta W(\boldsymbol{u}_h,\delta\boldsymbol{u}_h)$ we have already seen in Equation[3.67] that

$$\delta W(\boldsymbol{u}_h,\delta\boldsymbol{u}_h) = \delta\boldsymbol{u}^T\boldsymbol{R}$$

$$D\delta W(\boldsymbol{u}_h,\delta\boldsymbol{u}_h)[\Delta\boldsymbol{u}_h] = \delta\boldsymbol{u}^T\boldsymbol{K}^{tan}\Delta\boldsymbol{u}$$

$$D\delta W(\boldsymbol{u}_h,\delta\boldsymbol{u}_h)[\Delta\boldsymbol{\lambda}_h] = 0$$

where matrix $\boldsymbol{K}^{tan}$ includes also matrix $\boldsymbol{M}$. At this point if we define

$$\boldsymbol{A} = \int_\Gamma \boldsymbol{H}_\lambda^T \boldsymbol{H}_u \; d\Gamma$$

we end up with the following *Saddle-point problem*:

$$\boxed{\begin{bmatrix} \boldsymbol{K}^{tan} & \boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{u} \\ \Delta\boldsymbol{\lambda} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{R} + \boldsymbol{A}^T\boldsymbol{\lambda} \\ \boldsymbol{A}\,(\boldsymbol{u} - \bar{\boldsymbol{u}}) \end{bmatrix}. \qquad (5.19)}$$

In the linear static case we don't have to linearize δW^{tot} but we just need to discretize it. If we proceed with the classical FEM discretization we obtain the following linear block system:

$$\begin{bmatrix} \boldsymbol{K} & \boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}^{ext} \\ \bar{\boldsymbol{u}} \end{bmatrix}. \qquad (5.20)$$

In both the linear and the nonlinear case we have obtained a saddle point problem. It means that when the primal variable is minimum the Lagrange Multiplier is maximum. From a mathematical point of view the solution of this problem exists under certain conditions that have to be satisfied in order to have a well-posed problem.

5.4 Saddle Point Problem

In this section we are going to analyze the saddle point problem from a mathematical point of view, which issues arise from it and the functional spaces involved.
Let us consider two general Hilbert spaces X and M and let us denote their dual[3] spaces with X' and M'. The general saddle-point problem or constrained problem is:

Find $(u, \eta) \in X \times M$ such that:

$$\begin{cases} a(u,v) + b(v,\eta) =< l, v > & \forall v \in X,\ \ l \in X' \\ \\ b(u,\mu) =< \sigma, \mu > & \forall \mu \in M,\ \ \sigma \in M' \end{cases} \tag{5.21}$$

where

$$a(\cdot,\cdot) : X \times X \to \mathbb{R}$$

and

$$b(\cdot,\cdot) : X \times M \to \mathbb{R}$$

are two bilinear forms, while,

$$l(\cdot) : X \to X'$$

and

$$\sigma(\cdot) : M \to M'$$

[3] Given an Hilbert space V the dual space V' is the space of linear and bounded functional F on V: $V' = \{F : V \to \mathbb{R}\ s.t.\ \text{F is linear and bounded}\}$

are two functional and $< \cdot, \cdot >$ denotes the duality between X and X' or M and M'. Here, the duality has been indicated with the so called *crochet* notation, for instance $< l, v >= l(v)$.
It is possible to associate to the bilinear forms $a(\cdot, \cdot)$ and $b(\cdot, \cdot)$ two operators $A : X \to X'$ and $B : X \to M'$ such that:

$$< Aw, v >= a(w, v) \qquad \forall w, v \in X$$

$$< Bv, \mu >= b(v, \mu) \quad \forall v \in X, \mu \in M$$

Now, we can define the *adjoint operator* of B, $B^T : M \to X'$ in such way that

$$< B^T \mu, v >=< Bv, \mu >= b(v, \mu) \; \forall v \in X, \mu \in M.$$

Then, Problem[5.21] becomes:

Find $(u, \eta) \in X \times M$ such that

$$\begin{cases} Au + B^T \eta = l & \text{in } X' \\ Bu = \sigma & \text{in } M' \; . \end{cases} \tag{5.22}$$

Let us now define the following space

$$X^\sigma = \{v \in X \text{ s.t. } b(v, \mu) =< \sigma, \mu > \; \forall \mu \in M\}$$

meaning the space of v function belonging to X and satisfying the condition $b(v, \mu) =< \sigma, \mu >$. In particular, we define the space X^0 as

$$X^0 = \{v \in X \text{ s.t. } b(v, \mu) = 0 \; \forall \mu \in M\} = ker(B)$$

which is exactly the kernel [4] of the operator B. With these definitions we can rewrite Problem[5.22] as:

[4] The definition of kernel of an operator is

$$ker(B) = \{v \in X \text{ s.t. } < Bv, \mu >= b(v, \mu) = Bv = 0 \; \forall \mu \in M\} \, .$$

Find $u \in X^\sigma$ such that

$$a(u,v) =< l, v > \qquad \forall v \in ker(B). \tag{5.23}$$

Let us observe that the space X is the direct sum with X^0 and its orthogonal $(X^0)^\perp$ i.e.

$$X = X^0 \oplus (X^0)^\perp.$$

This will be useful later. Finally, we introduce another space

$$X^0_{polar} = \{g \in X' \text{ s.t. } < g, v >= 0 \, \forall v \in X^0\}.$$

The structure of these spaces is schematized in Fig[5.1]. The blue lines indicate the isomorphic operators.

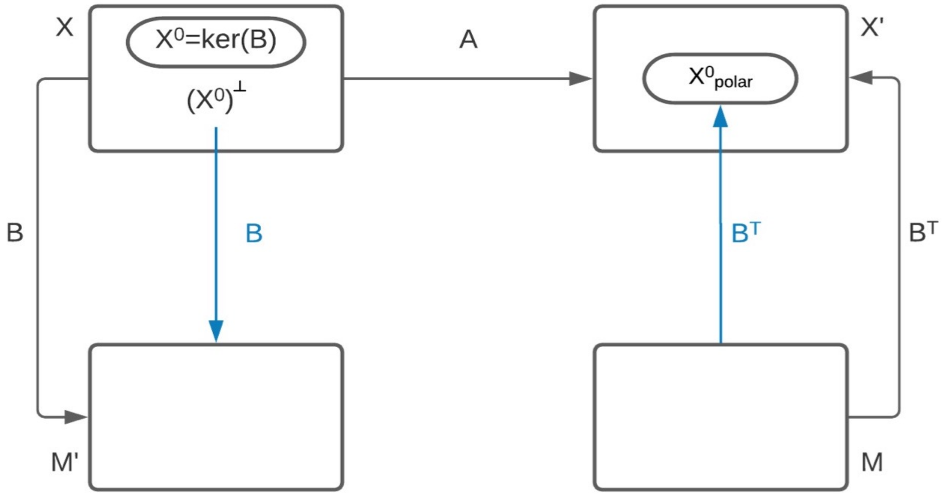

Figure 5.1: The picture shows the spaces X and M, their dual spaces X' and M' and the operators. The blue lines indicate the isomorphism.

From Fig[5.1] we can deduce what follows:

- $B : X \to M'$ is not in an isomorphism since it in not injective i.e. $ker(B) \neq \{0\}$.
- $B^T : M \to X'$ in not an isomorphism since it is not surjective i.e. $Im(B^T) \neq X'$.
- $B : (X^0)^{\perp} \to M'$ is an isomorphism.
- $B^T : M \to X^0_{polar}$ is an isomorphism.

We recall that an isomorphism is any linear application which is bijective and therefore invertible. The invertibility of B is very important. In fact, from Equation[5.22], solving the first equation by u and the second by η we obtain the following formal result

$$\begin{cases} u = A^{-1}\left(l - B^T\eta\right) \\ \\ \eta = \left(BA^{-1}B^T\right)^{-1}\left(BA^{-1}l - \sigma\right) \end{cases} \tag{5.24}$$

If we call R the operator $BA^{-1}B^T$ we will have that

$$R^{-1} = \left(BA^{-1}B^T\right)^{-1} = B^{-T}AB^{-1}$$

and therefore the solution η will exists if R is invertible, meaning if B^{-T} and B are invertible. But B^{-T} and B are invertible if they are bijective i.e. if they are isomorphism. A linear application is bijective if it is both surjective and injective and this last property is achieved if and only if $ker(\cdot) = 0$.
Therefore, we have to introduce an equivalent way to state that B and B^T are isomorphisms:

Lemma. The following properties are equivalent:

- There exists a constant $\beta > 0$ such that

$$\forall \mu \in M \;\; \exists v \in X \text{ with } v \neq 0 \text{ such that } b(u, \mu) \geq \beta ||v||_X ||\mu||_M \tag{5.25}$$

- The operator B^T is an isomorphism between M and X^0_{polar} and the following property holds

$$||B^T \mu||_{X'} = \sup_{v \in V, v \neq 0} \frac{< B^T \mu, v >}{||v||_X} \geq \beta ||\mu||_M \quad \forall \mu \in M \tag{5.26}$$

- The operator B is an isomorphism between $(X^0)^\perp$ and M' and the following property holds

$$||Bv||_{M'} = \sup_{\mu \in M, \mu \neq 0} \frac{< Bv, \mu >}{||v||_M} \geq \beta ||v||_X \quad \forall v \in (X^0)^\perp \tag{5.27}$$

Equation[5.25] is the celebrated **Brezzi-Babuska condition** or **inf-sup condition** and it can be rewritten as

$$\inf_{\mu \in M} \sup_{v \in X} \frac{b(v, \mu)}{||v||_X ||\mu||_M} \geq \beta \tag{5.28}$$

Therefore, Problem[5.21] is well-posed if:

- The bilinear forms $a(\cdot, \cdot)$ and $b(\cdot, \cdot)$ are continuous i.e. there exist two constant c_1 and c_2 greater than zero such that:

$$|a(w, v)| \leq c_1 ||w||_X ||v||_X$$

$$|b(w, \mu)| \leq c_2 ||w||_X ||\mu||_M$$

- There exist a constant c_0 greater than zero such that the bilinear form $a(\cdot, \cdot)$ is *elliptic on the kernel of* B:

$$a(v, v) \geq c_0 ||v||^2_X \quad \forall v \in X^0$$

- There exists a positive constant, $\beta > 0$, such that the BB-condition holds:
$$\inf_{\mu \in M} \sup_{v \in X} \frac{b(v,\mu)}{||v||_X ||\mu||_M} \geq \beta \tag{5.29}$$

Before proceeding any further, let us state an important theorem regarding the functional spaces to be used within the framework of boundary conditions.

Trace Theorem:
Let us consider an open bounded set $\Omega \subset \mathbb{R}^d$ with a sufficiently regular boundary (e.g. Lipschitzian), Γ. Thus, there exists a unique linear application, γ_0,
$$\gamma_0 : H^k(\Omega) \to L^2(\Gamma)$$
such that, given a function $v \in H^k(\Omega)$, with $k \geq 1$, $\gamma_0(v)$ coincides with function v restricted on Γ i.e. $\gamma_0(v) = v|_\Gamma$.

This result gives sense to Dirichlet boundary condition if we interpret the boundary values as trace. Moreover, the space $H^1_0(\Omega)$ can be interpreted as the kernel of the trace operator, meaning:

$$H^1_0(\Omega) = ker(\gamma_0) = \{v \in H^1(\Omega) \text{ s.t. } \gamma_0 v = 0\}. \tag{5.30}$$

Further investigations (see [10]) show that taking the trace of all functions belonging to $H^1(\Omega)$ does not yield the space $L^2(\Gamma)$:

$$\gamma_0(H^1(\Omega)) \subset L^2(\Gamma).$$

In other words, the trace operator, as defined above, is not surjective. In order to make the trace operator surjective, we must restrict the codomain to a specific set denoted as $H^{\frac{1}{2}}(\Gamma)$. In this case, we will have

$$H^{1/2}(\Gamma) = \gamma_0(H^1(\Omega)).$$

Thus, the surjective trace operator will be defined as

$$\gamma_0 : H^1(\Omega) \to H^{\frac{1}{2}}(\Gamma). \tag{5.31}$$

Usually the dual of $H^{\frac{1}{2}}(\Gamma)$ is denoted by $H^{-\frac{1}{2}}(\Gamma)$. Then, the trace theorem allows us to define the functional spaces involved in the Lagrange Multiplier problem.
Once these spaces have been defined, the **mechanical problem** can be stated as:

Find $(\boldsymbol{u},\boldsymbol{\lambda}) \in H^1(\Omega) \times H^{-1/2}(\Gamma)$ such that

$$\begin{cases} a(\boldsymbol{u},\boldsymbol{v}) + b(\boldsymbol{v},\boldsymbol{\lambda}) =< \boldsymbol{f},\boldsymbol{v} > & \forall \boldsymbol{v} \in H^1(\Omega) \\ b(\boldsymbol{u},\boldsymbol{q}) =< \boldsymbol{g},\boldsymbol{q} > & \forall \boldsymbol{q} \in H^{-1/2}(\Gamma) \end{cases} \tag{5.32}$$

Let us now denoting by h_u the length of the sides of the finite element along the boundary, Γ, with which we discretize the primal variable and h_λ the length of the linear elements for the Lagrange Multipliers. It can be shown (see for instance [3], [17]) that the BB-condition demands two different meshes respectively for the primal variable and the Lagrange Multipliers in order to be satisfied. In formula:

$$\frac{h_\lambda}{h_u} \geq C_\Omega \tag{5.33}$$

where C_Ω depends on the domain in some complicate way and it has to be greater than zero $C_\Omega > 1$. Moreover, condition in Equation[5.33] is telling us that the degrees of freedom of the Lagrange Multipliers variables have to be smaller than the degrees of freedom of the primal variable along the boundary where we want to impose boundary conditions. In general, called N the total degrees of freedom of the primal variable and M those of the Lagrange Multiplier one, the condition in Equation[5.28] is satisfied (see Quarteroni[28]) if:

$$N \geq M. \tag{5.34}$$

For example, we could use constant shape function $\mathcal{P}_0$ for the Lagrange Multipliers and linear shape function for the primal variable $\mathcal{P}_1$. However, we should stress that the $\mathcal{P}_0 - \mathcal{P}_1$ choice does not satisfied the BB-condition in general but only if the Lagrange Multipliers degrees of freedom are defined along the boundary (or interface) or just in few nodes. In order to understand better, let us consider FIG[5.2]. The figure shows two different cases where the BB condition is or is not satisfied. On the left we can see that the

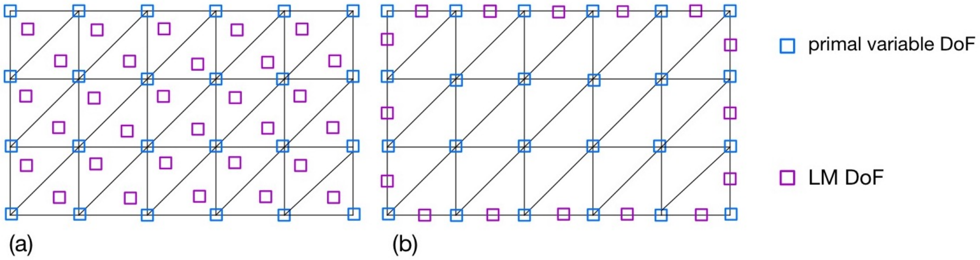

Figure 5.2: The figure shows the degrees of freedom of the primal and Lagrange Multipliers variables. (a): The BB-condition is not satisfied; the Lagrange Multipliers are defined in the entire domain. (b): the BB-condition is satisfied and the Lagrange Multipliers are defined just along the boundary.

Lagrange Multipliers degrees of freedom are defined among the entire domain. If we count them we will realize that $N = 48$ DoF while $M = 60$ DoF . Thus, condition [5.34] is not satisfied. Nevertheless, if we look at the figure on the right we have $N = 48$ DoF and $M = 32$ DoF . Thus, the condition $N \geq M$ is satisfied. For other similar examples, the reader is encouraged to consult [28].
This simple example shows how we must take special care in choosing the basic functions.

5.5 MPM Lagrange Multipliers Implementation

In many engineering applications, bodies and their material boundaries face large displacements and deformations. As a consequence, if we use some particles methods such as the MPM, we have to deal very often with *non-conforming* boundary conditions meaning that the material boundaries do not coincide with the background mesh (FIG[5.3]).

In this section we are going to analyze the implementation of the Lagrange Multipliers by using *boundary particles*. Boundary particles are mass-less particles used to track the deformation of the material boundaries FIG[5.4]. These particles can be fixed or can move independently from the motion.
For each background element containing at least one boundary particles we are going to define on it one Lagrange Multiplier degree of freedom. Since

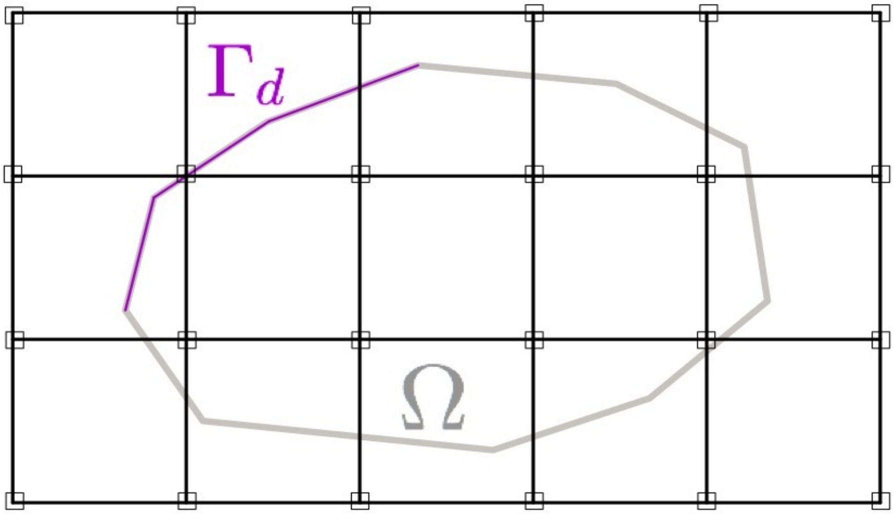

Figure 5.3: The figure shows how the boundary where we want to impose, for instance, the Dirichlet conditions (purple line) does not coincide with the boundary of the background mesh.

we want to use constant basis functions, it does not matter where we define this degree of freedom, for instance, we could define it in the center of the element.
As pointed out in the previous section, we have to satisfy the BB-condition. This condition is satisfied if the degrees of freedom of the Lagrange Multipliers, M, are sufficiently smaller than the primal variable ones, N. However, it is not always easy to evaluate a priori which is the ideal number of Lagrange Multipliers degrees of freedom (or number of boundary particles). Nevertheless, if we adopt $\mathcal{P}_0 - \mathcal{P}_1$ (FIG[5.6]) basis functions we should not worry as just few Lagrange Multipliers degree of freedom are defined. In the present work we set an arbitrary number of boundary particles and for each element containing at least one of these mass-less particles, we define a Lagrange Multiplier degree of freedom is defined. This procedure is schematized in FIG[5.5].

Once, we perform the $\mathcal{P}_0 - \mathcal{P}_1$ discretization we obtain Equation[5.19].

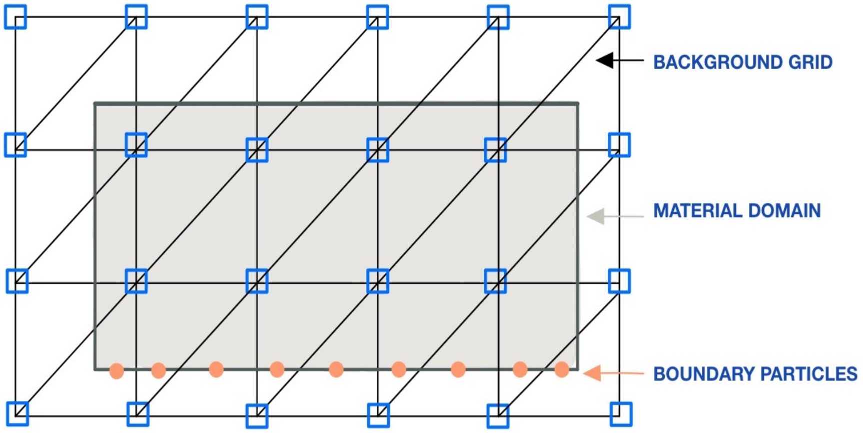

Figure 5.4: The figure shows some boundary particles.

5.6 Roller Boundary Conditions

When we deal with roller boundary with an arbitrary inclination, we have to introduce an appropriate rotation matrix $\mathcal{Q}$ that we use to rotate the involved degrees of freedom such as displacement and Lagrange Multiplier. Once we rotate the system accordingly to the roller inclination we can solve the rotated linear system, indicated with $\boldsymbol{K}^{\theta}\boldsymbol{u}^{\theta} = \boldsymbol{f}^{\theta}$. The quantities $(\#)^{\theta}$ represent the their own description in the axis-aligned configuration. Once we solve the rotated linear system we have to rotate back the solution to the original configuration. This is achieved via the inverse of the rotation matrix [5]. For instance, the displacement in its original configuration is $\boldsymbol{u} = \boldsymbol{Q}^{T}\boldsymbol{u}^{\theta}$. As the usual boundary conditions also the roller boundary can be conforming or nonconforming.

5.6.1 Roller Boundary Condition for FEM

We start the discussion about roller boundaries within the Finite Element framework since it is easier than Material Point Method. Let us consider the case of Roller Boundary Conditions with an arbitrary inclinations. In order

[5] The rotation matrix is usually orthogonal so the inverse matrix coincides with the transposed.

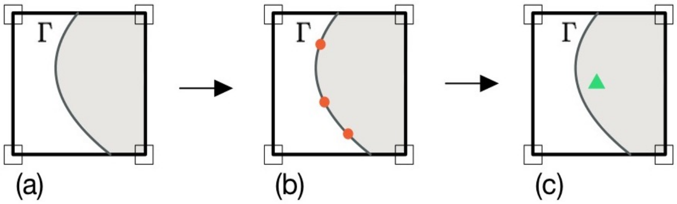

Figure 5.5: Lagrange Multiplier DoF definition sequence. (a) material domain where we want to impose nonconforming boundary conditions (b) boundary particles (in orange) defined along the boundary (c) Lagrange Multiplier DoF (green triangle).

to impose these boundary conditions let us consider an orthogonal rotation matrix $\boldsymbol{Q}$ defined in three-d as follows

$$\boldsymbol{Q} = \begin{bmatrix} \hat{n}_x & \hat{n}_y & \hat{n}_z \\ \hat{t}_x & \hat{t}_y & \hat{t}_z \\ \hat{q}_x & \hat{q}_y & \hat{q}_z \end{bmatrix} \tag{5.35}$$

where $\hat{\boldsymbol{n}}$ is the normal unit vector perpendicular to the inclined surface and $\hat{\boldsymbol{t}}$,$\hat{\boldsymbol{q}}$ are the unit vector parallel to the surface. Each constrained degrees of freedom has its own normal and parallel unit vectors since the inclination can change. For instance, in FIG[5.7], node 1 has different normal and parallel unit vectors than node 2.

By using matrix $\boldsymbol{Q}$ we are able to obtain the prescribed axis-aligned displacement, meaning

$$\boldsymbol{u}^{\theta} = \boldsymbol{Q}\boldsymbol{u}. \tag{5.36}$$

We have to proceed in this way for every constrained degrees of freedom. Moreover, we should also apply the rotation matrix to those entries in the tangential stiffness matrix corresponding to the constrained degrees of freedom. Let us consider again FIG[5.7]. The stiffness matrix inherent to the triangular element will be a 6 by 6 matrix. In particular we have

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}_{1,1} & \boldsymbol{K}_{1,2} & \boldsymbol{K}_{1,3} \\ \boldsymbol{K}_{2,1} & \boldsymbol{K}_{2,2} & \boldsymbol{K}_{2,3} \\ \boldsymbol{K}_{3,1} & \boldsymbol{K}_{3,2} & \boldsymbol{K}_{3,3} \end{bmatrix} \tag{5.37}$$

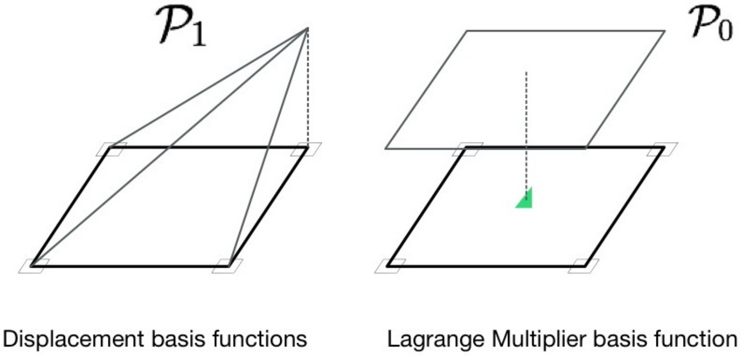
$\mathcal{P}_1$
$\mathcal{P}_0$
Displacement basis functions
Lagrange Multiplier basis function

Figure 5.6: Example of basis functions for a quadrilateral background mesh. It has been used $\mathcal{P}_1$ for primal variables and $\mathcal{P}_0$ for LM variables.

where each $\boldsymbol{K}_{i,j}$ is a 2 by 2 matrix. The local linear system associate to this element is $\boldsymbol{Ku} = \boldsymbol{f}$. Let us focus the attention in vector $\boldsymbol{u}$ whose components are: $\boldsymbol{u} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3]^T$. The constrained nodes are node 1 and node 2, therefore we should apply the rotation matrix to these two nodes. That means:

$$\boldsymbol{u}^\theta = \left[\begin{array}{ccc} \boldsymbol{Q}_1\boldsymbol{u}_1, & \boldsymbol{Q}_2\boldsymbol{u}_2, & \boldsymbol{u}_3 \end{array} \right]. \tag{5.38}$$

This is equivalent to define a new matrix $\mathcal{Q}$ and multiply the linear system by this matrix. In this specific case $\mathcal{Q}$ is

$$\boldsymbol{\mathcal{Q}} = \begin{bmatrix} \boldsymbol{Q}_1 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}_2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}. \tag{5.39}$$

The linear system $\boldsymbol{Ku} = \boldsymbol{f}$ is transformed into a new system $\boldsymbol{K}^\theta\boldsymbol{u}^\theta = \boldsymbol{f}^\theta$ by means of matrix $\boldsymbol{\mathcal{Q}}$ through the transformation $\boldsymbol{\mathcal{Q}K\mathcal{Q}}^T\boldsymbol{\mathcal{Q}u} = \boldsymbol{\mathcal{Q}f}$. In particular we have:

$$\boldsymbol{K}^\theta = \begin{bmatrix} \boldsymbol{Q}_1 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}_2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{K}_{1,1} & \boldsymbol{K}_{1,2} & \boldsymbol{K}_{1,3} \\ \boldsymbol{K}_{2,1} & \boldsymbol{K}_{2,2} & \boldsymbol{K}_{2,3} \\ \boldsymbol{K}_{3,1} & \boldsymbol{K}_{3,2} & \boldsymbol{K}_{3,3} \end{bmatrix} \begin{bmatrix} \boldsymbol{Q}_1^T & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}_2^T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} = \tag{5.40}$$

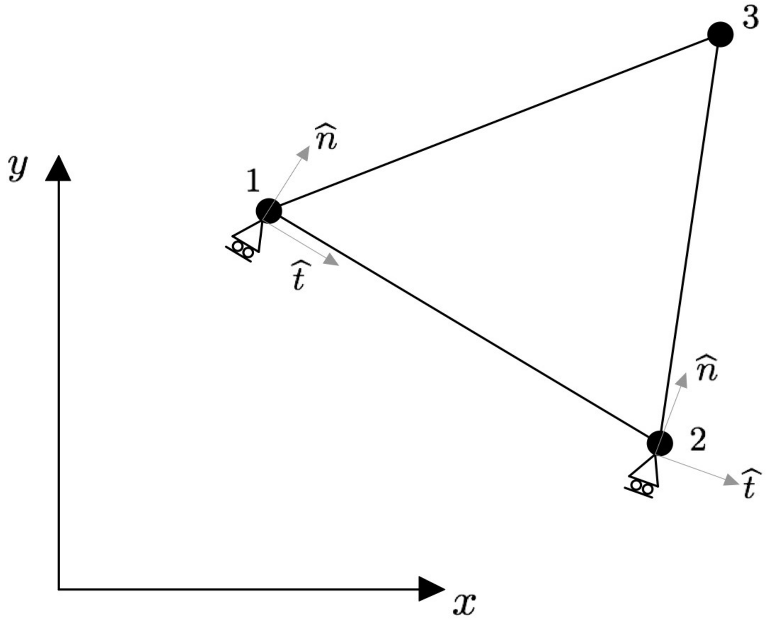

Figure 5.7: Constrained triangular element. The element has been constrained in node 1 and 2 by a roller condition. These two nodes have different normal and parallel unit vectors.

$$= \begin{bmatrix} \boldsymbol{Q}_1\boldsymbol{K}_{1,1}\boldsymbol{Q}_1^T & \boldsymbol{Q}_1\boldsymbol{K}_{1,2}\boldsymbol{Q}_2^T & \boldsymbol{Q}_1\boldsymbol{K}_{1,3} \\ \boldsymbol{Q}_2\boldsymbol{K}_{2,1}\boldsymbol{Q}_1^T & \boldsymbol{Q}_2\boldsymbol{K}_{2,2}\boldsymbol{Q}_2^T & \boldsymbol{Q}_2\boldsymbol{K}_{2,3} \\ \boldsymbol{K}_{3,1}\boldsymbol{Q}_1^T & \boldsymbol{K}_{3,2}\boldsymbol{Q}_2^T & \boldsymbol{K}_{3,3} \end{bmatrix}$$

$$\boldsymbol{u}^\theta = \begin{bmatrix} \boldsymbol{Q}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Q}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \boldsymbol{u}_3 \end{bmatrix} = \begin{bmatrix} \boldsymbol{Q}_1\boldsymbol{u}_1 \\ \boldsymbol{Q}_2\boldsymbol{u}_2 \\ \boldsymbol{u}_3 \end{bmatrix} \tag{5.41}$$

and

$$\boldsymbol{f}^\theta = \begin{bmatrix} \boldsymbol{Q}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Q}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ \boldsymbol{f}_3 \end{bmatrix} = \begin{bmatrix} \boldsymbol{Q}_1\boldsymbol{f}_1 \\ \boldsymbol{Q}_2\boldsymbol{f}_2 \\ \boldsymbol{u}_3 \end{bmatrix}. \tag{5.42}$$

We should observe that the terms inherent to the unconstrained node, node 3, have no been affected by the rotation matrix. This is correct since no constraint is acting on node 3.
Imagine now to have a grid with an arbitrary number of elements and constraints. We should performing what has been done so far for each node and then assembly the elemental contributions in order to get the global stiffness matrix. At this point we can adopt different numbering strategies. For example, as commonly done in FEM, the constrained degrees of freedom are ordered such that they are last. In this way the local stiffness matrix is structured as follows

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}_{\boldsymbol{uu}} & \boldsymbol{K}_{\boldsymbol{uu}_{\Gamma_D}} \\ \boldsymbol{K}_{\boldsymbol{u}_{\Gamma_D}\boldsymbol{u}} & \boldsymbol{K}_{\boldsymbol{u}_{\Gamma_D}\boldsymbol{u}_{\Gamma_D}} \end{bmatrix} \tag{5.43}$$

and matrix $\mathcal{Q}$ is

$$\mathcal{Q} = \begin{bmatrix} \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Q} \end{bmatrix} \tag{5.44}$$

where $\boldsymbol{Q}$ is a block diagonal matrix whose entries are matrices similar to Equation[5.35]. Therefore, the structure of matrix $\mathcal{Q}$ depends on the numbering of the mesh.

We now discuss the imposition of roller conditions in the context of Lagrange multipliers. The system to be solved is a block system. As seen above, the result matrix consists of the tangent stiffness matrix, $\boldsymbol{K}$ and the shape functions matrix integrated on the boundary, $\boldsymbol{A}$. The system resulting is shown below

$$\begin{bmatrix} \boldsymbol{K} & \boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{u} \\ \Delta\boldsymbol{\lambda} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{G} \end{bmatrix} \tag{5.45}$$

where for the sake of shortness we have indicate $\boldsymbol{R}+\boldsymbol{A}^T\boldsymbol{\lambda} = \boldsymbol{F}$ and $\boldsymbol{A}\left(\boldsymbol{u} - \bar{\boldsymbol{u}}\right) = \boldsymbol{G}$. We will refer to this block linear system in the following compact way:

$$\boldsymbol{\mathcal{K}}\,\boldsymbol{\Delta U} = \boldsymbol{\mathcal{F}}.$$

In Lagrange multiplier framework we have to define two total rotation matrices, one for the displacement and the other for the Lagrange Multiplier degrees of freedom. In fact, as written above, the solution vector is given by

$$\boldsymbol{\Delta U} = \left(\boldsymbol{\Delta u},\ \boldsymbol{\Delta\lambda}\right)^T.$$

Taking as studying case the element in FIG[5.7] we will have six degrees of freedom for displacement ad four for the Lagrange Multipliers meaning

$$\boldsymbol{\Delta U}_{10\times1} = \left(\boldsymbol{\Delta u}_{6\times1},\ \boldsymbol{\Delta\lambda}_{4\times1}\right)^T.$$

Therefore, in order to rotate vector $\boldsymbol{\Delta U}$ we have to select two different rotation matrices. The rotation matrix for the displacement is exactly equal to Equation[5.39] while the rotation matrix for the Lagrange Multipliers is similar to the rotation matrix for the displacement but without the identity term. In formula:

$$\boldsymbol{\mathcal{Q}}_u = \begin{bmatrix} \boldsymbol{Q}_1 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}_2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}_{6\times6} \qquad \boldsymbol{\mathcal{Q}}_\lambda = \begin{bmatrix} \boldsymbol{Q}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}_2 \end{bmatrix}_{4\times4}. \tag{5.46}$$

The total rotation matrix is made by these two contributes

$$\boldsymbol{\mathcal{Q}} = \begin{bmatrix} \boldsymbol{\mathcal{Q}}_u & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\mathcal{Q}}_\lambda \end{bmatrix}_{10\times10} \tag{5.47}$$

Thus, the rotated block linear system is:

$$\boldsymbol{\mathcal{Q}}\boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{Q}}^T\,\boldsymbol{\mathcal{Q}}\boldsymbol{\Delta U} = \boldsymbol{\mathcal{Q}}\boldsymbol{\mathcal{F}} \Rightarrow \boldsymbol{\mathcal{K}}^\theta\boldsymbol{\Delta U}^\theta = \boldsymbol{\mathcal{F}}^\theta \tag{5.48}$$

where the rotated matrix $\boldsymbol{\mathcal{K}}^{\theta}$ is

$$\boldsymbol{\mathcal{K}}^{\theta} = \begin{bmatrix} \boldsymbol{\mathcal{Q}}_{u,6\times6}\boldsymbol{K}_{6\times6}\boldsymbol{\mathcal{Q}}^{T}_{u,6\times6} & \boldsymbol{\mathcal{Q}}_{u,6\times6}\boldsymbol{A}^{T}_{6\times4}\boldsymbol{\mathcal{Q}}^{T}_{\lambda,4\times4} \\ \\ \boldsymbol{\mathcal{Q}}_{\lambda,4\times4}\boldsymbol{A}_{4\times6}\boldsymbol{\mathcal{Q}}^{T}_{u,6\times6} & \boldsymbol{0}_{4\times4} \end{bmatrix} = \tag{5.49}$$

$$\begin{bmatrix} \boldsymbol{K}^{\theta} & \boldsymbol{\mathcal{Q}}_{u,6\times6}\boldsymbol{A}^{T}_{6\times4}\boldsymbol{\mathcal{Q}}^{T}_{\lambda,4\times4} \\ \\ \boldsymbol{\mathcal{Q}}_{\lambda,4\times4}\boldsymbol{A}_{4\times6}\boldsymbol{\mathcal{Q}}^{T}_{u,6\times6} & \boldsymbol{0}_{4\times4} \end{bmatrix}$$

while vectors $\boldsymbol{\Delta U}^{\theta}$ and $\boldsymbol{\mathcal{F}}^{\theta}$ are straightforward.

5.6.2 Roller Boundary Condition for MPM

When we deal with MPM things are usually more complicated due to the mapping phases. For the purpose of the explanation we will consider the roller constraints only acting in the normal direction. Different strategies are possible to employ Lagrange Multipliers in the Material Point Method. The one we will describe below is the one implemented in the KratosMultiphysics framework.

For each element containing boundary particles we define a Lagrange Multiplier degree of freedom so that every element with boundary particles will have a degree of freedom associated with Lagrange multipliers. We observe that, using constant basis functions for Lagrange multipliers, it is indifferent where we define this degree of freedom. This degree of freedom will have a normal given by the average of the normals of all the boundary particles. Next, we have to map the normal vector to the background node by means of basis functions. In this way we obtain a resulting normal vector defined on each background node. The resulting vector will be used to rotate the degrees of freedom belonging to those elements which contain at least one boundary particle. Their neighbour elements with no boundary particles will have degrees of freedom which are only partially rotated. All the other nodes will be not rotated at all.

Now, we analyze how the roller boundaries act on the local contribution of an element who contains the boundary particles FIG[5.9]. First of all, we need to map the normal vector of each boundary particles to the background node. Once we do that, we obtain the resulting normal vector that we can use to rotate the various degrees of freedom. In this element we are going

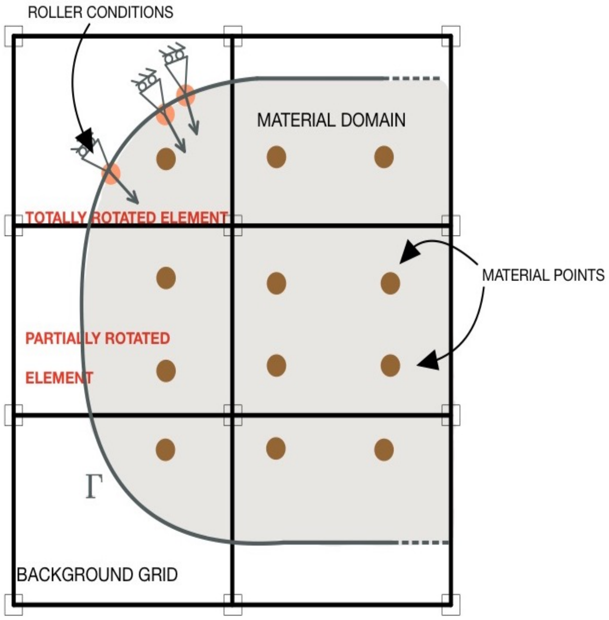

Figure 5.8: MPM constrained body example. The roller constraints are only acting in the normal direction. In orange we find the boundary particles while in brown the material points. It has been highlight the element totally rotated and one partially rotated.

to define the Lagrange Multiplier degrees of freedom whose normal vector is given by:

$$\hat{\boldsymbol{n}}_\lambda = \frac{\hat{\boldsymbol{n}}_A + \hat{\boldsymbol{n}}_B + \hat{\boldsymbol{n}}_C}{3}.$$

This *mean normal vector* is used to create the Lagrange Multipliers rotation matrix, $\mathcal{Q}_\lambda = \mathcal{Q}_\lambda(\hat{\boldsymbol{n}}_\lambda)$.
Taking as example FIG[5.9], the rotation matrix acting on the local tangent stiffness matrix is

$$\boldsymbol{\mathcal{Q}} = \begin{pmatrix} \boldsymbol{Q}_1 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}_2 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{Q}_3 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{Q}_4 \end{pmatrix}. \tag{5.50}$$

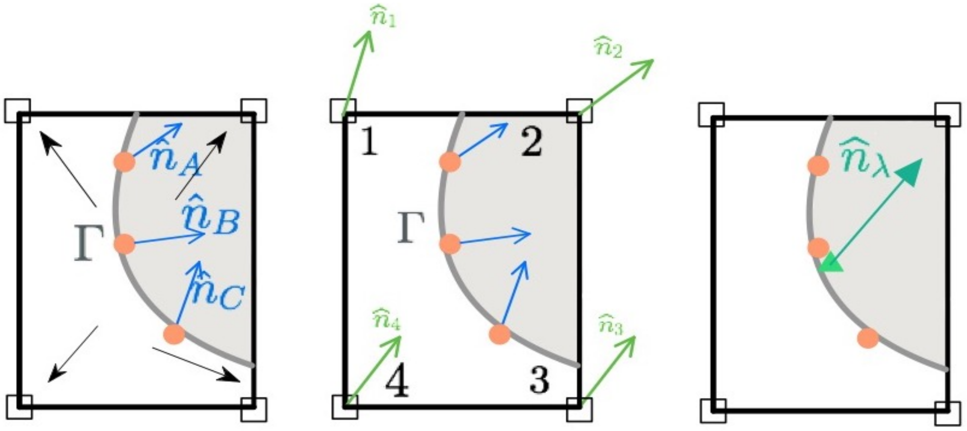

Figure 5.9: The figure shows on left the mapping phase of the normal vectors (blue) and at the center the resulting normal vectors (green) and on the right the Lagrange Multiplier degrees of freedom and its normal, $\hat{\boldsymbol{n}}_\lambda$. The numbering of the node has to be consider as local.

Therefore, we have that the rotated local linear system, without considering the Lagrange Multiplier degree of freedom, is

$$\boldsymbol{\mathcal{Q}} \boldsymbol{K} \boldsymbol{\mathcal{Q}}^T \, \boldsymbol{\mathcal{Q}} \Delta \boldsymbol{u} = \boldsymbol{\mathcal{Q}} \boldsymbol{f}$$

where

$$\boldsymbol{K}^{\theta} = \boldsymbol{\mathcal{Q}}\boldsymbol{K}\boldsymbol{\mathcal{Q}}^{T} = \begin{pmatrix} \boldsymbol{Q}_1 K_{1,1}\boldsymbol{Q}_1^T & \boldsymbol{Q}_1 K_{1,2}\boldsymbol{Q}_2^T & \boldsymbol{Q}_1 K_{1,3}\boldsymbol{Q}_3^T & \boldsymbol{Q}_1 K_{1,4}\boldsymbol{Q}_4^T \\ \boldsymbol{Q}_2 K_{2,1}\boldsymbol{Q}_1^T & \boldsymbol{Q}_2 K_{2,2}\boldsymbol{Q}_2^T & \boldsymbol{Q}_2 K_{2,3}\boldsymbol{Q}_3^T & \boldsymbol{Q}_2 K_{2,4}\boldsymbol{Q}_4^T \\ \boldsymbol{Q}_3 K_{3,1}\boldsymbol{Q}_1^T & \boldsymbol{Q}_3 K_{3,2}\boldsymbol{Q}_2^T & \boldsymbol{Q}_3 K_{3,3}\boldsymbol{Q}_3^T & \boldsymbol{Q}_3 K_{3,4}\boldsymbol{Q}_4^T \\ \boldsymbol{Q}_4 K_{4,1}\boldsymbol{Q}_1^T & \boldsymbol{Q}_4 K_{4,2}\boldsymbol{Q}_2^T & \boldsymbol{Q}_4 K_{4,3}\boldsymbol{Q}_3^T & \boldsymbol{Q}_4 K_{4,4}\boldsymbol{Q}_4^T \end{pmatrix} \tag{5.51}$$

$$\boldsymbol{\Delta u}^{\theta} = \boldsymbol{\mathcal{Q}}\boldsymbol{\Delta u} = \begin{pmatrix} \boldsymbol{Q}_{\mathbf{1}}\boldsymbol{\Delta u}_1 \\ \boldsymbol{Q}_2\boldsymbol{\Delta u}_2 \\ \boldsymbol{Q}_3\boldsymbol{\Delta u}_3 \\ \boldsymbol{Q}_4\boldsymbol{\Delta u}_4 \end{pmatrix} \tag{5.52}$$

and

$$\boldsymbol{f}^{\theta} = \boldsymbol{\mathcal{Q}}\boldsymbol{f} = \begin{pmatrix} \boldsymbol{Q}_{\mathbf{1}}\boldsymbol{f}_1 \\ \boldsymbol{Q}_2\boldsymbol{f}_2 \\ \boldsymbol{Q}_3\boldsymbol{f}_3 \\ \boldsymbol{Q}_4\boldsymbol{f}_4 \end{pmatrix}. \tag{5.53}$$

In Lagrange Multipliers framework we have to define two rotation matrices as we saw earlier in FEM. The rotation matrix for the Lagrange Multipliers is $\boldsymbol{\mathcal{Q}}_{\lambda}$ while the rotation matrix for the displacement is the same of Equation[5.50]. Thus, proceeding as done in Equation[5.49] we obtain:

$$\boldsymbol{\mathcal{K}}^{\theta} = \begin{bmatrix} \boldsymbol{\mathcal{Q}}_u\boldsymbol{K}\boldsymbol{\mathcal{Q}}_u^T & \boldsymbol{\mathcal{Q}}_u\boldsymbol{A}^T\boldsymbol{\mathcal{Q}}_{\lambda}^T \\ \boldsymbol{\mathcal{Q}}_{\lambda}\boldsymbol{A}\boldsymbol{\mathcal{Q}}_u^T & \mathbf{0} \end{bmatrix}, \tag{5.54}$$

$$\boldsymbol{\Delta U}^{\theta} = \begin{pmatrix} \boldsymbol{\mathcal{Q}}_u & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mathcal{Q}}_{\lambda} \end{pmatrix} \begin{pmatrix} \boldsymbol{\Delta u} \\ \boldsymbol{\Delta \lambda} \end{pmatrix}, \tag{5.55}$$

and

$$\boldsymbol{\mathcal{F}}^{\theta} = \begin{pmatrix} \boldsymbol{\mathcal{Q}}_u & \mathbf{0} \\ & \\ \mathbf{0} & \boldsymbol{\mathcal{Q}}_\lambda \end{pmatrix} \begin{pmatrix} \boldsymbol{F} \\ \\ \boldsymbol{G} \end{pmatrix}. \tag{5.56}$$

Let us consider now the case where the element does not contain any boundary particle, but the one next to it does FIG[5.10]. In this scenario we will have that the degrees of freedom are partially rotated. This is due to the fact that only the background nodes shared with the element containing a boundary particle will contribute to the rotation. In this case the rotation matrix is

$$\boldsymbol{\mathcal{Q}} = \begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{Q}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{Q}_4 \end{pmatrix} \tag{5.57}$$

and the rotated local tangent stiffness matrix will have some entries which are not rotated:

$$\boldsymbol{K}^{\theta} = \begin{pmatrix} \boldsymbol{K}_{1,1} & \boldsymbol{K}_{1,2} & \boldsymbol{K}_{1,3}\boldsymbol{Q}_3^T & \boldsymbol{K}_{1,4}\boldsymbol{Q}_4^T \\ & & & \\ \boldsymbol{K}_{2,1} & \boldsymbol{K}_{2,2} & \boldsymbol{K}_{2,3}\boldsymbol{Q}_3^T & \boldsymbol{K}_{2,4}\boldsymbol{Q}_4^T \\ & & & \\ \boldsymbol{Q}_3\boldsymbol{K}_{3,1} & \boldsymbol{Q}_3\boldsymbol{K}_{3,2} & \boldsymbol{Q}_3\boldsymbol{K}_{3,3}\boldsymbol{Q}_3^T & \boldsymbol{Q}_3\boldsymbol{K}_{3,4}\boldsymbol{Q}_4^T \\ & & & \\ \boldsymbol{Q}_4\boldsymbol{K}_{4,1} & \boldsymbol{Q}_4\boldsymbol{K}_{4,2} & \boldsymbol{Q}_4\boldsymbol{K}_{4,3}\boldsymbol{Q}_3^T & \boldsymbol{Q}_4\boldsymbol{K}_{4,4}\boldsymbol{Q}_4^T \end{pmatrix}. \tag{5.58}$$

Regarding the rotation of the degrees of freedom of the Lagrange Multipliers, no rotation needs to be done since there are none within the element.

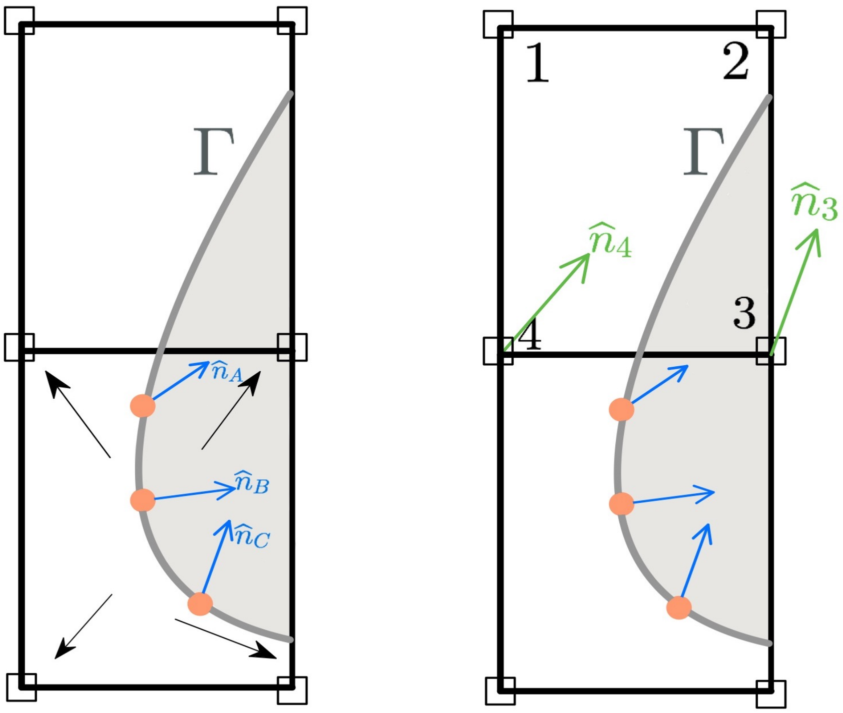

Figure 5.10: The upper element does not contain any boundary particle but the below does. Even in this case the numbering is local.

Chapter 6

Numerical Applications

In this chapter we are going to show some classical benchmark tests where we will impose nonconforming boundary conditions. Specifically, we will consider hyperelastic cantilever beam and we are going to compare the vertical displacement obtained by means of Lagrange Multipliers and Penalty approach. We expect that the Lagrange Multiplier approach leads to a smaller absolute error since we are appending exactly the boundary conditions. Then, we will consider also a clamped beam only by using the Lagrange Multipliers and we will check the quadratic convergence by computing the relative and absolute errors.

Next, we are going to simulate a granular flow obtained from a collapsing soil column. In such example FEM is not adequate due to the extreme large displacements involved. MPM is a good alternative to simulate this kind of problems. The resulting deformed configuration is compared with that one obtained by Bui et al. [11].

6.1 Hyperelastic Cantilever Beam

Let us consider hyperelastic neo-Hookian beam with square cross-section having as initial parameters: density $\rho = 1000\ kg/m^3$, Young's modulus $E = 90\ MPa$ and Poisson's ration $\nu = 0$. The beam has a length, $L = 8\ m$ and the sectional area is $A = 1\text{x}1\ m^2$. In order to consider the most general case the background grid had been rotated ($\theta = 36°$) to simulate the roller boundary condition.

The convergence study was carried out considering as a reference value for the vertical displacement the one indicated by Timoshenko's theory:

$$\delta_A^{ref} = -\left(\frac{\rho g(bhL)L^3}{8EI} + \frac{\rho g L^2}{2GA_s}\right) \tag{6.1}$$

where g is the gravity acceleration $g = 9.81\, m/s^2$; I is the inertia of the beam section, $I = \dfrac{bh^3}{12}$; A_s is the reduced cross section and G is the shear modulus equal to $G = \dfrac{E}{2(1+\nu)}$. The reference value is computed at point A (see FIG[6.1]).

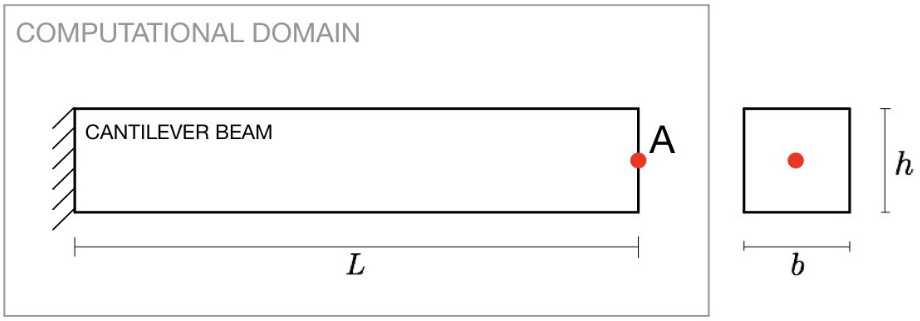

Figure 6.1: The figure shows the computation domain and the body, meaning the cantilever beam, with the geometrical information.

The absolute error has been calculated according to the following formula:

$$e_a = |\delta_A - \delta_A^{ref}| \tag{6.2}$$

where δ_A is the vertical displacement of point A (see FIG[6.1]).

FIG[6.2] shows that the error we obtain with the Lagrange Multipliers method is smaller than the one obtained with the Penalty method. This is due to the fact that the Lagrange Multiplier method is exact while Penalty is conditioned by the penalty factor which is added to the potential (see Equation[5.1]) in order to append the boundary conditions. The numerical solution obtained with Penalty, is therefore, conditioned by the factor, β.

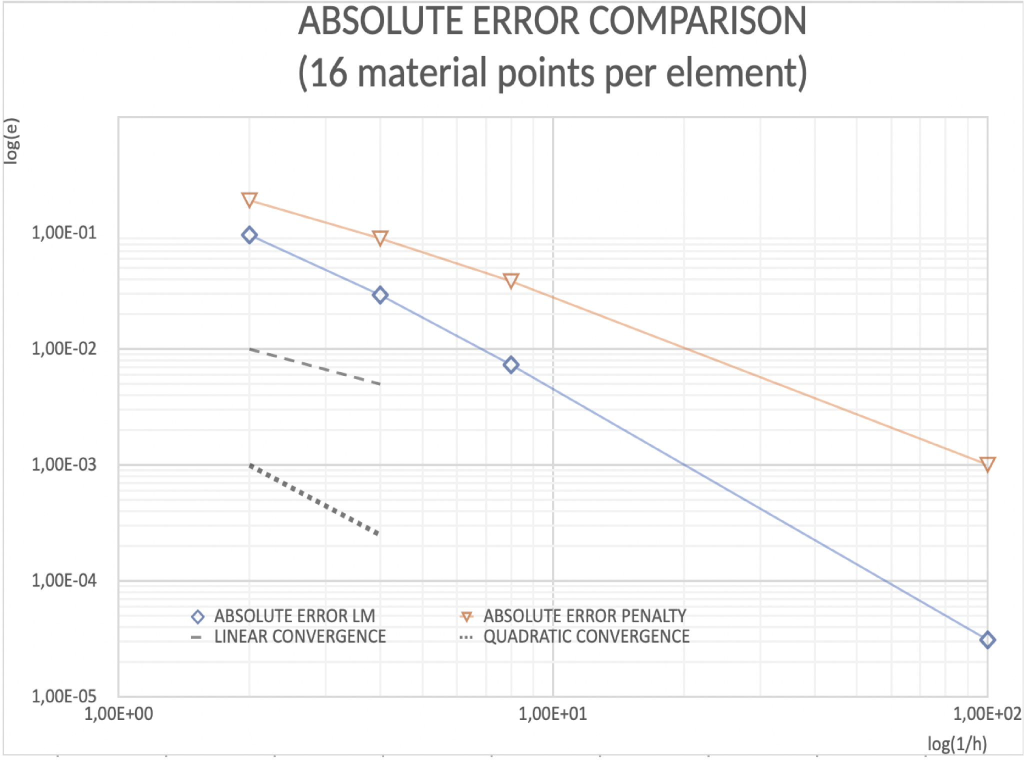

Figure 6.2: Absolute error obtained by means of Lagrange Multipliers (in blue) and Penalty methods (in orange).

For the Lagrange multipliers method this does not happen because the additional degrees of freedom are used to make explicit the boundary conditions and the constraints.

Regarding the convergence, we can observe that the Lagrange Multipliers approach reaches a quadratic convergence even for coarse computational meshes while the Penalty method only reaches the quadratic convergence for fine mesh size (see Chandra et al. [14]).

One of the main disadvantages of the Lagrange Multipliers method is the computational cost. Since we have added the degrees of freedom of the multipliers, the computational cost for the resolution of the linear system is greater than the computational cost of the Penalty method. This can be appreciated looking at the system of Equation[5.19].

6.2 Clamped Beam

In this section we are going to analyze a clamped beam in two different cases according to different boundary conditions. In both cases the right side of the beam is treated as non-conforming (see FIG[6.3]) and on this side we are going to use the Lagrange Multiplier to enforce the boundary conditions. We first consider a double-clamped beam (CASE 1) as shown in FIG[6.4].

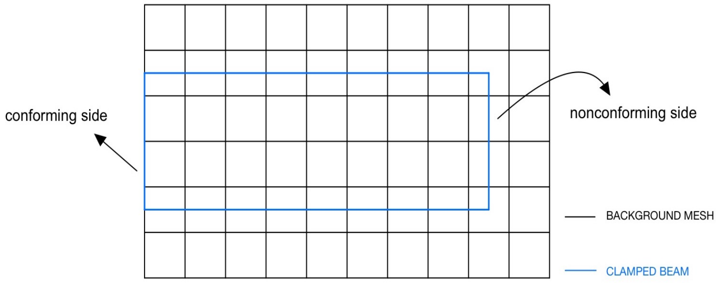

Figure 6.3: The figure schematizes the conforming and non-conforming sides.

and as a second case (CASE 2), we release the vertical displacements on the right edge as show in FIG[6.5].

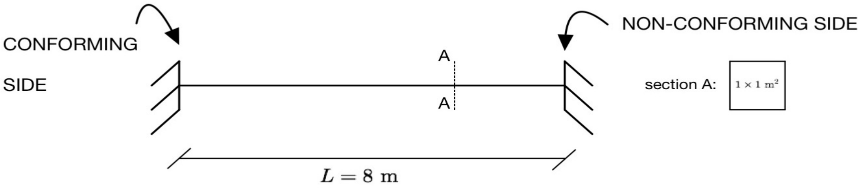

Figure 6.4: Schematic representation of the double-clamped beam (CASE 1).

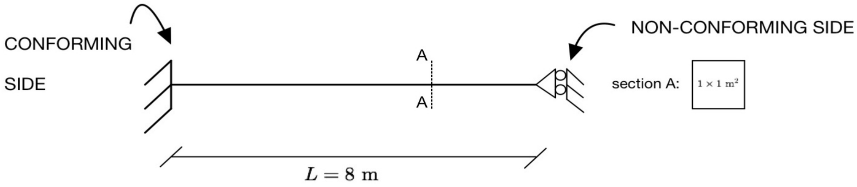

Figure 6.5: Schematic representation of a clamped beam whose vertical right-side displacement is allowed (CASE 2).

We proceed by comparing the numerical solutions obtained with the Material Point Method employing Lagrange Multipliers with the FEM one obtained with a mesh size h = 0.005m. The geometrical and constitutive properties are the same we use for the previous cantilever beam and 16 material points per element have been used.
CASE I:
In this case the numerical solution has been evaluated at the middle point of the beam. The absolute and relative error are plotted in FIG[6.6].

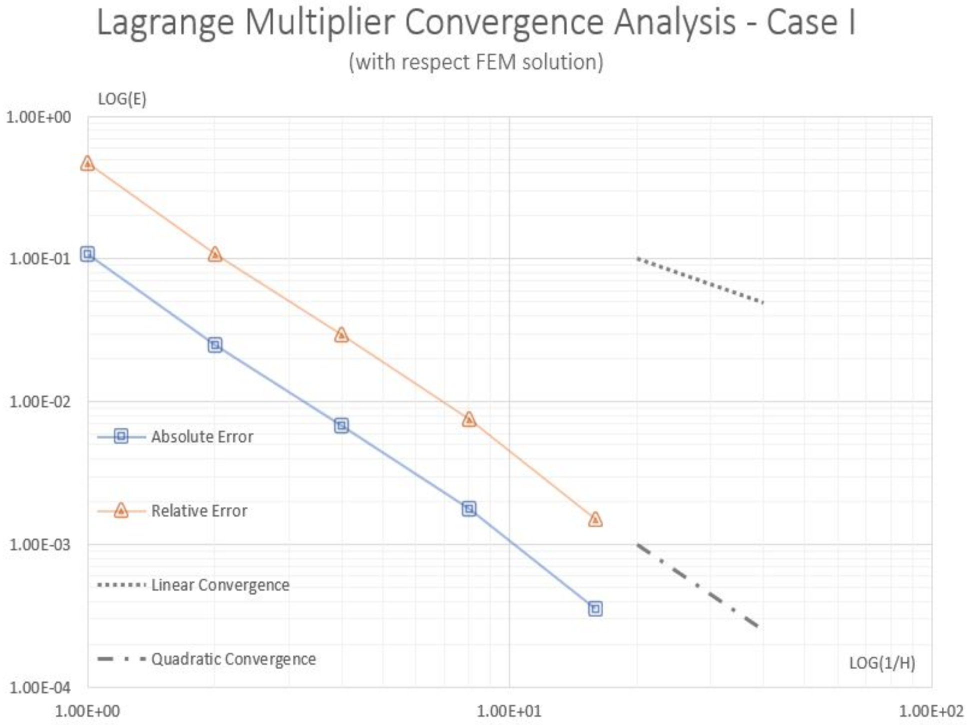

Figure 6.6: Relative and Absolute errors obtained for CASE 1 using Lagrange Multipliers to impose the right side boundary condition.

Similarly to what we saw before, the convergence is quadratic starting from coarse mesh size.

CASE 2:
In this case the right side is constrained is such way that only the vertical displacement are allowed. The numerical solution has been evaluated precisely at the right-end of the beam. Even in this case we observed a quadratic convergence. The absolute and relative error are plotted in FIG[6.7].

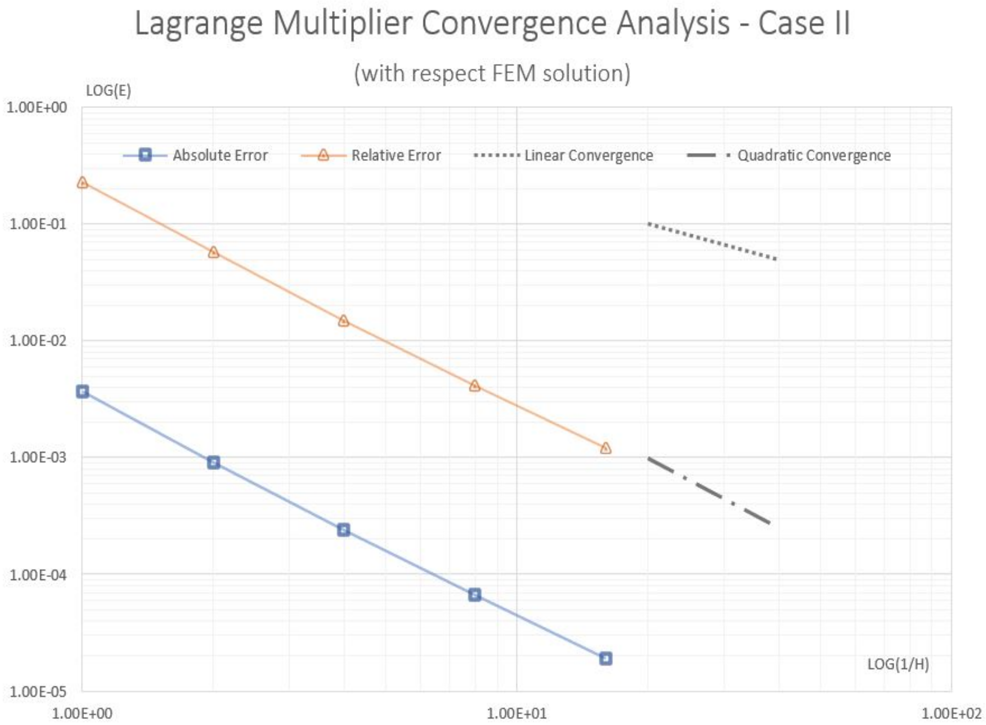

Figure 6.7: Relative and Absolute errors obtained for CASE 2 using Lagrange Multipliers to impose the right side boundary condition.

The order of convergence achieved is clearly similar to the previous example. In FIG[6.8] we report the deformed configuration of CASE 1 beam.

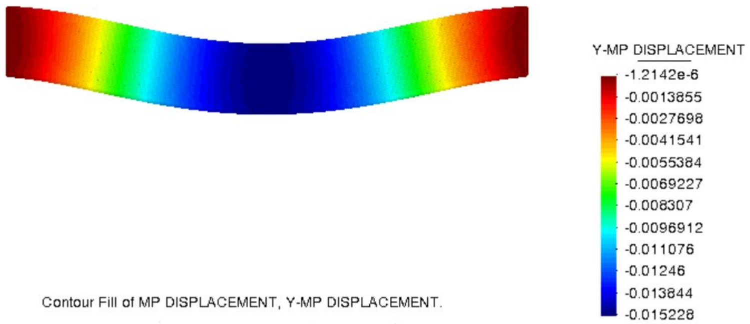

Figure 6.8: CASE 1. Deformation of the double-clamped beam under self-weight obtained by using a mesh size equal to $h = 0.125$ and with 16 material points per element.

6.3 Granular Column

The Material Point Method is particular efficient under very large displacement and deformation regime. In this example we are going to consider a situation where a portion of granular material is maintained still by three wall but one of them is removable (see FIG[6.9]). The left and the bottom walls are treated as nonconforming walls and therefore we are going to use

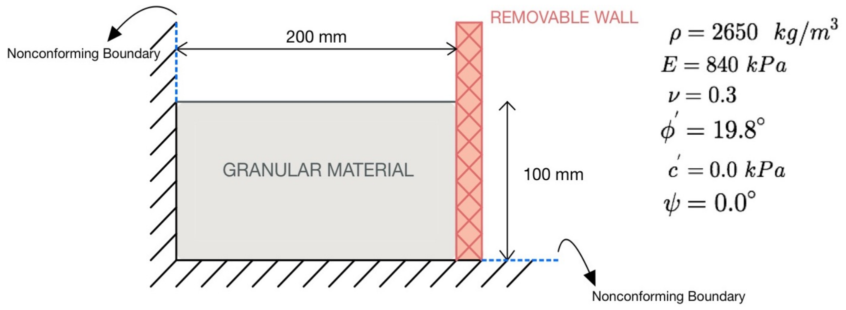

Figure 6.9: Geometry of the granular material domain, nonconforming boundaries and material parameters

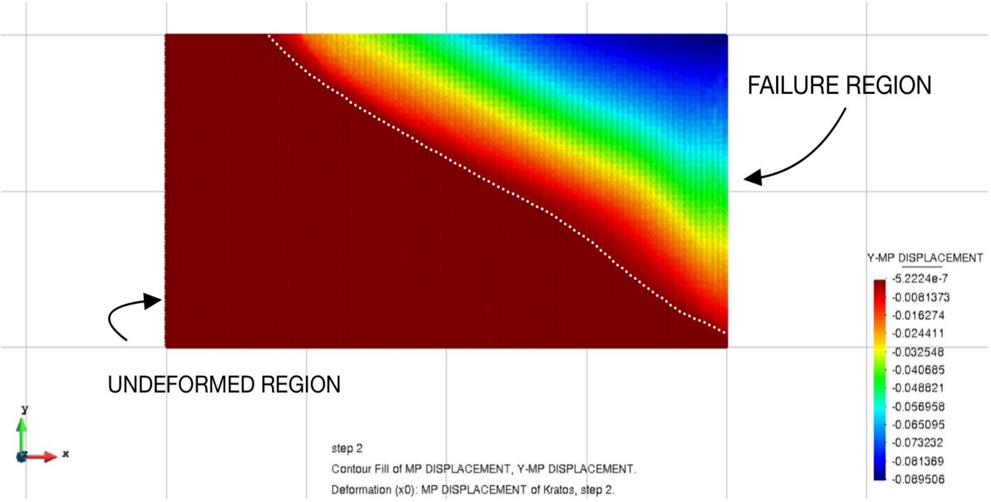

Figure 6.10: Failure region and undeformed region.

the Lagrange Multipliers in order to append the homogeneous boundary conditions.
Let us suppose that at time $t_0 = 0$ s we suddenly remove the wall: what we obtain will be a granular flow on the right side and the material will show a *failure zone* since a portion of granular material is collapsed under its weight due to the absence of the removable wall. Thus, in this situation, we are going to face large deformation and MPM could be a better suited choice than FEM.
The numerical simulation was carried out consistently to the experiments conducted, first by Bui et al. [11], and then by Chandra et al. [14]. In particular, the numerical experiments presented is this thesis is compared to the results obtained by [11] with the SPH (Smoothed Particle Hydrodynamics) method.
The granular material is modeled as a non-associated elastoplastic model, assuming Mohr-Coulomb yield criterion. The elastic material parameter and the plastic variables are summarized in FIG[6.9].

Once we remove the wall, the granular column will collapse under its own weight and what we expect is a failure region. The simulation, reported in FIG[6.10] and FIG[6.11], shows that on the right side the granular material landscapes while on the left side it remains undeformed.

Finally, we show in FIG[6.12] the results comparison between the de-

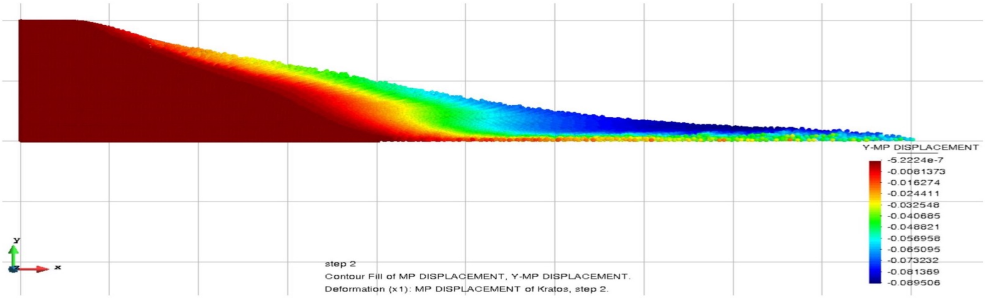

Figure 6.11: Deformed configuration after collapsing.

formed configuration obtained with Lagrange Multiplier, the deformed configuration obtained by Bui et al. [11] by using SPH and the deformed configuration obtained by means of conforming boundary conditions. The results are in accordance with the case with conforming boundaries and reproduce correctly the experimental setting.

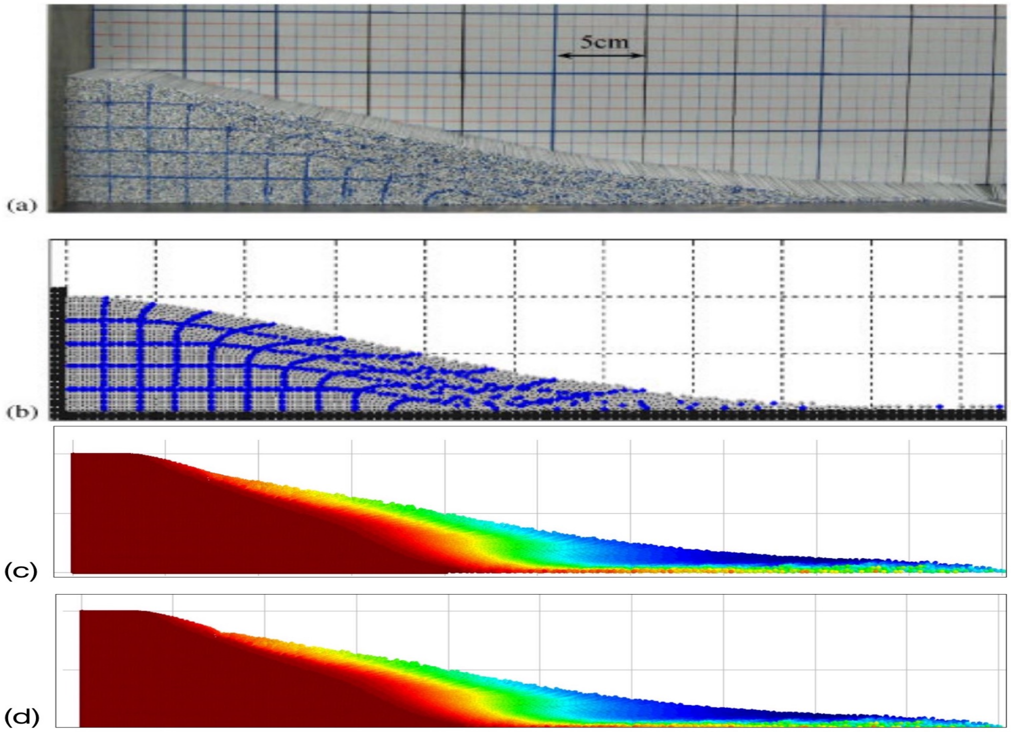

Figure 6.12: Comparison between the experimental data (a), the deformed configuration obtained by Bui et al. [11] by using SPH (b), the deformed configuration obtained with Lagrange Multipliers (c) and deformed configuration obtained by means of conforming boundary conditions (d).

Chapter 7

Conclusions

In this thesis we examined the use of Lagrange Multipliers in computational mechanics. The main goal of this thesis was to formulate the problem of non-conforming boundary conditions treated by means of Lagrange Multipliers applied to the Finite Element Method and Material Point Method. Chapters 2,3 and 4 served as a physical-mathematical introduction to the definition of the mechanical problem and then, in chapter 5, we addressed the problem of Lagrange Multipliers in detail. We analyzed how the constrained mechanical problem turns out to be a saddle point problem (see Equation[5.19]). This problem is well-posed if the Ladyshenskaya-Brezzi-Babuska condition is satisfied. For engineering applications involving boundary or interface condition problems, the LBB-condition is satisfied simply by considering linear basis functions for the primary variable and constant basis functions for the degrees of freedom of the Multipliers. This choice of basis functions was implemented within the Kratos Multiphysics framework with which the numerical applications of chapter 6 were performed. The following box describes the main steps treated in the chapters mentioned before.

Nonconforming Boundary Condition

$\Downarrow$

Lagrange Multipliers Method

$\Downarrow$

Ladyshenskaya-Brezzi-Babuska Condition

$\Downarrow$

$\mathcal{P}_1 - \mathcal{P}_0$ Basis Functions

In chapter 6 we first compared the method of Lagrange Multipliers with Penalty. What comes out, as we expected, is that the latter approach is less accurate than the first one. In fact the convergence obtained with the Multipliers is quadratic even for coarse mesh while this does not happen in Penalty where the convergence is only linear. This suggests that, although MPM is not as accurate as FEM in the context of small deformations (where by small we mean not necessarily infinitesimal) and quadratic convergence is obtained even for coarse meshes using Lagrange Multipliers, in the context of large deformations and displacements (where MPM is a winner over FEM) the convergence that will be obtained will be more than quadratic. The cantilever beam example shows exactly what just described. It follows then that the Lagrange Multipliers method shows faster convergence and smaller error than the Penalty method. The main disadvantage is in the computational cost: the Lagrange Multipliers method adds extra degrees of freedom that increase the size of the linear system to solve.
Thus, the numerical experiments allowed us to compare numerically the Penalty method and the Lagrange Multipliers method. The following boxes show the main differences between these two methods.

LAGRANGE MULTIPLIER

- **Advantages**: It solves with high accuracy the mechanical constrained problem → higher convergence rate than Penalty.
- **Disadvantages**:
  1. It is expensive from a computational point of view.
  2. We need to satisfy the BB-condition meaning we cannot choose arbitrary basis functions.

PENALTY

- **Advantages**
  1. We are free to choose arbitrary basis function.
  2. It is cheaper from a computational point of view.
- **Disadvantages**:
  1. We are not solving the exact mechanical problem due to the penalty factor → lower convergence rate than Lagrange Multipliers.
  2. Sometimes it is not possible to tune the parameter in such way to enforce correctly the constraint.

In chapter 5 we saw the physical meaning of the Lagrange Multipliers. From a dimensional analysis turned out that the Multipliers can be meant as forces. This is very useful in coupled problem. In these kinds of problems two or more bodies interact through their boundaries. Then, the boundary conditions of one of these bodies will be the resulting force (or resulting stress) that the others act on it. Therefore, thanks to the Lagrange Multipliers it is straightforward to obtain this force. Future researches should focus to formulate a mathematical and numerical model implementing the Lagrange Multipliers in coupled problems within the particles methods framework.

List of References

[1] Ivo Babuška. “The finite element method with Lagrangian multipliers”. In: *Numerische Mathematik* 20.3 (1973), pp. 179–192.

[2] Helio JC Barbosa and Thomas JR Hughes. “Boundary Lagrange multipliers in finite element methods: error analysis in natural norms”. In: *Numerische Mathematik* 62.1 (1992), pp. 1–15.

[3] Helio JC Barbosa and Thomas JR Hughes. “The finite element method with Lagrange multipliers on the boundary: circumventing the Babuška-Brezzi condition”. In: *Computer Methods in Applied Mechanics and Engineering* 85.1 (1991), pp. 109–128.

[4] Ted Belytschko et al. *Nonlinear finite elements for continua and structures*. John wiley & sons, 2014.

[5] Javier Bonet, Antonio J Gil, and Richard D Wood. *Nonlinear solid mechanics for finite element analysis: dynamics*. Cambridge University Press, 2021.

[6] Javier Bonet, Antonio J Gil, and Richard D Wood. *Nonlinear solid mechanics for finite element analysis: statics*. Cambridge University Press, 2016.

[7] Jeremiah U Brackbill, Douglas B Kothe, and Hans M Ruppel. “FLIP: a low-dissipation, particle-in-cell method for fluid flow”. In: *Computer Physics Communications* 48.1 (1988), pp. 25–38.

[8] Jeremiah U Brackbill and Hans M Ruppel. “FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions”. In: *Journal of Computational physics* 65.2 (1986), pp. 314–343.

[9] Franco Brezzi. "On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers". In: *Publications mathématiques et informatique de Rennes* S4 (1974), pp. 1–26.

[10] Franco Brezzi and Michel Fortin. *Mixed and hybrid finite element methods*. Vol. 15. Springer Science & Business Media, 2012.

[11] Ha H Bui et al. "Lagrangian meshfree particles method (SPH) for large deformation and failure flows of geomaterial using elastic–plastic soil constitutive model". In: *International journal for numerical and analytical methods in geomechanics* 32.12 (2008), pp. 1537–1570.

[12] Franco Cardin and Marco Favretti. *Modelli fisico matematici*. Cleup, 2013.

[13] Bodhinanda Chandra. "Soil-structure interaction simulation using a coupled implicit material point-finite element method". In: (2019).

[14] Bodhinanda Chandra et al. "Nonconforming Dirichlet boundary conditions in implicit material point method by means of penalty augmentation". In: *Acta Geotechnica* 16.8 (2021), pp. 2315–2335.

[15] J Cotela et al. "Migration of a generic multi-physics framework to HPC environments". In: (2013).

[16] Pooyan Dadvand, Riccardo Rossi, and Eugenio Oñate. "An object-oriented environment for developing finite element codes for multi-disciplinary applications". In: *Archives of computational methods in engineering* 17.3 (2010), pp. 253–297.

[17] Wolfgang Dahmen and Angela Kunoth. "Appending boundary conditions by Lagrange multipliers: Analysis of the LBB condition". In: *Numerische Mathematik* 88.1 (2001), pp. 9–42.

[18] Shuonan Dong. "Methods for constrained optimization". In: *Massachusetts Institute of Technology, Massachusetts* (2006).

[19] James Fern et al. *The material point method for geotechnical engineering: a practical guide*. CRC Press, 2019.

[20] V Mataix Ferrándiz et al. *KratosMultiphysics (Version 8.0)*. 2020.

[21] FH Harlow. *Pic method for fluid dynamics calculations*. Tech. rep. Los Alamos Scientific Lab., N. Mex., 1961.

[22] Francis H Harlow. “Fluid dynamics in group T-3 Los Alamos national laboratory:(LA-UR-03-3852)”. In: *Journal of Computational Physics* 195.2 (2004), pp. 414–433.

[23] Gerhard A Holzapfel. “Nonlinear solid mechanics: a continuum approach for engineering science”. In: *Meccanica* 37.4 (2002), pp. 489–490.

[24] Ilaria Iaconeta et al. “An implicit material point method applied to granular flows”. In: *Procedia Engineering* 175 (2017), pp. 226–232.

[25] Ilaria Iaconeta et al. “Comparison of a material point method and a galerkin meshfree method for the simulation of cohesive-frictional materials”. In: *Materials* 10.10 (2017), p. 1150.

[26] A Larese et al. “Implicit MPM and coupled MPM-FEM in geomechanics”. In: *Computational mechanics* 175 (2019), pp. 226–232.

[27] Vinh Phu Nguyen et al. “On a family of convected particle domain interpolations in the material point method”. In: *Finite Elements in Analysis and Design* 126 (2017), pp. 50–64.

[28] Alfio Quarteroni. *Modellistica numerica per problemi differenziali.* Vol. 97. Springer, 2016.

[29] Deborah Sulsky, Zhen Chen, and Howard L Schreyer. “A particle method for history-dependent materials”. In: *Computer methods in applied mechanics and engineering* 118.1-2 (1994), pp. 179–196.

[30] Deborah Sulsky and A Kaul. “Implicit dynamics in the material-point method”. In: *Computer Methods in Applied Mechanics and Engineering* 193.12-14 (2004), pp. 1137–1170.

[31] Deborah Sulsky, Shi-Jian Zhou, and Howard L Schreyer. “Application of a particle-in-cell method to solid mechanics”. In: *Computer physics communications* 87.1-2 (1995), pp. 236–252.

[32] William Trench. “The method of lagrange multipliers”. In: (2013).

[33] Alban de Vaucorbeil, Vinh Phu Nguyen, and Christopher R Hutchinson. “A Total-Lagrangian Material Point Method for solid mechanics problems involving large deformations”. In: *Computer Methods in Applied Mechanics and Engineering* 360 (2020), p. 112783.

[34] Alban de Vaucorbeil et al. “Material point method after 25 years: Theory, implementation, and applications”. In: *Advances in applied mechanics* 53 (2020), pp. 185–398.

[35] Bin Wang et al. “Development of an implicit material point method for geotechnical applications”. In: *Computers and Geotechnics* 71 (2016), pp. 159–167.

[36] Zdzisław Więckowski. “The material point method in large strain engineering problems”. In: *Computer methods in applied mechanics and engineering* 193.39-41 (2004), pp. 4417–4438.

[37] Peter Wriggers. *Nonlinear finite element methods*. Springer Science & Business Media, 2008.

[38] Xiong Zhang, Zhen Chen, and Yan Liu. *The material point method: a continuum-based particle method for extreme loading cases*. Academic Press, 2016.

Appendix A

Appendix

A.1 Linearized Kinematic

Directional derivatives arise from the linearization of virtual works. These directional derivatives involves quantities such as the deformation gradient $\boldsymbol{F}$, the Green-Lagrange tensor $\boldsymbol{E}$, the Cauchy-Green tensor $\boldsymbol{C}$ and so on. In this section we are going to compute the directional derivatives of the main quantities involved.

A.1.1 Directional derivative of the Deformation Gradient

We recall that the deformation gradient $\boldsymbol{F}$, already stated in Equation[2.9], is a two-point tensor defined as

$$\boldsymbol{F}(\mathbf{X}, t) = \frac{\partial \boldsymbol{\phi}(\boldsymbol{X}, t)}{\partial \boldsymbol{X}}; \qquad F_{aA} = \frac{\partial \phi_a}{\partial X_A}. \tag{A.1}$$

In order to linearize the deformation gradient, let us consider an increment $\boldsymbol{u}$ of the motion $\boldsymbol{\phi}(\boldsymbol{X}, t)$. The directional derivative is

$$\begin{aligned}
&D\boldsymbol{F}(\boldsymbol{\phi}(\boldsymbol{X},t))[\boldsymbol{X}] = \frac{d}{d\epsilon}\bigg|_{\epsilon=0} \boldsymbol{F}(\boldsymbol{\phi}(\boldsymbol{X},t)) + \epsilon\boldsymbol{u}) \\
&= \frac{d}{d\epsilon}\bigg|_{\epsilon=0} \frac{\partial(\boldsymbol{\phi}(\boldsymbol{X},t) + \epsilon\boldsymbol{u})}{\partial\boldsymbol{X}} \\
&= \frac{d}{d\epsilon}\bigg|_{\epsilon=0} \left(\frac{\partial\boldsymbol{\phi}(\boldsymbol{X},t)}{\partial\boldsymbol{X}} + \epsilon\frac{\partial\boldsymbol{u}}{\partial\boldsymbol{X}}\right) \\
&= \frac{d}{d\epsilon}\bigg|_{\epsilon=0} \frac{\partial\boldsymbol{\phi}(\boldsymbol{X},t)}{\partial\boldsymbol{X}} + \frac{d}{d\epsilon}\bigg|_{\epsilon=0} \epsilon\frac{\partial\boldsymbol{u}}{\partial\boldsymbol{X}} \\
&= \frac{d}{d\epsilon}\bigg|_{\epsilon=0} \epsilon\frac{\partial\boldsymbol{u}}{\partial\boldsymbol{X}} = \frac{\partial\boldsymbol{u}}{\partial\boldsymbol{X}} \\
&= \frac{\partial\boldsymbol{u}(\boldsymbol{x})}{\partial\boldsymbol{X}} = \frac{\partial\boldsymbol{u}}{\partial\boldsymbol{x}}\frac{\partial\boldsymbol{x}}{\partial\boldsymbol{X}} = \nabla\boldsymbol{u}\ \boldsymbol{F} \quad \text{(spatial)} \\
&= \frac{\partial\boldsymbol{u}(\boldsymbol{X})}{\partial\boldsymbol{X}} = \frac{\partial\boldsymbol{u}}{\partial\boldsymbol{X}}\frac{\partial\boldsymbol{X}}{\partial\boldsymbol{X}} = \nabla_0\boldsymbol{u} \quad \text{(material)}
\end{aligned} \tag{A.2}$$

A.1.2 Directional Derivative of Green-Lagrange tensor

The Green-Lagrange tensor is a material strain tensor defined as:

$$\boldsymbol{E} = \frac{1}{2}\left(\boldsymbol{F}^T\boldsymbol{F} - \boldsymbol{I}\right); \qquad E_{AB} = \frac{1}{2}\left(F_{aA}F_{aB} - \delta_{AB}\right) \tag{A.3}$$

Considering properties [3.10],[3.11] and making use of the directional derivative of the deformation gradient in Equation[A.2], the directional derivative of the Green-Lagrange strain tensor is

$$\begin{aligned}
&D\boldsymbol{E}[\boldsymbol{u}] = \tfrac{1}{2}D\left(\boldsymbol{F}^T\boldsymbol{F} - \boldsymbol{I}\right)[\boldsymbol{u}] = \tfrac{1}{2}D\left(\boldsymbol{F}^T\boldsymbol{F}\right)[\boldsymbol{u}] \\
&= \tfrac{1}{2}\left(D\boldsymbol{F}^T[\boldsymbol{u}]\boldsymbol{F} + \boldsymbol{F}^T D\boldsymbol{F}[\boldsymbol{u}]\right) \\
&= \tfrac{1}{2}\left(\boldsymbol{F}^T\nabla\boldsymbol{u}^T\boldsymbol{F} + \boldsymbol{F}^T\nabla\boldsymbol{u}\ \boldsymbol{F}\right) \\
&= \tfrac{1}{2}\boldsymbol{F}^T\left(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T\right)\boldsymbol{F} = \boldsymbol{F}^T\boldsymbol{\epsilon}\boldsymbol{F}
\end{aligned} \tag{A.4}$$

where $\boldsymbol{\epsilon}$ is the small strain tensor.

A.1.3 Directional derivative of left and right Cauchy-Green tensors

The right Cauchy-Green tensor $\boldsymbol{C}$ is a material strain tensor defined as

$$\boldsymbol{C} = \boldsymbol{F}^T\boldsymbol{F}; \qquad C_{AB} = F_{aA}F_{aB} \tag{A.5}$$

and its directional derivative is

$$\begin{aligned} D\boldsymbol{C}[\boldsymbol{u}] &= D\left(\boldsymbol{F}^T\boldsymbol{F}\right)[\boldsymbol{u}] = \\ &= D\boldsymbol{F}^T[\boldsymbol{u}]\boldsymbol{F} + \boldsymbol{F}^T D\boldsymbol{F}[\boldsymbol{u}] \\ &= \boldsymbol{F}^T\nabla\boldsymbol{u}^T\boldsymbol{F} + \boldsymbol{F}^T\nabla\boldsymbol{u}\ \boldsymbol{F} \\ &= \boldsymbol{F}^T\left(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T\right)\boldsymbol{F} = 2\boldsymbol{F}^T\boldsymbol{\epsilon}\boldsymbol{F}. \end{aligned} \tag{A.6}$$

The left Cauchy-Green tensor $\boldsymbol{b}$ is a spatial strain tensor defined as

$$\boldsymbol{b} = \boldsymbol{F}\boldsymbol{F}^T; \qquad b_{ab} = F_{aA}F_{bA} \tag{A.7}$$

and its directional derivative is

$$\begin{aligned} D\boldsymbol{b}[\boldsymbol{u}] &= D\left(\boldsymbol{F}\boldsymbol{F}^T\right)[\boldsymbol{u}] = \\ &= D\boldsymbol{F}[\boldsymbol{u}]\boldsymbol{F}^T + \boldsymbol{F}D\boldsymbol{F}^T[\boldsymbol{u}] \\ &= \nabla\boldsymbol{u}\ \boldsymbol{F}\boldsymbol{F}^T + \boldsymbol{F}\boldsymbol{F}^T\nabla\boldsymbol{u} \\ &= \nabla\boldsymbol{u}\ \boldsymbol{b} + \boldsymbol{b}\nabla\boldsymbol{u}^T \end{aligned} \tag{A.8}$$

A.1.4 Directional derivative of the time rate GL Tensor

The time rate of the Green-Lagrange tensor is defined as

$$\dot{\boldsymbol{E}} = \frac{1}{2}\left(\dot{\boldsymbol{F}}^T\boldsymbol{F} + \boldsymbol{F}^T\dot{\boldsymbol{F}}\right). \tag{A.9}$$

As we will see, the principle of virtual works involves the virtual time rate of the Green-Lagrange tensor that by analogy is defined as

$$\delta\dot{\boldsymbol{E}} = \frac{1}{2}\left(\delta\dot{\boldsymbol{F}}^T\boldsymbol{F} + \boldsymbol{F}^T\delta\dot{\boldsymbol{F}}\right), \tag{A.10}$$

where $\delta\dot{\boldsymbol{F}} = \nabla_0\delta\boldsymbol{v}$. It is worth observing that, since the virtual velocity $\boldsymbol{\delta v}$ is not a function of the body configuration, the directional derivative of $\nabla_0\delta\boldsymbol{v}$ is zero. Then, we can write

$$\begin{aligned} D\delta\dot{\boldsymbol{E}}[\boldsymbol{u}] &= \tfrac{1}{2}D\left(\delta\dot{\boldsymbol{F}}^T\boldsymbol{F} + \boldsymbol{F}^T\delta\dot{\boldsymbol{F}}\right)[\boldsymbol{u}] \\ &= \tfrac{1}{2}\left(D\left(\nabla_0\delta\boldsymbol{v}^T\boldsymbol{F}\right)[\boldsymbol{u}] + D\left(\boldsymbol{F}^T\nabla_0\delta\boldsymbol{v}\right)[\boldsymbol{u}]\right) \\ &= \tfrac{1}{2}\left(\nabla_0\delta\boldsymbol{v}^T D\boldsymbol{F}[\boldsymbol{u}] + D\boldsymbol{F}^T[\boldsymbol{u}]\nabla_0\delta\boldsymbol{v}\right) \\ &= \tfrac{1}{2}\left(\nabla_0\delta\boldsymbol{v}^T\nabla_0\boldsymbol{u} + \nabla_0\boldsymbol{u}^T\nabla_0\delta\boldsymbol{v}\right). \end{aligned} \tag{A.11}$$

A.2 FEM Discretization

Here, we are going to write the FEM discretization of the relevant quantities present in the weak formulation and in its linearization.

A.2.1 FEM Discretization of the relevant kinematic quantities.

- Material position

$$\boldsymbol{X} = \sum_{i=1}^{n} N_i(\boldsymbol{X})\boldsymbol{X}_i, \tag{A.12}$$

- Spatial position

$$\boldsymbol{x} = \sum_{i=1}^{n} N_i(\boldsymbol{X})\boldsymbol{x}_i, \tag{A.13}$$

- Displacement

$$\boldsymbol{u} = \sum_{i=1}^{n} N_i(\boldsymbol{X})\boldsymbol{u}_i, \tag{A.14}$$

- Material Velocity

$$\boldsymbol{v}(\boldsymbol{X}) = \sum_{i=1}^{n} N_i(\boldsymbol{X}) \boldsymbol{v}_i(t), \tag{A.15}$$

- Material Acceleration

$$\boldsymbol{a}(\boldsymbol{X}) = \sum_{i=1}^{n} N_i(\boldsymbol{X}) \boldsymbol{a}_i(t), \tag{A.16}$$

- Virtual velocity

$$\delta\boldsymbol{v}(\boldsymbol{X}) = \sum_{i=1}^{n} N_i(\boldsymbol{X}) \delta\boldsymbol{v}_i, \tag{A.17}$$

- Deformation gradient

$$\boldsymbol{F} = \sum_{i=1}^{n} \boldsymbol{x}_i \otimes \nabla_0 N_i(\boldsymbol{X}), \tag{A.18}$$

- Right Cauchy-Green tensor

$$\boldsymbol{C} = \boldsymbol{F}^T\boldsymbol{F} = \sum_{i=1}^{n}\sum_{j=1}^{n} (\boldsymbol{x}_i \cdot \boldsymbol{x}_j)\, \nabla_0 N_i(\boldsymbol{X}) \otimes \nabla_0 N_j(\boldsymbol{X}), \tag{A.19}$$

- Left Cauchy-Green tensor

$$\boldsymbol{b} = \boldsymbol{F}\boldsymbol{F}^T = \sum_{i=1}^{n}\sum_{j=1}^{n} (\, \nabla_0 N_i(\boldsymbol{X}) \cdot \nabla_0 N_j(\boldsymbol{X})\,)\, \boldsymbol{x}_i \otimes \boldsymbol{x}_j, \tag{A.20}$$

- Small strain tensor

$$\boldsymbol{\epsilon} = \frac{1}{2}\sum_{i=1}^{n} (\, \boldsymbol{u}_i \otimes \nabla N_i(\boldsymbol{X}) + \nabla N_i(\boldsymbol{X}) \otimes \boldsymbol{u}_i\,)\,, \tag{A.21}$$

- Rate of deformation tensor

$$\boldsymbol{d} = \frac{1}{2}\sum_{i=1}^{n} (\, \boldsymbol{v}_i \otimes \nabla N_i(\boldsymbol{X}) + \nabla N_i(\boldsymbol{X}) \otimes \boldsymbol{v}_i\,)\,, \tag{A.22}$$

- Rate of virtual deformation tensor

$$\delta\boldsymbol{d} = \frac{1}{2}\sum_{i=1}^{n} (\, \delta\boldsymbol{v}_i \otimes \nabla N_i(\boldsymbol{X}) + \nabla N_i(\boldsymbol{X}) \otimes \delta\boldsymbol{v}_i\,)\,. \tag{A.23}$$

A.2.2 FEM Discretization of the Inertial, Internal and External Virtual Works

- Inertial Term

$$
\begin{aligned}
&\delta W_m^{(e)}(\boldsymbol{\phi}, \delta \boldsymbol{v}_h) = \int_{v^{(e)}} \rho \boldsymbol{a}_h \cdot \delta \boldsymbol{v}_h \; dv \\
&= \int_{v^{(e)}} \rho \boldsymbol{a}_h \cdot \sum_{i=1}^{n} N_i(\boldsymbol{X}) \delta \boldsymbol{v}_i \; dv \\
&= \sum_{i=1}^{n} \delta \boldsymbol{v}_i \cdot \int_{v^{(e)}} N_i(\boldsymbol{X}) \rho \boldsymbol{a}_h \; dv \\
&= \sum_{i=1}^{n} \delta \boldsymbol{v}_i \cdot \int_{v^{(e)}} N_i(\boldsymbol{X}) \rho \sum_{j=1}^{n} N_j(\boldsymbol{X}) \boldsymbol{a}_j \; dv \\
&= \sum_{i=1}^{n} \delta \boldsymbol{v}_i \cdot \sum_{j=1}^{n} \left(\int_{v^{(e)}} \rho N_i(\boldsymbol{X}) N_j(\boldsymbol{X}) \; dv \right) \boldsymbol{a}_j \\
&= \sum_{i=1}^{n} \delta \boldsymbol{v}_i \cdot \sum_{j=1}^{n} \left(\boldsymbol{M}_{ij}^{(e)} \boldsymbol{a}_j \right) = \delta \boldsymbol{v}^{T(e)} \boldsymbol{M}^{(e)} \boldsymbol{a}^{(e)}
\end{aligned}
\tag{A.24}
$$

- Internal Term

$$\begin{aligned}
\delta W^{(e)}_{int}(\boldsymbol{\phi}, \delta \boldsymbol{v}_h) &= \int_{v^{(e)}} \boldsymbol{\sigma} : \delta \boldsymbol{d}_h \; dv \\
&= \int_{v^{(e)}} \boldsymbol{\sigma} : \frac{1}{2} \sum_{i=1}^{n} (\; \delta \boldsymbol{v}_i \otimes \nabla N_i(\boldsymbol{X}) + \nabla N_i(\boldsymbol{X}) \otimes \delta \boldsymbol{v}_i \;) \; dv \\
&= \int_{v^{(e)}} \boldsymbol{\sigma} : \sum_{i=1}^{n} (\; \delta \boldsymbol{v}_i \otimes \nabla N_i(\boldsymbol{X})) \; dv \\
&= \sum_{i=1}^{n} \delta \boldsymbol{v}_i \cdot \int_{v^{(e)}} \boldsymbol{\sigma} \nabla N_i(\boldsymbol{X}) \; dv \\
&= \sum_{i=1}^{n} \delta \boldsymbol{v}_i \cdot \boldsymbol{T}^{(e)}_i = \delta \boldsymbol{v}^{T,(e)} \cdot \boldsymbol{T}^{(e)}
\end{aligned} \tag{A.25}$$

- External Term

$$\begin{aligned}
\delta W^{(e)}_{ext}(\boldsymbol{\phi}, \delta \boldsymbol{v}_h) &= \int_{v^{(e)}} \boldsymbol{f} \cdot \delta \boldsymbol{v}_h \; dv + \int_{a^{(e)}} \boldsymbol{t} \cdot \delta \boldsymbol{v}_h \; da \\
&= \int_{v^{(e)}} \boldsymbol{f} \cdot \sum_{i=1}^{n} N_i(\boldsymbol{X}) \delta \boldsymbol{v}_i \; dv + \int_{a^{(e)}} \boldsymbol{t} \cdot \sum_{i=1}^{n} N_i(\boldsymbol{X}) \delta \boldsymbol{v}_i \; da \\
&= \sum_{i=1}^{n} \delta \boldsymbol{v}_i \cdot \left(\int_{v^{(e)}} \boldsymbol{f} N_i(\boldsymbol{X}) \; dv + \int_{a^{(e)}} \boldsymbol{t} N_i(\boldsymbol{X}) \; da \right) \\
&= \sum_{i=1}^{n} \delta \boldsymbol{v}_i \cdot \boldsymbol{F}^{(e)}_i = \delta \boldsymbol{v}^{T,(e)} \cdot \boldsymbol{F}^{(e)}
\end{aligned} \tag{A.26}$$

A.2.3 FEM discretization of the Linearized Virtual Works

- Material Term

$$\begin{aligned}
&D\delta W_{int}^{mat,(e)}(\boldsymbol{\phi}, \delta \boldsymbol{v}_h)[\boldsymbol{u}_h] = \\
&= \int_{v^{(e)}} \frac{1}{2} \sum_{i=1}^{n} \left(\delta \boldsymbol{v}_i \otimes \nabla N_i + \nabla N_i \otimes \delta \boldsymbol{v}_i\right) : \hat{\mathbb{C}} : \frac{1}{2} \sum_{j=1}^{n} \left(\boldsymbol{u}_j \otimes \nabla N_j + \nabla N_j \otimes \boldsymbol{u}_j\right) \, dv \\
&= \int_{v^{(e)}} \delta \boldsymbol{d}^T \boldsymbol{D} \boldsymbol{\epsilon} \, dv = \sum_{i=1}^{n} \delta \boldsymbol{v}_i \cdot \sum_{j=1}^{n} \left(\int_{v^{(e)}} \boldsymbol{B}_i^T \boldsymbol{D} \boldsymbol{B}_j \, dv \right) \boldsymbol{u}_j = \delta \boldsymbol{v}^{T,(e)} \cdot \boldsymbol{K}^{mat,(e)} \boldsymbol{u}^{(e)}
\end{aligned} \tag{A.27}$$

- Geometric Term

$$\begin{aligned}
&D\delta W_{int}^{geo,(e)}(\boldsymbol{\phi}, \delta \boldsymbol{v}_h)[\boldsymbol{u}_h] = \int_{v^{(e)}} \boldsymbol{\sigma} : \left(\nabla \boldsymbol{u}_h^T \nabla \delta \boldsymbol{v}_h\right) \, dv \\
&= \int_{v^{(e)}} \boldsymbol{\sigma} : \sum_{i=1}^{n} \sum_{j=1}^{n} \left[\left(\delta \boldsymbol{v}_i \cdot \boldsymbol{u}_j\right) \nabla N_j \otimes \nabla N_i \right] \, dv \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\delta \boldsymbol{v}_i \cdot \boldsymbol{u}_j\right) \int_{v^{(e)}} \nabla N_i \cdot \boldsymbol{\sigma} \nabla N_j \, dv \\
&= \sum_{i=1}^{n} \delta \boldsymbol{v}_i \cdot \left[\sum_{j=1}^{n} \left(\int_{v^{(e)}} \nabla N_i \cdot \boldsymbol{\sigma} \nabla N_j \boldsymbol{I} \, dv \right) \boldsymbol{u}_j \right] = \delta \boldsymbol{v}^{T,(e)} \cdot \boldsymbol{K}^{geo,(e)} \boldsymbol{u}^{(e)}
\end{aligned} \tag{A.28}$$