Literature study

High order material point method

February 10, 2016

R.P.W.M. Tielen

Technische Universiteit Delft

TU Delft Delft University of Technology

Deltares Enabling Delta Life

Challenge the future

HIGH ORDER MATERIAL POINT METHOD

FEBRUARY 10, 2016

by

R.P.W.M. Tielen

Supervisors: Ir. E.D. Wobbes TU Delft
Dr.ir. M. Möller TU Delft
Dr.-Ing. L.I. Beuth Deltares

TU Delft Delft University of Technology

Deltares Enabling Delta Life

CONTENTS

1

INTRODUCTION

The finite element method (FEM) is a well known method to numerically solve partial differential equations. In continuum mechanics however, when problems involving large deformations are considered, the use of the (updated Lagrangian) FEM can lead to numerical inaccuracies [1].

As an alternative numerical method, the material point method (MPM) has been developed. MPM uses a fixed background mesh and a set of material points moving through the mesh to model the deforming solid. Linear basis functions are commonly used to project the quantities of interest from the nodes to the material points and vice versa.

The use of linear basis functions has its disadvantages. The discontinuity of the shape function derivatives across element boundaries may lead to numerical problems within the MPM. Furthermore, physical quantities such as stresses are not accurately reproduced. The use of higher-order basis functions for which the derivatives are continious over element boundaries is expected to reduce these numerical problems leading to a more accurate MPM solution. The aim of this master project is to develop a high order material point method that makes use of higher-order B-spline basis functions. Besides a reduction of the numerical problems described above, a decrease of the computational costs might be achieved with this type of basis functions, when the same accuracy can be achieved as with linear basis functions while using a coarser mesh.

This thesis is carried out together with Deltares, a Dutch research and consulting company. At Deltares underwater slope failures will be investigated with the MPM. Since the computational domain might cover hunderds of meters, long computing times are expected. A reduction of the computational costs is therefore relevant to perform these simulations in reasonable time.

This report is the result of the literature study performed in the period October-December 2015. The objective of this report is to describe the material point method and discuss the problems that occur while using Lagrange nodal basis functions of order one or two. The results obtained during the first months of the master project form the basis to develop a high order material point method.

Chapter 2 gives a brief review of the development of MPM. The different steps of the material point method are described in detail. The chapter includes a discussion of the problems that occur when linear basis functions are used within MPM. In the beginning of Chapter 3 a number of benchmark problems are provided. The performance of the 1D MPM using linear basis functions is investigated for these benchmark problems.

Chapter 4 introduces a general framework for deriving a higher-order projection from the material points to the nodes. In [2] this projection was used to improve the spatial convergence when using linear basis functions within MPM. In Chapter 5 quadratic Lagrange nodal basis functions will be introduced. The results obtained with a 1D version of MPM which uses these basis functions is compared with the 1D MPM version which uses linear basis functions.

2

MATERIAL POINT METHOD

This chapter gives an introduction into the material point method. In the first section of this chapter the basic concept of the material point method is explained. Then a review of the development of the material point method is given after which each step within MPM is discussed in more detail. Finally, numerical problems encountered when using basis functions for which the derivatives are discontinuous across element boundaries are described.

2.1. GENERAL DESCRIPTION

2.1.1. FUNCTIONING OF MPM

The material point method is used to numerically solve partial differential equations that describe large deformation processes of continuum bodies. As with the finite element method, the spatial domain is discretized by a finite number of elements. A time integration scheme is used to obtain the solution of the partial differential equation over time.

With the material point method, a matter is represented as a set of material points which store all physical properties of the material like mass, velocity and stresses. The material points move through the mesh over time, representing the deforming body. In this thesis, also the name *particles* is used to refer to material points.

At the beginning of a time step, all this information is projected onto the nodes of the background grid. On this mesh the equations of motion are solved, after which the updated velocities are mapped back to the particles. The stresses are computed at the material points from the nodal displacements. At the end of each time step, the background mesh is reset to its original position. Figure 2.1 shows a single time step of the material point method.

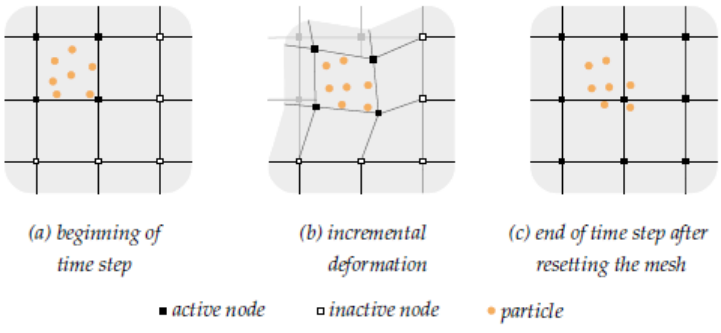

Figure 2.1: The basic concept of the material point method illustrated. Retrieved from [3].

If an element is filled with at least one particle, the nodes associated with this element are called *active*. After each time step, the set of active nodes can change when a particle crosses the boundary of an element.

2.1.2. DEVELOPMENT OF MPM

The material point method can be considered as a *mesh-based particle method*. Mesh-based particle methods use a background mesh and a set of material points moving through this mesh [3].

One of the earliest mesh-based particle methods is the particle-in-cell (PIC) method developed at Los Alamos National Laboratory by F. H. Harlow for fluid dynamics analyses [4]. In this method, material points only carry information on mass and position of the continuum, not on velocities or stresses. Dissipation of energy is characteristic for this method [5].

A next step was the introduction of the Fluid-implicit particle method (FLIP) by Brackbill and Ruppel [5] in 1986. In this method, not only mass and position but also other properties such as momentum and energy are assigned to each particle. It has been shown in [6] that this method conserves kinetic energy, if a consistent mass matrix is used.

In 1993, the FLIP method was extended by Sulsky et al. for application to solid mechanics [1]. Here, particles are interpreted to be material points that are followed through the complete calculation process. This new method was called the material point method by Sulsky and Schreyer in [7].

The material point method can be seen as an extension of the finite element method which combines a Lagrangian and Eulerian description of the material deformation.

In the Lagrangian description, the observer follows the material as it moves through the mesh. Therefore, the position of nodes is updated every timestep according to the calculated nodal velocities. Hence, in Lagrangian FEM the computational mesh follows the deforming continuum, trying to represent it as accurate as possible with the chosen boundary approximation. However, when (very) large deformations are considered the mesh gets tangled leading to unrealistic results.

In the Eulerian description the observer is fixed in space as the material moves through the mesh. Therefore, the nodes remain fixed during the computations while the material deforms. This has the advantage that the mesh will not get tangled when large deformations are considered. A disadvantage of the Eulerian approach is the appearance of convective terms [3]. Figure 2.2 shows both the Lagrangian and Eulerian approach in an initial and deformed state.

The material point method combines the Lagrangian and Eulerian approach in the following way: the equations of motion are solved on the fixed background grid, while particles carry all the physical properties of the material and move through the mesh over time. Since the mesh is reset at the end of each time step, the mesh can not become tangled unless very large deformations over a single time step are considered.

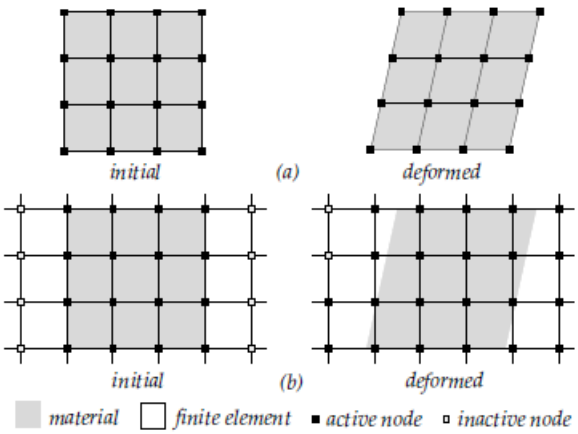

Figure 2.2: The Lagrangian (a) and Eulerian (b) approach in an initial and deformed state. Retrieved from [3].

2.2. MATHEMATICAL MODEL

In this section the material point method will be described in more detail. The section starts with a number of definitions, after which the weak form of the conservation of momentum will be derived. At the end of this section, every step of the material point method is discussed, including the discretization in space and time.

Since the focus of this literature study is on the material point method in 1D, we describe the governing equations and space discretization only in 1D. The description can easily be transferred to 2D and 3D, see [3].

2.2.1. DEFINITIONS

Consider the deformation of a continuum with initial domain Ω_0 at time $t = 0$ s. The configuration Ω_t represents the state of the continuum after deformation at time t and will be referred to as the current configuration. The position of a material point x^t at time t can be written as a function of its initial position x^0 and t:

$$x^t = f(x^0, t).$$

The displacement $u(x,t)$ of a material point can be derived from its current and initial position by the following relation:

$$u(x,t) = x^t - x^0.$$

The velocity and acceleration of a material point can be obtained from the displacement by taking the derivative with respect to t:

$$\begin{aligned} v(x,t) &= \frac{\partial u}{\partial t}(x,t), \\ a(x,t) &= \frac{\partial v}{\partial t}(x,t). \end{aligned}$$

The absence of the convective term in both formulas is due to the Lagrangian description.

2.2.2. GOVERNING EQUATIONS

The conservation of linear momentum in 1D is given by:

$$\rho \frac{\partial v}{\partial t} = \frac{\partial \sigma}{\partial x} + \rho g, \tag{2.1}$$

where ρ denotes the density, g the gravitational acceleration and σ the stress.

For linear-elastic materials, the constitutive relation between stress and strain ϵ is given by

$$\frac{\partial \sigma}{\partial t} = E \frac{\partial \epsilon}{\partial t},$$

where the strain is related to the displacement by the following equation:

$$\frac{\partial \epsilon}{\partial t} = \frac{\partial v}{\partial x}.$$

By using these equations, we can rewrite Equation (2.1) as

$$\rho \frac{\partial^2 u}{\partial t^2} = E \frac{\partial^2 u}{\partial x^2} + \rho g. \tag{2.2}$$

This partial differential equation describes the displacement of a continuum in $\mathbb{R}$ due to self weight. The partial differential equation is second order in time and space.

To illustrate the solution procedure, consider Equation (2.2) with homogeneous boundary and initial conditions

$$\begin{aligned} u(0,t) &= 0, \\ \frac{\partial u}{\partial x}(L,t) &= \frac{p_0}{E}, \\ u(x,0) &= 0, \\ \frac{\partial u}{\partial t}(x,0) &= 0. \end{aligned}$$

Since this hyperbolic partial differential equation is second order in time, $\frac{\partial u}{\partial t}$ has to be given at time $t = 0$. The Neumann boundary condition at $x = L$ describes the application of a load at the boundary of the continuum.

The weak form of this partial differential equation is derived by multiplying both sides with a test function w and integrating over the domain $\Omega = [0, L]$:

$$\int_0^L w\rho \frac{\partial^2 u}{\partial t^2} \mathrm{d}x = \int_0^L wE \frac{\partial^2 u}{\partial x^2} \mathrm{d}x + \int_0^L w\rho g \mathrm{d}x.$$

Function w has to be continuous and zero on the part of the boundary where essential boundary conditions are presented. The next step is to apply integration by parts:

$$\begin{aligned} \int_0^L w\rho \frac{\partial^2 u}{\partial t^2} \mathrm{d}x &= \int_0^L wE \frac{\partial^2 u}{\partial x^2} \mathrm{d}x + \int_0^L w\rho g \mathrm{d}x \\ &= wE \frac{\partial u}{\partial x}\Big|_0^L - \int_0^L \frac{\partial w}{\partial x} E \frac{\partial u}{\partial x} \mathrm{d}x + \int_0^L w\rho g \mathrm{d}x \\ &= w(L,t)p_0 - \int_0^L \frac{\partial w}{\partial x} E \frac{\partial u}{\partial x} \mathrm{d}x + \int_0^L w\rho g \mathrm{d}x, \end{aligned}$$

where the boundary integral vanishes at $x = 0$ because $w = 0$ on the part of the boundary where essential boundary conditions are presented.

2.2.3. SPACE DISCRETIZATION

Both w and u are approximated in terms of basis functions ϕ and time dependent coefficients:

$$\begin{aligned} u(x,t) \approx u_h(x,t) = \sum_{j=1}^{n} \phi_j(x) u_j(t) = \boldsymbol{\phi}(x)^T \mathbf{u}(t), \\ w(x,t) \approx w_h(x,t) = \sum_{j=1}^{n} \phi_j(x) w_j(t) = \boldsymbol{\phi}(x)^T \mathbf{w}(t), \end{aligned}$$

where

$$\boldsymbol{\phi}(x) = [\phi_1(x) \;\; \phi_2(x) \;\; \dots \;\; \phi_n(x)]^T, \qquad \mathbf{u}(t) = [u_1(t) \;\; u_2(t) \;\; \dots \;\; u_n(t)]^T, \qquad \mathbf{w}(t) = [w_1(t) \;\; w_2(t) \;\; \dots \;\; w_n(t)]^T.$$

By using these expressions and the fact that the coefficients only depend on t the weak form becomes

$$\begin{aligned} \int_0^L (\boldsymbol{\phi}^T \mathbf{w})^T \rho (\boldsymbol{\phi}^T \ddot{\mathbf{u}}) \mathrm{d}x &= \boldsymbol{\phi}(L)^T \mathbf{w} p_0 - \int_0^L (\boldsymbol{\phi}'^T \mathbf{w})^T E (\boldsymbol{\phi}'^T \mathbf{u}) \mathrm{d}x + \int_0^L \boldsymbol{\phi}^T \mathbf{w} \rho g \mathrm{d}x, \\ \mathbf{w}^T \int_0^L \boldsymbol{\phi} \rho \boldsymbol{\phi}^T \mathrm{d}x \, \ddot{\mathbf{u}} &= \mathbf{w}^T \boldsymbol{\phi}(L) p_0 - \mathbf{w}^T \int_0^L \boldsymbol{\phi}' E \boldsymbol{\phi}'^T \mathrm{d}x \, \mathbf{u} + \mathbf{w}^T \int_0^L \boldsymbol{\phi} \rho g \mathrm{d}x. \end{aligned} \tag{2.3}$$

Here $\ddot{\mathbf{u}}$ denotes the second derivative of $\mathbf{u}$ with respect to t and $\boldsymbol{\phi}'$ the derivative of $\boldsymbol{\phi}$ with respect to x.Since the representation of $w_h(x,t)$ in terms of the basis functions $\boldsymbol{\phi}$ and time dependent coefficients $\mathbf{w}$ is unique, it suffices to test Equation (2.3) only for the basis functions that span the test space.

This results in the following equation:

$$\int_0^L \boldsymbol{\phi}\rho\boldsymbol{\phi}^T \mathrm{d}x\, \ddot{\mathbf{u}} \quad = \quad \boldsymbol{\phi}(L)p_0 - \int_0^L \boldsymbol{\phi}' E \boldsymbol{\phi}'^T \mathrm{d}x\, \mathbf{u} + \int_0^L \boldsymbol{\phi}\rho g \mathrm{d}x\,.$$

This is commonly written as

$$\mathbf{M}\ddot{\mathbf{u}} \quad = \quad \mathbf{F}^{\text{trac}} - \mathbf{F}^{\text{int}} + \mathbf{F}^{\text{grav}}.$$

The global mass matrix $\mathbf{M}$ and global force vectors $\mathbf{F}^{\text{trac}}, \mathbf{F}^{\text{int}}$ and $\mathbf{F}^{\text{grav}}$ are, respectively, defined by

$$\begin{aligned}
\mathbf{M} &= \int_0^L \boldsymbol{\phi}\rho\boldsymbol{\phi}^T \mathrm{d}x, \\
\mathbf{F}^{\text{trac}} &= \boldsymbol{\phi}(L)p_0, \\
\mathbf{F}^{\text{int}} &= \int_0^L \boldsymbol{\phi}' E \boldsymbol{\phi}'^T \mathrm{d}x\, \mathbf{u}, \\
\mathbf{F}^{\text{grav}} &= \int_0^L \boldsymbol{\phi}\rho g \mathrm{d}x.
\end{aligned}$$

In one dimension the computational domain Ω becomes a line segment $[0, L]$. In this thesis, the line segment is discretized by defining n_e elements with element size $h = \frac{L}{n_e}$. The boundary of element e_i consists of nodes i and $i+1$, where the position of node i is denoted by x_i. Figure 2.3 illustrates the partition of a line segment.

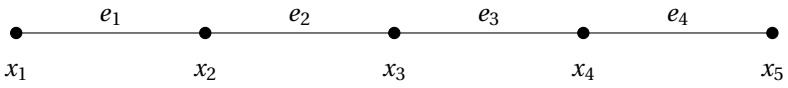

Figure 2.3: Line segment of length L consisting of four elements and five nodes.

Since the elements are pairwise disjoint, we can evaluate the integrals above by summing the integrals over one single element. The restriction of a basis function on a single element is called a shape function. We will denote a shape function by N_i and the gradient of a shape function by B_i.

For example, an arbitrary entry (i, j) of the global mass matrix is given by

$$\mathbf{M}_{(i,j)} = \sum_{k=1}^{n_e} \int_{e_k} N_i(x)\rho(x)N_j(x)\mathrm{d}x.$$

To evaluate the integrals over an element e_k, a numerical integration rule is used. In FEM different numerical integration rules can be used, each involving different numbers, weights and locations of integration points. In MPM however, the material points are used as integration points to approximate the integrals over an element. More information about the numerical integration performed within the material point method is provided in Section A.3.

With the material point method, multiple types of shape functions can be used. At first, we will only consider linear shape functions. It should be noted that each element consists of two nodes situated at the boundary of the elements when linear basis functions are used. In case of higher-order basis functions, however, an element might possess more than two nodes.

Let e_i be an element defined by the nodes x_i and x_{i+1}. Each element is transformed to a reference element via the mapping

$$\mathrm{T}_{e_i} : e_{\text{ref}} \to e_i$$

defined by

$$\mathrm{T}_{e_i}(\xi) = x_i + (x_{i+1} - x_i)\xi,$$

where $\xi \in [0,1]$. Since the Jacobian of this transformation is non-singular for each ξ, the inverse transformation exists and is given by

$$\mathrm{T}_{e_i}^{-1}(x) = \frac{x - x_i}{x_{i+1} - x_i}.$$

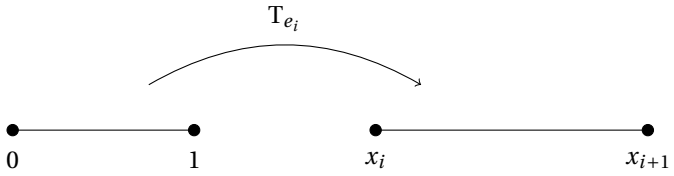

Figure 2.4: Transformation from element e_{ref} to reference element e_i.

Figure 2.4 illustrates the mapping T_{e_i} from the reference element e_{ref} to element e_i.

If a particle is situated in element e_i at position $x_p \in [x_i, x_{i+1}]$ we refer to x_p as the *global position* of this particle. We define $\xi_p = \frac{x_p - x_i}{x_{i+1} - x_i}$ to be the *local position* of a particle.

On e_{ref} we define the following linear shape functions

$$\begin{aligned} \hat{N}_1(\xi) &= 1 - \xi, \\ \hat{N}_2(\xi) &= \xi. \end{aligned}$$

where $\xi \in [0,1]$. The relation between the shape functions $\hat{N}_i$ and N_i is then given by

$$\hat{N}_i(T_{e_i}^{-1}(x_p)) = N_i(x_p).$$

To reduce the storage space, only the values of the shape functions at the local particle positions are stored. Note that these values are sufficient to perform the numerical integration within MPM. Every time step the local position of the particles is updated.

From the shape functions, a piecewise linear basis function can be constructed for each node. An example of such basis function is shown in Figure 2.5.

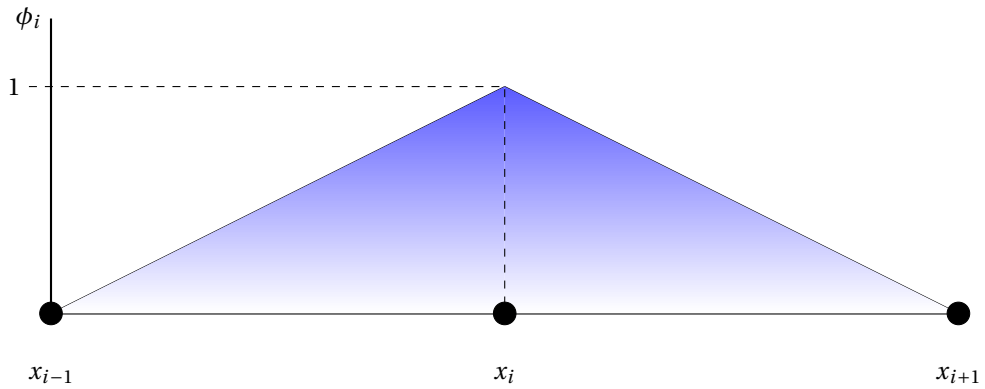

Figure 2.5: Basis function ϕ_i corresponding to node i.

Linear basis functions have the following properties:

- The basis functions are piecewise linear.
- For all $i \in \{1, \ldots, n\}$ the basis function ϕ_i has compact support on $[x_{i-1}, x_{i+1}]$.
- For all $x \in [0, L]$ we have $\sum_{i=1}^{n} \phi_i(x) = 1$. (Partition of unity property.)

2.2.4. MPM SOLUTION

Let Ω be our domain of computation and define a set of n_p particles representing a continuum body in its initial configuration. Each particle is assigned an initial position x_p, velocity v_p, mass m_p, volume V_p and stress σ_p, with $p \in \{1,2,\ldots,n_p\}$. Depending on the context, other quantities such as the gravitational force f_p^{grav} or a traction force f_p^{trac} may be assigned to particles.

A grid is defined consisting of n_e elements and n_n nodes x_i, with $i \in \{1,2,\ldots,n_n\}$. Initially the number of particles in an element is chosen to be the same for all active elements. We denote the number of particles in an element by $n_{p,e}$. Figure 2.6 shows a discretized domain $\Omega \subset \mathbb{R}$ in which equally distributed material points are defined.

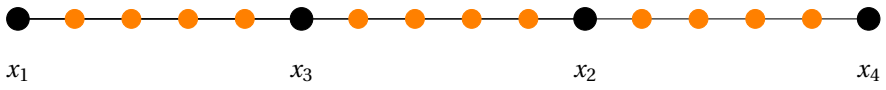

Figure 2.6: A grid with domain $\Omega \subset \mathbb{R}$ discretized with 3 elements and 4 material point per element.

Since the equations of motion are solved on the background mesh, the physical information of the particles has to be transferred to the grid points. The information of a particle is transferred to the nodes of the element in which the particle is situated. In Section 2.2.4 this projection is explained in detail.

To make the transfer possible, basis functions ϕ_i are used which can be of arbitrary order. We will denote the gradient of a basis function by $\nabla\phi_i$, with $i \in \{1,\ldots,n_n\}$. The properties of different choices of basis functions and their effect on the performance of the material point method will be investigated during this master project.

Suppose that at time t all the physical information of the particles is known. In order to determine the state of the particles at time $t+\Delta t$, the mass matrix $\mathbf{M}$ and force vectors $\mathbf{F}$ have to be determined and the assembled equations of motion have to be solved for the nodal accelerations. Once the nodal accelerations are known, the state of the particles at time $t+\Delta t$ can be calculated. Here, the *Modified Lagrangian algorithm* proposed in [3] will be used. Figure 2.7 shows an overview of the calculations performed during one time step.

MASS MATRIX AND FORCE VECTORS

The global mass matrix and global force vectors are determined by assembling the mass matrices and force vectors of each element. At the beginning of a new time step, the element mass matrices $\mathbf{M_e}^t$ and element force vectors $\mathbf{F_e}^{\text{grav},t}$, $\mathbf{F_e}^{\text{trac},t}$ and $\mathbf{F_e}^{\text{int},t}$ are determined. There are three element force vectors, based on the gravitational, external and internal forces applied on the material.

The element matrices and vectors are determined by projecting the information from the particles to the nodes. The particles in the elements are used as integration points to approximate the integrals:

$$
\begin{aligned}
\mathbf{M_e}^t_{(i,j)} &= \sum_{p=1}^{n_{p,e}} m_p N_i(x_p^t) N_j(x_p^t), \\
\mathbf{F_e}^{\text{grav},t}_{(i)} &= \sum_{p=1}^{n_{p,e}} f_p^{\text{grav},t} N_i(x_p^t), \\
\mathbf{F_e}^{\text{trac},t}_{(i)} &= \sum_{p=1}^{n_{p,e}} f_p^{\text{trac},t} N_i(x_p^t), \\
\mathbf{F_e}^{\text{int},t}_{(i)} &= \sum_{p=1}^{n_{p,e}} \sigma_p^t V_p B_i(x_p^t).
\end{aligned}
$$

The global mass matrix $\mathbf{M}^t \in \mathbb{R}^{n_n \times n_n}$ and force vectors $\mathbf{F}^{\text{grav},t}, \mathbf{F}^{\text{trac},t}$ and $\mathbf{F}^{\text{int},t} \in \mathbb{R}^{n_n \times 1}$ can be determined by assembling the element matrices and element vectors. In Section A.1, the assembly procedure is explained in more detail. Instead of the consistent mass matrix, a lumped mass matrix may be used to reduce the computational time. More explanation about lumped matrices can be found in Section A.2. The numerical integration rule used in the material point method is examined in Section A.3.

Once the mass matrix and total force vector are known, the equation of motion can be solved for the nodal accelerations at time t:

$$\mathbf{a}^t = (\mathbf{M}^t)^{-1}\mathbf{F}^t.$$

The nodal accelerations at time t are used to update the particle velocities v_p:

$$v_p^{t+\Delta t} = v_p^t + \sum_{i=1}^{n_n} \Delta t \phi_i(x_p^t) a_i^t.$$

In order to determine the nodal velocities the updated nodal momentum vector $\mathbf{P}^{t+\Delta t}$ has to be known. This vector is formed by assembling the element momentum vectors $\mathbf{P_e}^{t+\Delta t}$ which are defined by

$$\mathbf{P_e}_{(i)}^{t+\Delta t} = \sum_{p=1}^{n_{pe}} m_p \phi_i(x_p^t) v_p^{t+\Delta t}.$$

The nodal velocities are then given by the following relation:

$$\mathbf{v}^{t+\Delta t} = (\mathbf{M}^t)^{-1}\mathbf{P}^{t+\Delta t}.$$

With the nodal velocities at time $t+\Delta t$, the nodal incremental displacements at time $t+\Delta t$ can be calculated:

$$\Delta u_i^{t+\Delta t} = v_i^{t+\Delta t} \Delta t$$

The information on the nodes is used to determine the displacement and position of the particles:

$$\begin{aligned} u_p^{t+\Delta t} &= u_p^t + \sum_{i=1}^{n_n} \phi_i(x_p^t) \Delta u_i^{t+\Delta t}, \\ x_p^{t+\Delta t} &= x_p^t + \sum_{i=1}^{n_n} \phi_i(x_p^t) \Delta u_i^{t+\Delta t}, \end{aligned}$$

Finally, the stress of the material points is updated:

$$\sigma_p^{t+\Delta t} = \sigma_p^t + (E - \sigma_p^t) \Delta \epsilon_p^{t+\Delta t},$$

where

$$\Delta \epsilon_p^{t+\Delta t} = \sum_{i=1}^{n_n} \nabla \phi_i(x_p^t) \Delta u_i^{t+\Delta t}.$$

In the last step, both volume and density are updated for each particle:

$$\begin{aligned} V_p^{t+\Delta t} &= (1 + \Delta \epsilon_p^{t+\Delta t}) V_p^t, \\ \rho_p^{t+\Delta t} &= \frac{\rho_p^t}{(1 + \Delta \epsilon_p^{t+\Delta t})}, \end{aligned}$$

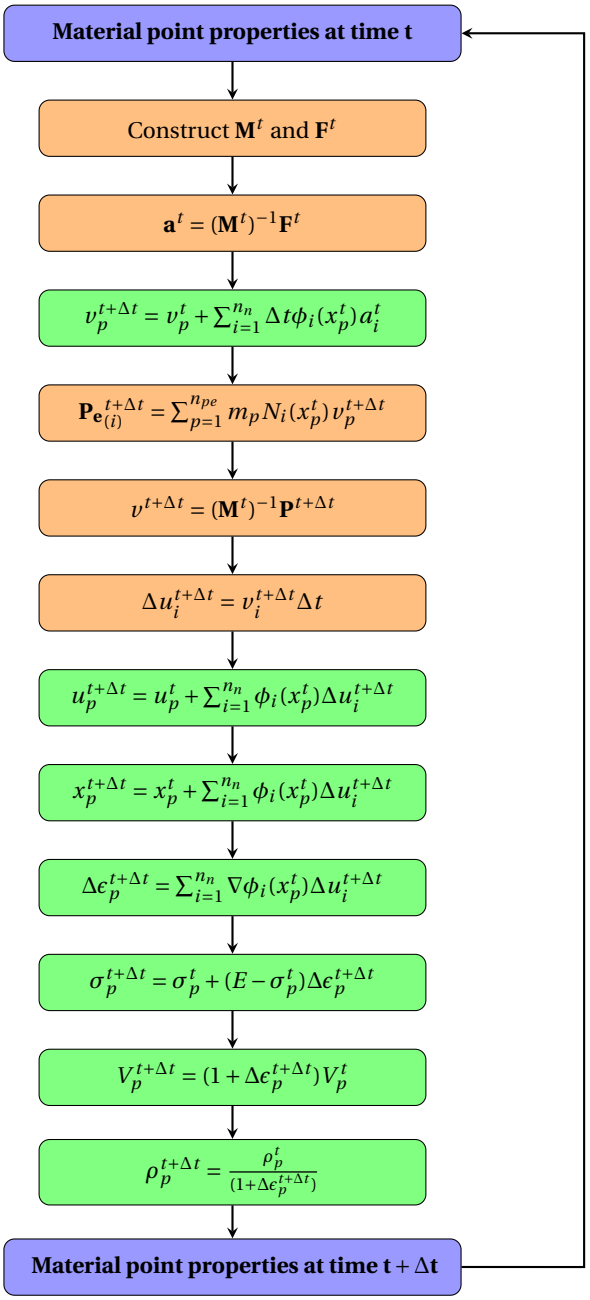

Figure 2.7: Overview of MPM calculations within a time step. The green blocks denote the update of the material point properties.

2.2.5. TIME INTEGRATION SCHEME

The nodal velocities at $t+\Delta t$ are determined by the mass matrix and the updated momentum vector:

$$v^{t+\Delta t} = (\mathbf{M}^t)^{-1}\mathbf{P}^{t+\Delta t}.$$

The updated momentum vector is formed by assembling the element momentum vectors $\mathbf{P_e}$. The element momentum vectors depend on the particle velocities:

$$\mathbf{P_e}_{(i)}^{t+\Delta t} = \sum_{p=1}^{n_{pe}} m_p \phi_i(x_p^t) v_p^{t+\Delta t}.$$

The particle velocities can be calcultad from the nodal accelerations at time t:

$$v_p^{t+\Delta t} = \mathbf{v}_p^t + \sum_{i=1}^{n_n} \Delta t \phi_i(x_p^t) a_i^t.$$

Hence, the nodal velocities at time $t+\Delta t$ are calculated from the nodal accelerations at time t. The nodal displacements at time $t+\Delta t$ are calculated from the nodal velocities at time $t+\Delta t$:

$$u_i^{t+\Delta t} = u_i^t + v_i^{t+\Delta t}\Delta t.$$

This time integration scheme, which combines an explicit and implicit step, is known as the Euler-Cromer method [8].

In [8] a stability criterion was determined for the finite element method based on the Von Neumann method and the Matrix method, leading to the following upper bound for the time step size:

$$\Delta t \leq \frac{h}{\sqrt{\frac{E}{\rho}}}. \tag{2.4}$$

A stability criterion for the material point method is harder to determine, since the critical time step depends on the number of particles inside each element [8]. Since the number of particles in each element can change over time, the critical time step might change over time. Therefore, it is recomended to determine the critical time step size at every time level.

In practice, we will use the stability criterion obtained for the finite element method to choose the time step size.

2.3. NUMERICAL PROBLEMS

The use of linear basis functions within MPM has its disadvantages. As stated in the introduction, the discontinuity of the basis function derivatives across element boundaries lead to numerical problems. In this section these problems are described in more detail.

2.3.1. GRID CROSSING ERROR

With the MPM, material points might eventually cross boundaries of elements during a computation. These grid crossings influence the internal forces calculated at the nodes from the material point data.

To explain this effect, recall that the internal force at node i is calculated in the following way:

$$\mathbf{F}_{(i)}^{\text{int},t} = \sum_{p=1}^{n_p} \sigma_p^t V_p^t \nabla\phi(x_p^t).$$

Figure 2.8 denotes a mesh consisting of two elements, each having two particles initially. Assume $h = 1$ and that each particle has the same stress σ and volume V, both constant over time. The internal force at node 2 is then given by:

$$\mathbf{F}_2^{\text{int,t}} = \sum_{p=1}^{n_p} \sigma V \nabla\phi(x_p^t) = 2\sigma V - 2\sigma V = 0.$$

Suppose one particle moves from the first element to the second element. The internal force at node 2 then suddenly becomes:

$$\mathbf{F}_2^{\text{int,t}} = \sum_{p=1}^{n_p} \sigma V \nabla\phi(x_p^t) = \sigma V - 3\sigma V = -2\sigma V.$$

Hence, the crossing of the boundary of an element leads to a non-physical difference in the internal forces.

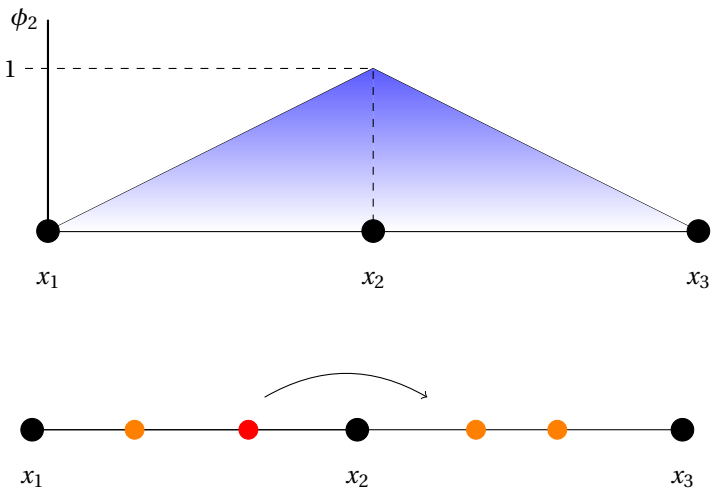

Figure 2.8: Illustration of a grid crossing error when a particle crosses the boundary of an element.

Grid crossing errors can have a serious effect on the performance of the material point method. One way to reduce the effect of grid crossing errors might be the use of functions for which the gradient is continuous across element boundaries, for example quadratic B-spline basis functions.

In [9] the spatial convergence of the material point method was determined when using different types of basis functions. Besides linear basis functions, both quadratic and cubic B-spline functions were used. When using linear basis functions, a lack of convergence was observed. Both quadratic and cubic B-spline functions showed spatial convergence up to a relatively high number of elements. Therefore, the use of higher-order B-spline functions is recommended in [9].

2.3.2. QUADRATURE ERROR

Besides the error associated with grid crossings, there is another type of error that effects the performance of the material point method. When the internal force at node i is determined, the particle volume is used to approximate the domain over which integration is performed. However, the volumes of the particles might not sum up to the size of an element, leading to an quadrature error.

Figure 2.9 illustrates the problem that might occur when particles move through the background grid. In this case, linear basis functions are used leading to discontinuities accross the boundary of an element. The configuration shown at the right leads to a quadrature error since the volume of the particle does not coincide with the element boundary. It was shown in [9] that the use of quadratic and cubic B-spline functions reduces the quadrature error.

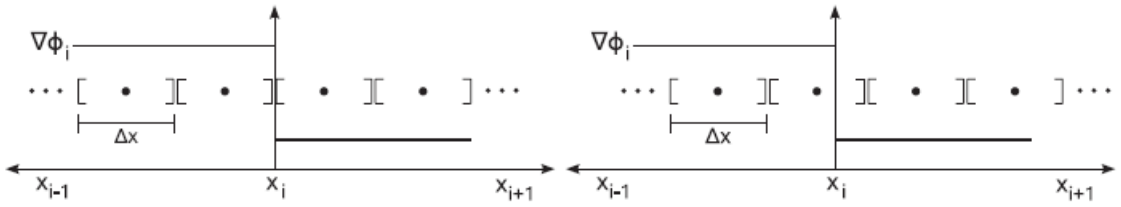

Figure 2.9: A particle configuration where the discontinuity of the basis functions is respected (left) and a configuration where it is not respected (right). Reprinted from [9].

2.4. UPDATED LAGRANGIAN FEM

The updated Lagrangian FEM (ULFEM) renders accurate results as long as mesh distortions remain low. Since for large deformation problems an analytical solution is not always available, ULFEM can be used to obtain a reference solution for validation of the MPM. In order to be able to validate the solution obtained with the material point method, we implemented the updated Lagrangian finite element method.

Compared with the standard FEM, the position of the nodes is now updated every time step. The position of the nodes at time $t+\Delta t$ can be determined by the following equation:

$$u_i^{t+\Delta t} = u_i^t + v_i^{t+\Delta t}\Delta t.$$

Since the position of the nodes is no longer fixed, the element size is no longer constant over time and might vary across the discretized space. Hence, the element vectors and matrices have to be determined at the beginning of every timestep.

When considering large deformations, an extra term has to be added to determine the stress at integration point c

$$\sigma_c^{t+\Delta t} = \sigma_c^t + (E - \sigma_c^t)\Delta\epsilon_c^{t+\Delta t},$$

where

$$\Delta\epsilon_c^{t+\Delta t} = \sum_{j=1}^{n_n} B_j(x_c^t)\Delta u_j^{t+\Delta t}.$$

The solution obtained with the updated Lagrangian FEM will be used as a reference solution for the material point method solution. The validation of the updated Lagrangian FEM can be found in A.4.

3

1D LINEAR MATERIAL POINT METHOD

In this chapter the material point method in 1D is discussed which makes use of linear basis functions. Different benchmark problems are described which possess an analytical solution. A 1D MPM code has been implemented with Matlab. Validation of the code is presented with the different benchmark problems. Besides checking the ability of the MPM code to reproduce the analytical solution of the corresponding partial differential equations, spatial convergence will be investigated.

3.1. BENCHMARK PROBLEMS

In this chapter three benchmark problems are considered to test the implemented material point method described in Chapter 2. The obtained results will be used as a reference to study the performance of a high order material point method.

The first benchmark problem models the vibrations of string which is fixed at both ends. The displacement of the string is caused by an initial velocity. The gravitational acceleration is set equal to zero for this test case and no external force is applied on the string. Since the displacements are relatively small, this benchmark will be used to obtain results without the effect of grid crossing. For the displacement of mode 1 an analytical solution exists.

The second benchmark problem models the movement of a column of soil in 1D. The displacement of the soil is caused by the self-weight of the soil. Since the deformations are relatively large, this benchmark problem will be used to examine the effect of grid crossing. An analytical solution exists for this benchmark problem.

The last benchmark problem vibrations of a bar with one fixed end. The displacement of the bar is caused by a dynamic load applied at the free end of the bar. This benchmark problem was presented in [9] and will be used to investigate whether we can obtain the same results as the authors.

3.2. ACCURACY

Numerically solving Equation (2.2) gives an approximation u_h of the exact solution $\bar{u}$. Since both space and time discretization lead to numerical errors, we can write

$$|u_h(x,t) - \bar{u}(x,t)| = \mathcal{O}(\Delta x^n) + \mathcal{O}(\Delta t^m),$$

where Δx denotes the element size h. For the Euler-Cromer method, it is known that $m = 1$ [3]. To determine the spatial convergence, the error $|u_h(x,t) - \bar{u}(x,t)|$ at time t is approximated by the Reduced Mean Square (RMS) error:

$$e^{\text{RMS}} = \sqrt{\frac{\sum_{p=1}^{n_p} [u_h(x_p,t) - \tilde{u}(x_p,t)]^2}{n_p}}.$$

A numerical method is said to converge in space with order n if for $\Delta t \to 0$ the reducation of Δx by a factor 2 leads to a reduction of the error $|u_h(x,t) - \bar{u}(x,t)|$ by a factor 2^n.

3.3. VIBRATING STRING

3.3.1. DESCRIPTION

The first benchmark concerns the vibrations of string which is fixed at both ends. At $t = 0$ s, an initial velocity $v_0(x)$ is prescribed causing the vibration of the string. Figure 3.1 gives a schematic overview of the vibrating string problem.

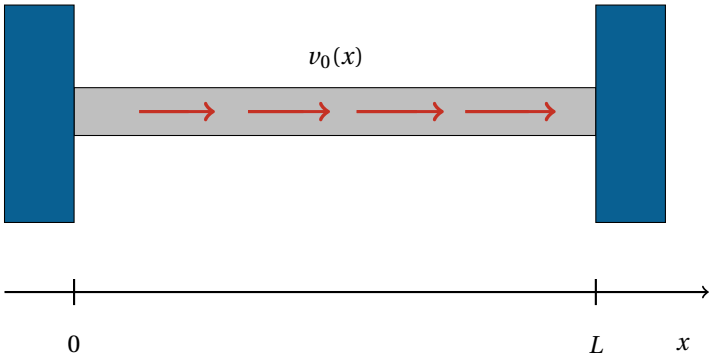

Figure 3.1: The vibrating string problem.

To model the vibrations of a string with both ends fixed, consider the one-dimensional wave equation [10]

$$\frac{\partial^2 u}{\partial t^2} = \frac{E}{\rho}\frac{\partial^2 u}{\partial x^2},$$

with boundary and initial conditions:

$$\begin{aligned} u(0,t) &= 0, \\ u(L,t) &= 0, \\ u(x,0) &= 0, \\ \frac{\partial u}{\partial t}(x,0) &= v_0 \cdot \sin\left(\frac{\pi x}{L}\right). \end{aligned}$$

Since the partial differential equation is second order in time also $\frac{\partial u}{\partial t}$ has to be specified at time $t = 0$ s. At time $t = 0$ s there is no displacement, but there is an initial velocity. The analytical solution of mode 1 is given by

$$u(x,t) = \frac{v_0}{\omega_1} \cdot \sin(\omega_1 t) \cdot \sin(\beta_1 x),$$

where

$$\begin{aligned} \omega_1 &= \frac{\pi\sqrt{\frac{E}{\rho}}}{L}, \\ \beta_1 &= \frac{\pi}{L}. \end{aligned}$$

An overview of the parameter values is provided in Table 3.1.

Quantity	Symbol	Value	Unit
Density	ρ	1	kg/m^3
Young's modulus	E	100	Pa
Length	L	25	m
Velocity	v_0	0.1	m/s

Table 3.1: Parameters used for the model of a vibrating string with fixed ends.

3.3.2. RESULTS

For the vibrating string problem the spatial convergence is determined. The parameters used in these simulations can be found in Table 3.1. The number of elements was varied between 4 and 64 and a time step size of $\Delta t = 1 \cdot 10^{-4}$ s was used. For all configurations, the critical time step size was not exceeded.

Figure 3.2 shows the spatial convergence of the material point method that makes use of linear basis functions. Different numbers of particles per cell (PPC) and a different number of elements were used in this convergence study. The RMS error was determined at time $t = 0.1$ s. During these simulations, no grid crossings occured before $t = 0.1$ s.

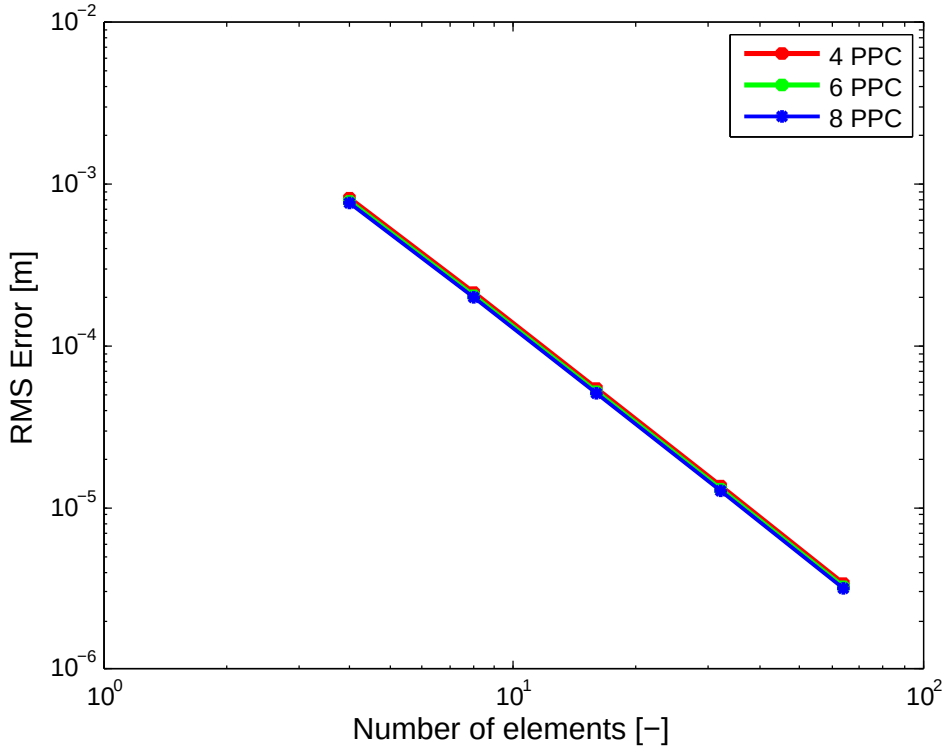

Figure 3.2: Spatial convergence with different numbers of PPC at time $t = 0.1$ s.

The figure illustrates the positive influence of increasing the number of elements on the RMS error. Furthermore, when the domain is discretized by the same number of elements, increasing the number of particles decreases the RMS error . To determine the rate of convergence, the RMS errors are presented in Table 3.2. We conclude that the material point method with linear basis functions shows quadratic convergence for the vibrating string problem.

elements	e^{RMS}	$\frac{e^{\text{RMS}}(h)}{e^{\text{RMS}}(h/2)}$	$\log_2\left(\frac{e^{\text{RMS}}(h)}{e^{\text{RMS}}(h/2)}\right)$
4	$8.2921 \cdot 10^{-4}$		
8	$2.1496 \cdot 10^{-4}$	3.858	1.948
16	$5.4230 \cdot 10^{-5}$	3.964	1.987
32	$1.3588 \cdot 10^{-5}$	3.991	1.997
64	$3.3983 \cdot 10^{-6}$	3.998	1.999

Table 3.2: Accuracy of numerical solution at time $t = 0.1$ s with 4 PPC.

Figure 3.3 and 3.4 illustrate respectively the position and velocity of the particle situated directly left of the middle of the vibrating string with 4 PPC and 20 elements. A time step size was used of $\Delta t = 1 \cdot 10^{-3}$ s.

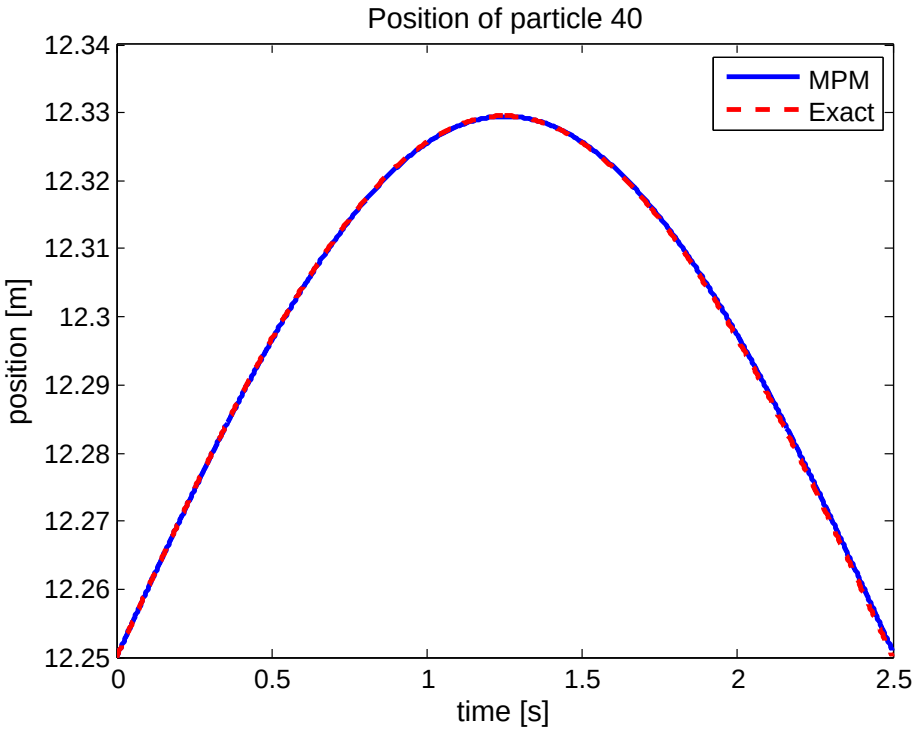

Figure 3.3: Displacement of the particle situated directly left of the middle of the vibrating string as a function of time with 4 PPC and a time step size of $\Delta t = 1 \cdot 10^{-3}$ s. The number of elements is equal to 20. Grid crossing is not observed.

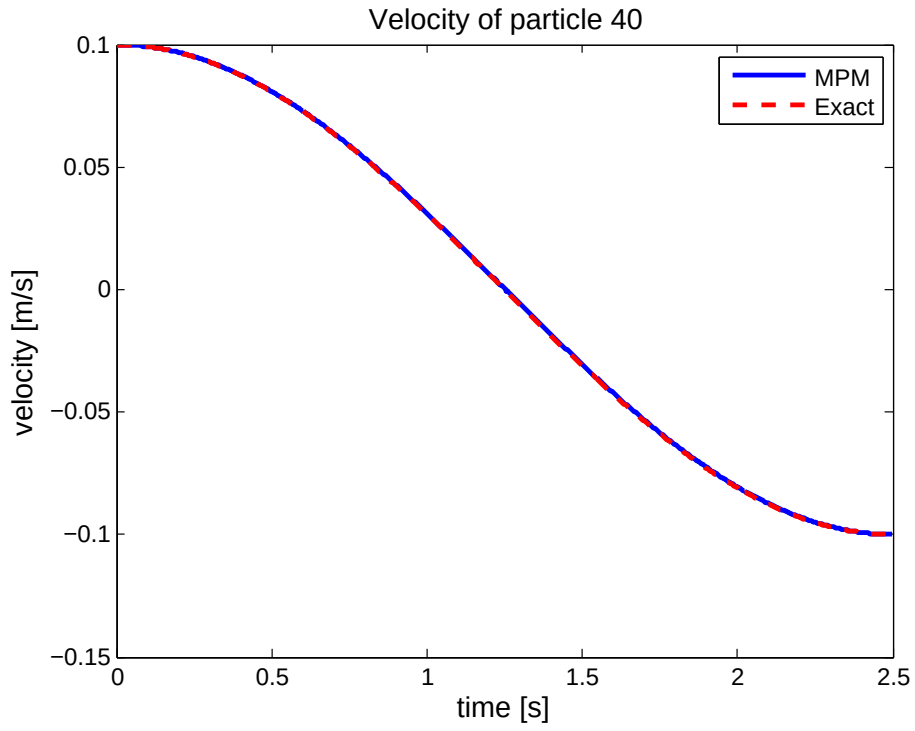

Figure 3.4: Velocity of the particle situated directly left of the middle of the vibrating string as a function of time with 4 PPC and a time step size of $\Delta t = 1 \cdot 10^{-3}$ s. The number of elements is equal to 20. Grid crossing is not observed.

3.4. SOIL COLUMN UNDER SELF-WEIGHT

3.4.1. DESCRIPTION

The second benchmark problem concerns the movement of a column of soil due to self-weight. Since only displacement in vertical direction is considered, the benchmark reduces to a one-dimensional problem. Figure 3.5 gives a schematic overview of the oedometer test.

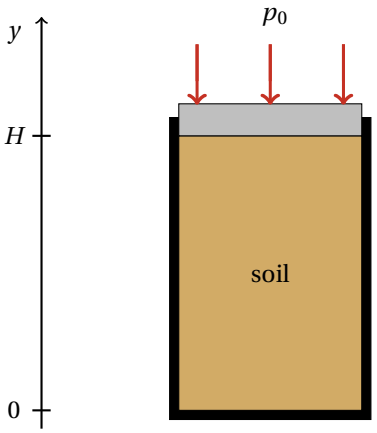

Figure 3.5: The oedometer test.

The second benchmark problem can be modeled by the following equation [8]:

$$\frac{\partial^2 u}{\partial t^2} = \frac{E}{\rho}\frac{\partial^2 u}{\partial y^2} - g,$$

with initial and boundary conditions:

$$\begin{aligned} u(0,t) &= 0, \\ \frac{\partial u}{\partial y}(H,t) &= \frac{p_0}{E}, \\ u(y,0) &= 0, \\ \frac{\partial u}{\partial t}(y,0) &= 0. \end{aligned}$$

The derivation of the analytical solution of this partial differential equation with a non-homogeneous boundary condition can be found in [8] and is given by:

$$u(y,t) = \frac{1}{2}\frac{\rho g y^2}{E} + \frac{(p_0 - \rho g H)y}{E} + \sum_{n=1}^{\infty} u_n \cos\left(\frac{\sqrt{\frac{E}{\rho}}(2n-1)\pi t}{2H}\right)\sin\left(\frac{(2n-1)\pi y}{2H}\right),$$

where

$$u_n = \frac{8H(2\pi p_0 n(-1)^n + 2\rho g H - \pi p_0(-1)^n)}{(4n^2 - 4n + 1)(2n-1)\pi^3 E}.$$

Here, a soil of column subjected to self weight is considered, so $p_0 = 0$. The used parameters are listed in Table 3.3.

Quantity	Symbol	Value	Unit
Density	ρ	1	kg/m^3
Young's modulus	E	$5 \cdot 10^4$	Pa
Gravitational acceleration	g	-9.81	m/s^2
Column height	H	25	m
Load	p_0	0	Pa

Table 3.3: Parameters used to model the soil column subjected to self-weight.

3.4.2. RESULTS

For the second benchmark the spatial convergence is determined. During the simulations, material points crossed the boundary of an element when a relatively small number of elements was used to discretize the spatial domain. These grid crossings effect the quality of the MPM solution.

Figure 3.6 shows the spatial convergence of the material point method when using different numbers of PPC. A time step size Δt of $1 \cdot 10^{-4}$ s was used. Initially, convergence is observed when increasing the number of elements. When a minimum element size is reached, grid crossing has a negative effect on the RMS error. Furthermore, the time discretization error influences the RMS error more when the number of elements is increased. A lack of convergence is observed.

When more material points are defined in each element, the outer material points are situated closer to the element boundary at time $t = 0$ s. Therefore, grid crossing will take place when a lower number of elements is used for spatial disretization, effecting the quality of the MPM solution. This explains why a higher number of PPC leads to an earlier lack of convergence when increasing the number of elements.

However, when the number of elements is further increased, the use of more PPC leads to a lower RMS errror. Hence, increasing the number of particles per cell will eventually decrease the effect of grid crossing. This is consistent with the findings in [11].

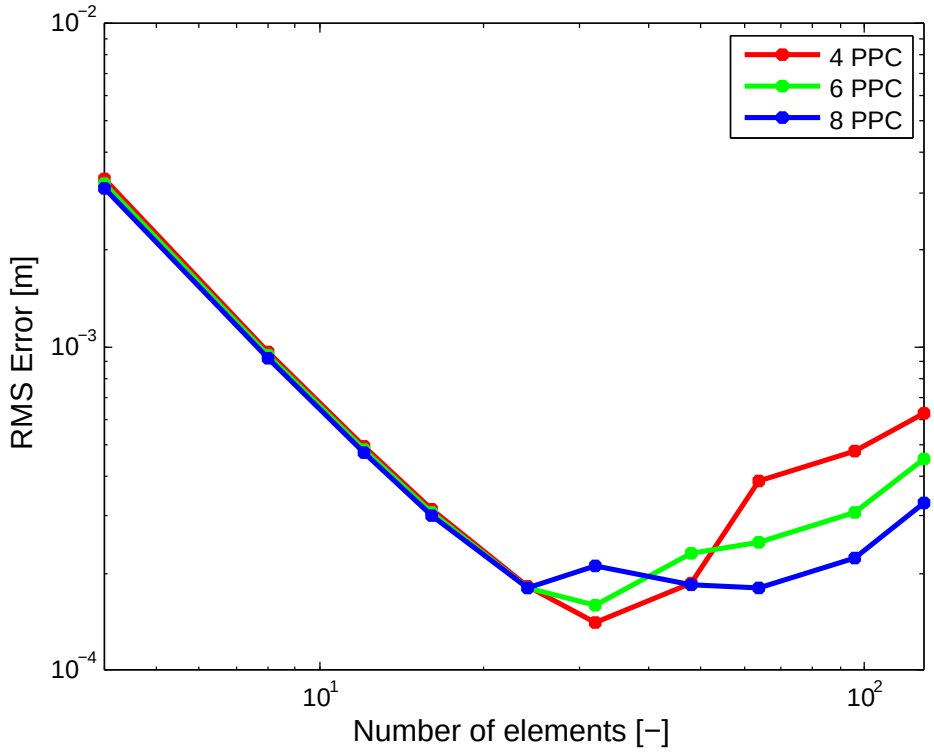

Figure 3.6: Spatial convergence with different numbers of PPC at time $t = 0.2$ s.

To illustrate the effect of grid crossings, we adapted the number of elements from 40 to 80 while keeping the other parameters identical. Figure 3.7 and 3.8 show the velocity of the particle just beneath the middle of the column of soil at $y = 12.5$ m when using respectively 40 and 80 elements. A time step size was chosen of $\Delta t = 1 \cdot 10^{-3}$ s, which is a smaller time step size than the critical time step size . The number of particles per element was set equal to 2.

In case of a grid with 40 elements none of the material points crosses the boundary of an element. The obtained solution is almost identitcal to the analytical solution, except at the horizontal parts of the exact solution. At these parts the MPM solution oscillates around the exact solution. These oscillations were also present with FEM.

By setting the number of elements equal to 80, particles cross the boundary of the elements leading to grid crossing errors. Compared with the MPM solution obtained with 40 elements, the number of oscillations and the amplitude of the oscillations increase drastically. The oscillations occur not only at the horizontal parts of the exact solution, but during the entire time interval.

The effect of grid crossing can also be observed when the displacement of a particle is plotted. Figure 3.9 and 3.10 show the displacement of the particle situated directly below the node with initial position $y = 12.5$ m.

The displacement obtained with the material point method with 40 elements is almost identical to the exact solution. However, when 80 elements are used the obtained displacement differs from the exact solution. Initially, there is a small underestimation of the displacement. The difference between the obtained and exact solution becomes larger as time increases.

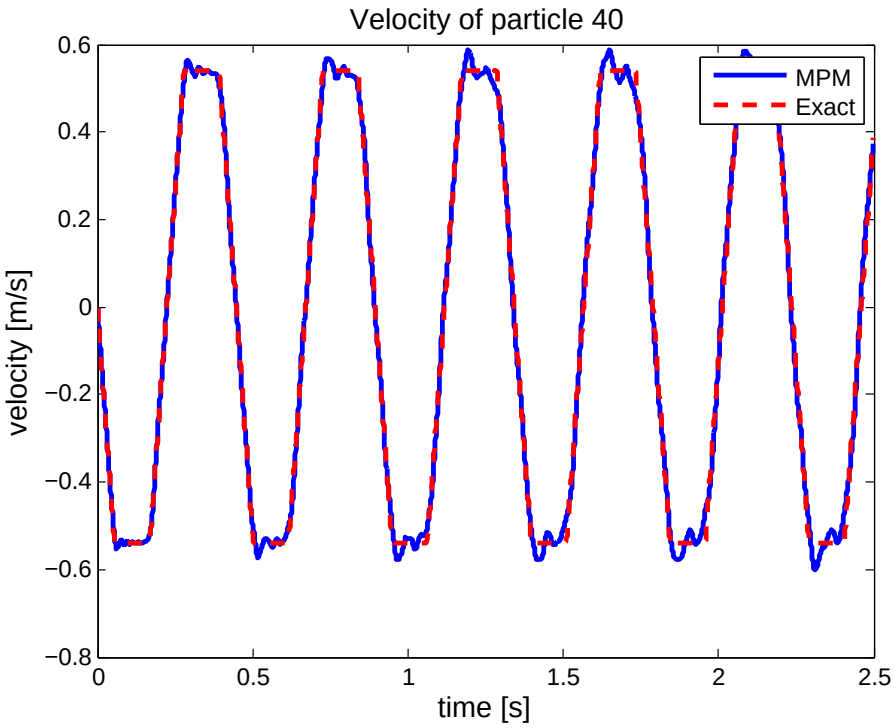

Figure 3.7: Velocity of the particle situated just beneath the middle of the column as a function of time with 2 PPC and a time step size of $\Delta t = 1 \cdot 10^{-3}$ s. The number of elements is equal to 40. Grid crossing is not observed.

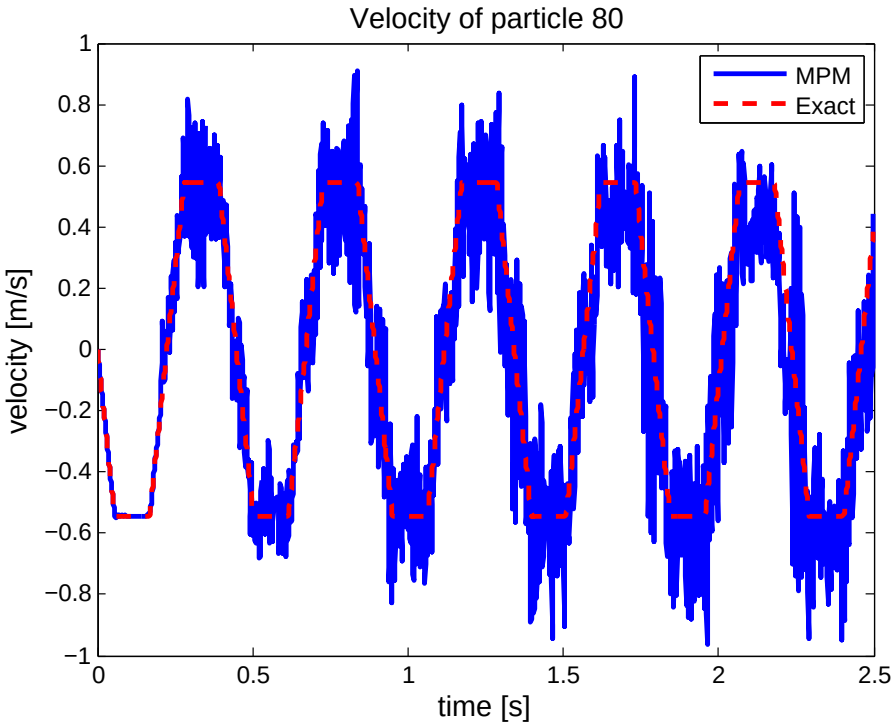

Figure 3.8: Velocity of the particle situated just beneath the middle of the column as a function of time with 2 PPC and a time step size of $\Delta t = 1 \cdot 10^{-3}$ s. The number of elements is equal to 80. Grid crossing is observed.

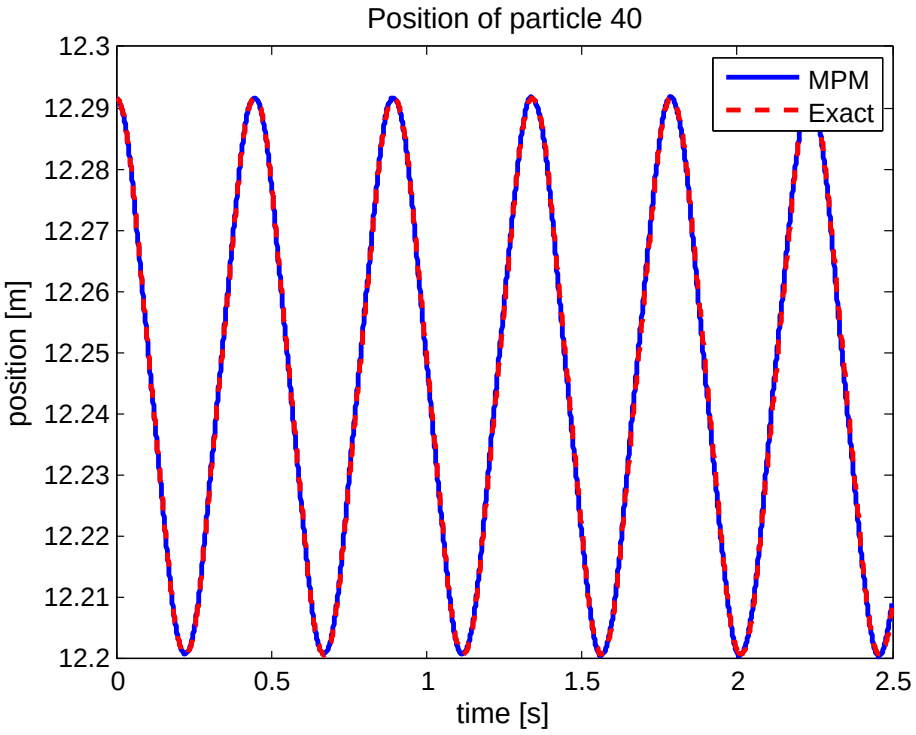

Figure 3.9: Displacement of the particle situated just beneath the middle of the column as a function of time with 2 PPC and a time step size of $\Delta t = 1 \cdot 10^{-3}$ s. The number of elements is equal to 40. Grid crossing is not observed.

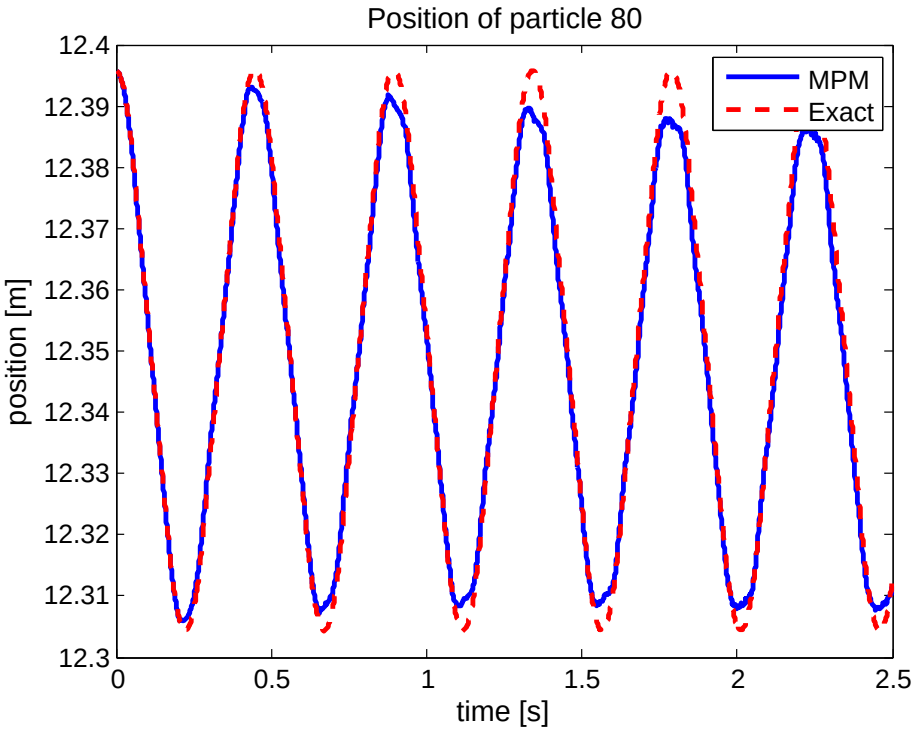

Figure 3.10: Displacement of the particle situated just beneath the middle of the column as a function of time with 2 PPC and a time step size of $\Delta t = 1 \cdot 10^{-3}$ s. The number of elements is equal to 80. Grid crossing is observed.

3.5. VIBRATING BAR WITH DYNAMIC TRACTION

3.5.1. DESCRIPTION

A one dimensional bar is considered with one free end and one fixed end. The bar is driven by the following force function applied at the free end:

$$q(x,t) = \delta(x-L)H(t)\tau\sin\left(\frac{x\pi t}{L}\right).$$

Figure 3.11 gives a schematic overview of the vibrating string problem.

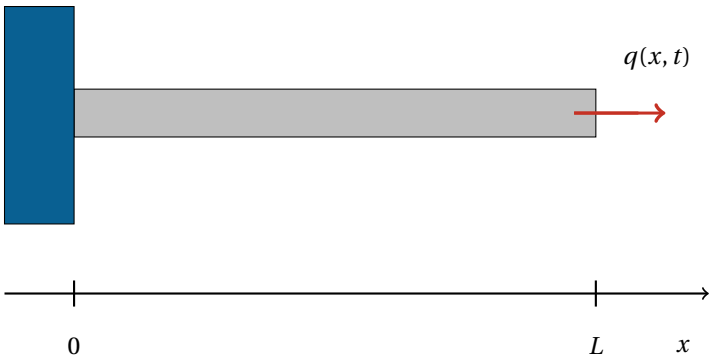

Figure 3.11: The vibrating bar with dynamic traction.

This problem can be modeled by the following partial differential equation

$$\frac{\partial^2 u}{\partial t^2} = \frac{E}{\rho}\frac{\partial^2 u}{\partial x^2}.$$

At time $t = 0$, there is no displacement and no initial velocity. The boundary and initial conditions are

$$\begin{aligned} u(0,t) &= 0, \\ \frac{\partial u}{\partial x}(L,t) &= \frac{\tau}{\rho}\sin\left(\frac{\pi t}{L}\right), \\ u(x,0) &= 0, \\ \frac{\partial u}{\partial t}(x,0) &= 0. \end{aligned}$$

The analytical solution of this partial differential equation can be found in [9]:

$$u(x,t) = \begin{cases} 0 & t \in [0, L-x] \\ \alpha[1+\cos(\omega(t+x))] & t \in [L-x, L+x] \\ \alpha[\cos(\omega(t+x)) - \cos(\omega(t-x))] & t \in [L+x, 3L-x] \\ \alpha[-1-\cos(\omega(t-x))] & t \in [3L-x, 3L+x] \\ 0 & t \in [3L+x, 4L] \end{cases}$$

Where $\alpha = \frac{L\tau}{\rho\pi}$ and $\omega = \frac{\pi}{L}$. An overview of the values of the parameters can be found in Table 3.4

Quantity	Symbol	Value	Unit
Density	ρ	100	kg/m^3
Young's modulus	E	100	Pa
Length	L	1	m
Amplitude force	τ	1	–

Table 3.4: Parameters used for the model of a bar with a free end and a dynamic traction force.

The bar was discretized by placing 3 material points per element equidistantly inside the mesh. The traction force is applied on the material point at the end of the bar and mapped to the nodes of the element the material point is situated in. The traction force at node i can be found by interpolating the force function $q(t)$:

$$\mathbf{F}_i^{\text{int},t} = \phi_i(L + u(L,t))q(L,t)$$

Since the load is applied on a single particle, the internal force vector will only have two non-zero entries. A time step size was chosen $\Delta t = 1 \cdot 10^{-3}$ s.

3.5.2. RESULTS

In [9] a spatial convergence study was performed with different types of shape functions. The spatial convergence with linear shape functions obtained in [9] is compared to the spatial convergence in this study.

As a measure for the error between the exact solution and the obtained solution with the material point method the RMS error was determined at time $t = 1$ s when the displacement of the bar is maximal.

Please note that there are small differences between this study and the one presented in [9] which may influence the results regarding spatial convergence. The time step size Δt was not mentioned in the article. Furthermore, it is not certain how the particles are positioned at $t = 0$ s. Please note that a slightly modified version of the material point method is used in [9], where the nodal velocity is directly updated from the nodal acceleration:

$$v_i^{t+\Delta t} = v_i^t + \Delta t a_i.$$

Since the time step size Δt was not mentioned in the article, the time step size stated above is used. The results can be found in Figure 3.12. For a small number of elements we observe convergence. When increasing the number of elements further, a lack of convergence is observed. The RMS error is slightly higher with the present implementation compared with the errors found in [9]. The observed differences might be caused by the small differences between both this convergence study and the one presented in [9].

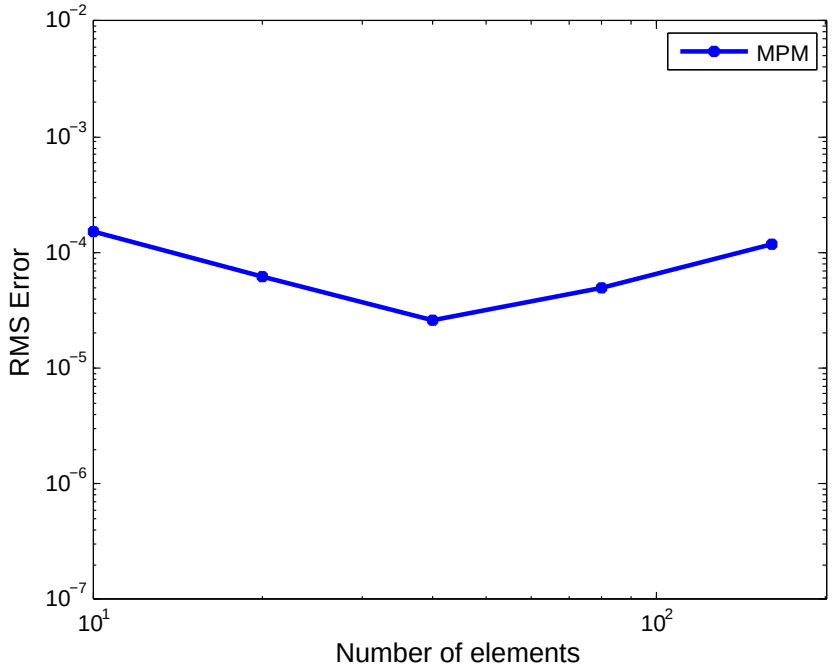

Figure 3.12: Spatial convergence of MPM with 3 PPC and a time step size of $\Delta t = 1 \cdot 10^{-3}$ s.

3.6. CONCLUSIONS

In this chapter three benchmark problems were considered to test the implemented material point method which uses linear basis functions. The first benchmark showed that for small deformations, when the material point almost stay at their original position, the spatial convergence of the material point method is second order. Grid crossings did not occur with the used number of elements to discretize the spatial domain.

The second benchmark showed that for larger deformations, when the material points move through the mesh, the rate of spatial convergence drops even without grid crossing errors. Eventually, grid crossings lead to a lack of convergence. It was shown how grid crossings effect the obtained solution for displacement and velocity of the particles.

In the final benchmark, the RMS errors were slightly higher compared with the errors found in [9]. Small differences between both studies might explain this difference. For a very few number of elements, convergence is observed. Further increasing the number of elements leads to a lack of convergence.

The results illustrate the numerical problems encountered when using linear basis functions within MPM.

4

MOVING LEAST SQUARES

4.1. INTRODUCTION

For small deformations, where the material points almost stay at their original position, the spatial convergence of the material point method is second order according to [2]. However, for large deformations the rate of convergence decreases or even a lack of convergence is observed in [2]. In Chapter 3 similar results were obtained for the benchmark problems. The vibrating string showed second order convergence in space, while a lack of spatial convergence was observed for the soil column under self-weight.

One of the reasons for the lack of convergence when particles become unevenly distributed, is the fact that in the material point method the material points are used as integration points. Consider for example the equation for an arbitrary entry (i, j) of an element matrix:

$$(\mathbf{M}_e^t)_{(i,j)} = \sum_{p=1}^{n_{p,e}} m_p N_i(x_p^t) N_j(x_p^t).$$

If the element mass matrix is lumped, an arbitrary diagonal entry (i, i) can be calculated as follows:

$$(\mathbf{M}_e^t)_{(i,i)} = \sum_{p=1}^{n_{p,e}} m_p N_i(x_p^t).$$

The equation above can be interpreted as the projection of the mass of the particles on the nodes where the weight of each particle is given by the shape function evaluated at the particle position.

Since this projection might affect the spatial convergence when particles become unevenly distributed, an alternative projection might improve the convergence properties of the material point method.

In this chapter we will introduce a projection based on the reconstruction of a function from data points. One can think of reconstructing the velocity field based on the particle velocities. The velocity at the nodes can then simply be obtained by evaluating this velocity field at the nodal points.

To reconstruct a function from data points, we will use the concept of Moving Least Squares (MLS). In the next section, this approach will be explained in more detail. In [2] this projection was used to improve the spatial convergence of the material point method for large deformation problems.

The chapter starts with a general introduction and the derivation of the projection. Furthermore, the spatial convergence properties of this projection and the projection used in MPM will be discussed.

4.2. MOVING LEAST SQUARES APPROACH

The basic idea of the projection is to reconstruct a function based on data points. In this section we will explain how this reconstruction is performed.

Suppose that function u should be approximated on a domain Ω while using only values $u(x_i)$ for $i = \{1,\dots n\}$ are known. It is possible to assign to every value i a subdomain Ω_i such that $x_i \in \Omega_i$ and $\bigcup_{i=1}^{n} = \Omega$. Subdomains might overlap, i.e. it is not required that $\Omega_i \cap \Omega_j = \emptyset$ for all $i, j \in \{1,\dots,n\}$

The purpose is to find an optimal approximation of u denoted by $u^h(x)$ of the form

$$u(x) \approx u^h(x) = \sum_{j=1}^{n} p_j(x) a_j(x) = \mathbf{p}^T(x)\mathbf{a}(x),$$

where p_j are the monomial basis functions and a_j the corresponding coefficients.

The vector $\mathbf{p}$ has the following form:

$$\mathbf{p}^T = [1 \quad x \quad x^2 \quad x^3 \quad \dots \quad x^m]. \tag{4.1}$$

Note that the coefficients $\mathbf{a}(x)$ have not been specified yet. By choosing them in an optimal way the approximation u^h can be obtained. First, it should be specified what we consider as an optimal choice for the coefficients $\mathbf{a}(x)$.

Since the nodal values of the function u are known, this information is used to construct the approximation. Therefore, we define the local approximation function $u^h(x, x_i)$:

$$u^h(x, x_i) = \sum_{j=1}^{n} p_j(x_i) a_j(x) = \mathbf{p}^T(x_i)\mathbf{a}(x),$$

In order to obtain 'the best' projection, the weighted difference between the local approximations and the nodal values is minimized. Hence, the optimal choice of $\mathbf{a}(x)$ can be determined by minimizing the quadratic form

$$\begin{aligned} J &= \sum_{i=1}^{n} w^h(x - x_i)\left[u^h(x, x_i) - u(x_i)\right]^2, \\ &= \sum_{i=1}^{n} w^h(x - x_i)\left[\mathbf{p}^T(x_i)\mathbf{a}(x) - u(x_i)\right]^2, \end{aligned}$$

where $w^h(x - x_i)$ is a weight function depending on a parameter h. The parameter h is a measure for the support of the weight function. The weight function has satisfy certain conditions:

- $w^h(x - x_i) > 0$ for all $x \in \Omega_i$.
- $w^h(x - x_i) = 0$ outside the subdomain Ω_i.
- $\int_\Omega w^h(x - x_i)\mathrm{d}\Omega = 1$.
- The function $w^h(||x - x_i||)$ decreases monotonically.
- $\lim_{h\to 0} w^h(||x - x_i||) = \delta(||x - x_i||)$

In the next section the optimal choice of $\mathbf{a}(x)$ is determined by differentiating the quadratic form and setting it equal to zero. For the determination of the optimal coefficients $\mathbf{a}(x)$ the weight function will not be specified.

Furthermore, it should be noted that this construction makes use of data points. This implies that it does not matter whether the projection is made from the nodes to the particles or vice versa. In both cases an MLS approach can be used. In practice, however, we will use an MLS approach to project from the particles to the nodes.

4.3. MINIMIZING QUADRATIC FORM

To determine the coefficients $\mathbf{a}$ of the approximation u^h, the following quadratic form has to be minimized:

$$J = \sum_{i=1}^{n} w^h(x - x_i) \left[\mathbf{p}^T(x_i)\mathbf{a}(x) - u(x_i)\right]^2.$$

This equation can be written in the following form:

$$J = (\mathbf{Pa} - \mathbf{u})^T \mathbf{W}(\mathbf{Pa} - \mathbf{u}),$$

where

$$\begin{aligned} \mathbf{u}_{(i)} &= u(x_i), \\ \mathbf{P}_{(i,j)} &= p_i(x_j), \\ \mathbf{W}_{(i,i)} &= w^h(x - x_i). \end{aligned}$$

Note that matrix $\mathbf{W}$ is a diagonal matrix and depends on x.

To minimize the quadratic form J, we take the derivative with respect to the coefficient vector $\mathbf{a}$:

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{a}} &= 2(\mathbf{Pa} - \mathbf{u})^T \mathbf{W} \frac{\partial(\mathbf{Pa} - \mathbf{u})}{\partial \mathbf{a}}, \\ &= 2(\mathbf{Pa} - \mathbf{u})^T \mathbf{WP}, \end{aligned}$$

where the fact that $\mathbf{W}$ is symmetric is used. More information about the derivative of a quadratic form can be found in A.6. By taking the transpose of the expression above, we obtain:

$$\begin{aligned} \left(\frac{\partial J}{\partial \mathbf{a}}\right)^T &= 2\left((\mathbf{Pa} - \mathbf{u})^T \mathbf{WP}\right)^T, \\ &= 2\left(\mathbf{P}^T \mathbf{W}(\mathbf{Pa} - \mathbf{u})\right), \\ &= 2\left(\mathbf{P}^T \mathbf{WPa} - \mathbf{P}^T \mathbf{Wu}\right). \end{aligned}$$

Hence, the choice of $\mathbf{a}$ which minimizes J can be obtained by solving

$$\mathbf{P}^T \mathbf{WPa} - \mathbf{P}^T \mathbf{Wu} = \mathbf{0}.$$

To simplify notation, we define the matrix $\mathbf{A}$ and $\mathbf{B}$:

$$\begin{aligned} \mathbf{A} &= \mathbf{P}^T \mathbf{WP}, \\ \mathbf{B} &= \mathbf{P}^T \mathbf{W}. \end{aligned}$$

After substituting these expressions, we obtain the following equation:

$$\mathbf{Aa} - \mathbf{Bu} = \mathbf{0}.$$

Hence, $\mathbf{a}$ is defined by

$$\mathbf{a}(x) = \mathbf{A}^{-1}(x)\mathbf{B}(x)\mathbf{u}.$$

Based on $\mathbf{a}$, the approximation $u^h(x)$ can be constructed. Denote the optimal choice of $\mathbf{a}$ by $\bar{\mathbf{a}}$. Then we can write for the optimal approximation of u:

$$u^h(x) = \mathbf{p}^T(x)\bar{\mathbf{a}}(x) = \mathbf{p}^T(x)\mathbf{A}^{-1}(x)\mathbf{B}(x)\mathbf{u}.$$

4.4. MOVING LEAST SQUARES AND MPM

In the previous section, the optimal approximation of u was derived based on an MLS approach. This approximation will be used to improve the projection in the material point method from particles to the nodes.

By choosing a specific weight function w^h and basis vector $\mathbf{p}$, the projection used in MPM can be reproduced. To illustrate this, recall from the previous section the obtained function u^h:

$$u^h(x) = \mathbf{p}^T(x)\bar{\mathbf{a}}(x) = \mathbf{p}^T(x)\mathbf{A}^{-1}(x)\mathbf{B}(x)\mathbf{u}.$$

Assume that the values of u are known at the particle positions x_p for $p = \{1, \dots n_p\}$. Furthermore, suppose that basis vector $\mathbf{p}$ is just the scalar 1, so $m = 0$ in Equation (4.1). By using the expressions from the previous section, we obtain for the matrix $\mathbf{A}$ and $\mathbf{B}$:

$$\begin{aligned} \mathbf{A}(x) &= \sum_{p=1}^{n_p} w(x - x_p), \\ \mathbf{B}(x)\mathbf{u} &= \sum_{p=1}^{n_p} w(x - x_p)u(x_p). \end{aligned}$$

By using the equations above, the approximation u^h is given by

$$u^h(x) = \mathbf{p}^T(x)\mathbf{A}^{-1}(x)\mathbf{B}(x)\mathbf{u} = \frac{\sum_{p=1}^{n_p} w(x - x_p)u(x_p)}{\sum_{p=1}^{n_p} w(x - x_p)} = \sum_{p=1}^{n_p} \phi_p(x)u(x_p),$$

where

$$\phi_p(x) = \frac{w(x - x_p)}{\sum_{p=1}^{n_p} w(x - x_p)}.$$

To reproduce the projection used in the material point method, choose the following weight function:

$$w^h(x - x_p) = \begin{cases} 1 - \frac{|x - x_p|}{h} & \text{if } \frac{|x - x_p|}{h} \leq 1 \\ 0 & \text{else.} \end{cases}$$

In [2], the convergence properties of the projection used in the material point method was investigated. Based on the known values in the material points, the function $u(x) = \sin(\pi x)$ was approximated and evaluated at the nodal points. In every element three material points were defined to approximate the function u. The error was defined in the following way:

$$E = \sqrt{\sum_{j=1}^{n} h\left(u(x_j) - u^h(x_j)\right)^2},$$

where x_j are the nodal points. To obtain the order of spatial convergence of the projection, the error was determined for decreasing values of h.

When the material points were equally distributed, the order of convergence equals 2. However, when the material points were arbitrarily distributed, the order of convergence dropped to 1.

To improve the projection used in the material point method, the basis vector $\mathbf{p}$ was extended:

$$\mathbf{p} = [1, x].$$

For the weight function, the same function is chosen as with the material point method. For equally as well as arbitrarily distributed material points second order convergence was obtained.

These results show that the spatial convergence of the projection used in MPM can be improved by using a linear basis.

In [2] a linear basis was used to adapt some of the projections from the material points to the nodes. Furthermore, other weight functions were used to define the projection. For each of these projections the changes made to adapt the material point method are discussed.

PARTICLE VELOCITY

In the material point method presented in Section 2.2.4 the nodal velocity is obtained by

$$\mathbf{v}^{t+\Delta t} = \mathbf{M}^{-1}\mathbf{P}^{t+\Delta t},$$

where

$$(\mathbf{P}_e^{t+\Delta t})_i = \sum_{p=1}^{n_{pe}} m_p N_i(x_p^t) v_p^{t+\Delta t}.$$

In the modified version of the material point method, the nodal velocities are directly determined from the particle velocities. First, the velocity field at time $t+\Delta t$ is reconstructed based on the particle velocities:

$$v^{t+\Delta t}(x) = \sum_{p=1}^{n_p} \Phi_p(x)\, v(x_p^{t+\Delta t}),$$

where Φ is the basis function we obtain if we use a cubic spline as weight function:

$$w^h(x-x_p) = \begin{cases} \frac{2}{3} - 4r_h^2 + 4r_h^3 & \text{if } r_h \le \frac{1}{2} \\ \frac{4}{3} - 4r_h + 4r_h^2 - \frac{4}{3}r_h^3 & \text{if } \frac{1}{2} \le r_h \le 1 \\ 0 & \text{else,} \end{cases}$$

where

$$r_h = \frac{|x-x_p)|}{2h}.$$

The nodal velocities can easily be obtained by evaluating the function $v^{t+\Delta t}$ at the nodal points.

PARTICLE DENSITY AND STRESS

These two changes will be discussed together, since the same approach is used for both in [2]. Since the particle densities and stresses are known, their values are used to reconstruct the density and stress field by using a quadratic spline as weight function. If we denote the obtained basis functions by Q_p, the approximated fields can be given by:

$$\rho^t(x) = \sum_{p=1}^{n_p} Q_p(x)\rho^t(x_p^t),$$

$$\sigma^t(x) = \sum_{p=1}^{n_p} Q_p(x)\sigma^t(x_p^t).$$

The weight function used to obtain the basis functions Q_p is given by

$$w^h(x-x_p) = \begin{cases} \frac{3}{4} - 4r_h^2 & \text{if } r_h \le \frac{1}{2} \\ \frac{1}{2}(\frac{3}{2} - r_h)^2 & \text{if } \frac{1}{2} \le r_h \le \frac{3}{2} \\ 0 & \text{else,} \end{cases}$$

where

$$r_h = \frac{|x-x_p)|}{h}.$$

In the original material point method the density and stresses of the material points are used to determine the element matrices and element vectors. The movement of these particles can have a negative effect on the quality of the numerical integration rule used in MPM.

Based on the reconstructed fields, the density and stress are determined at the integration points. The location of these integration points depends on the numerical integration rule that is used. One can think of one point Gauss quadrature or three point Newton Cotes quadrature. The integration rule that is most suited depends on the used basis functions. The important point is that the material points are no longer used as integration points.

During the rest of the master project, this alternative projection will be implemented. It is expected that the described adaptions will improve the spatial convergence when the material points become arbitrary distributed over the domain. The results obtained with this modified projection will be compared with the results described in Chapter 3.

5

1D QUADRATIC MATERIAL POINT METHOD

In Chapter 3 a version of the material point method was discussed which makes use of linear basis functions. The numerical problems due to the discontinuity of the shape functions across element boundaries were illustrated with the results obtained for three benchmark problems.

In this chapter a version of the material point method is discussed which makes use of Lagrange based quadratic basis functions. The chapter start with the introduction of these polynomials. Furthermore, the numerical problems encountered will be discussed. The results obtained with this type of basis functions are compared with the results obtained in Chapter 3.

5.1. SHAPE FUNCTIONS

In contrast to the use of linear basis functions, every element consists of three nodes when using quadratic basis functions. The third node is situated in the middle of an element. Figure 5.1 illustrates the partition of a line segment in three elements.

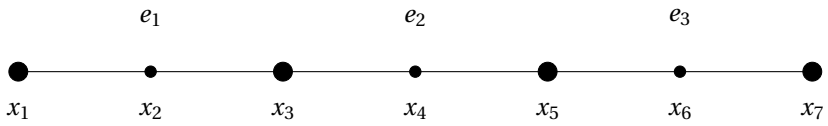

Figure 5.1: Line segment of length L consisting of three elements and seven nodes.

The elements are transformed to a reference element to make the implementation easier. The reference element e_{ref} consists of the three nodes x_1, x_2 and x_3. Three quadratic shape functions are defined on e_{ref} such that

$$N_i(x_j) = \delta_{ij},$$

for $i, j = 1, 2, 3$. Since the values of the shape functions are known at the three nodes, the shape functions are uniquely determined by these nodal values. Hence, on e_{ref} the following shape functions are defined:

$$\begin{aligned} \hat{N}_1(\xi) &= 2\xi^2 - 3\xi + 1, \\ \hat{N}_2(\xi) &= -4\xi^2 + 4\xi, \\ \hat{N}_3(\xi) &= 2\xi^2 - \xi, \end{aligned}$$

where $\xi \in [0, 1]$. The shape functions are shown in Figure 5.2.

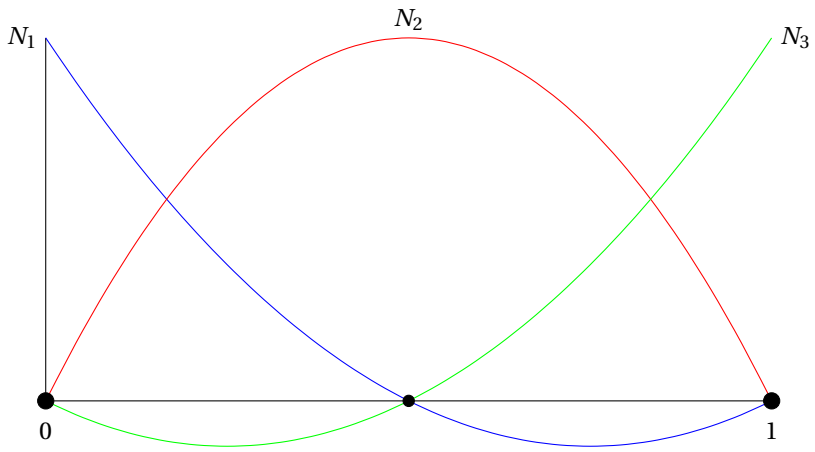

Figure 5.2: Quadratic shape functions defined on reference element e_{ref}.

As with linear basis functions, the shape functions $\hat{N}_i$ are independent of the element size and global position of the particles. To reduce storage space, only the values of the shape functions in the local particle positions are stored. The shape functions possess the partition of unity property, implying that for all $\xi \in [0,1]$ we have

$$\sum_{i=1}^{3} \hat{N}_i(\xi) = 1.$$

From the shape functions associated to a node, a quadratic basis function can be constructed. Figure 5.3 illustrates a basis function associated to a node at the boundary of an element. The basis function ϕ_i associated to a node at the boundary of an element is piecewise quadratic and has compact support on $[x_{i-2}, x_{i+2}]$.

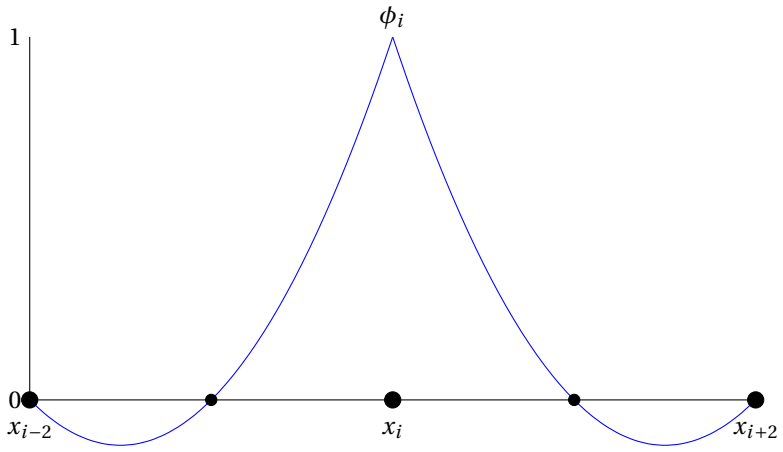

Figure 5.3: Basis function ϕ_i corresponding to node i.

As can be seen in Figure 5.2, the shape functions associated with the nodes at the boundary of an element take negative values on the interval $[0,1]$. Since the values of the shape functions at the particle positions are used to determine the element mass matrices, the global mass matrix might obtain negative valued entries.

Due to these negative values, the lumping procedure described in A.2 can not be applied and solving

$$\mathbf{M}^t \mathbf{a}^t = \mathbf{F}^t.$$

becomes relatively expensive. Furthermore, the negative entries in the mass matrix might cause instability of the solution scheme [12].

5.2. RESULTS

To illustrate the problems that occur when using Lagrange based quadratic basis functions, the first benchmark is considered. The spatial domain is discretized by 4 elements and each element consists of 4 equidistantly placed material points. Table 3.1 provides an overview of the used parameter values.

The (approximated) displacement of an arbitrary point $x_c \in [0, L]$ at time t can be calculated by

$$u_h(x_c, t) = \sum_{i=1}^{n_n} \phi_i(x_c) u_i(t),$$

where n_n denotes the number of nodes. The function u_h was reconstructed pointwisely in each element by 1000 points to obtain a realistic plot of the obtained solution.

Figure 5.5 shows the displacement at time $t = 0.5$ s as a function of x with the use of Lagrange based quadratic basis functions. The obtained solution at time $t = 0.5$ s differs from the exact solution, especially at the points situated in the middle of an element. The displacement obtained with linear basis functions is shown in Figure 5.6. When using linear basis functions the solution differs less from the exact solution.

When the number of elements is increased, stability problems occur with the use of Lagrange based quadratic basis functions. Figure 5.4 demonstrates the displacement of the particle situated directly left of the middle of the string at $x = 12.5$ m.

The obtained results illustrate the problems that occur when using quadratic basis functions based on Lagrange polynomials.

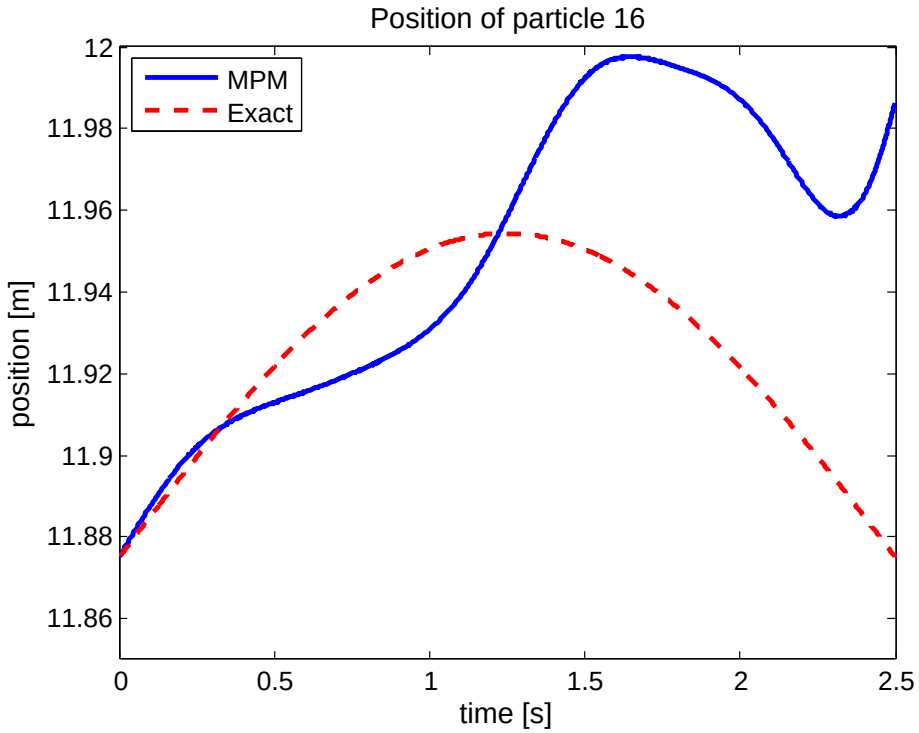

Figure 5.4: Displacement of the particle situated directly left of the middle of the string as a functions of time with 4 PPC and a time step size of $\Delta t = 1 \cdot 10^{-3}$ s. The number of elements equals 8.

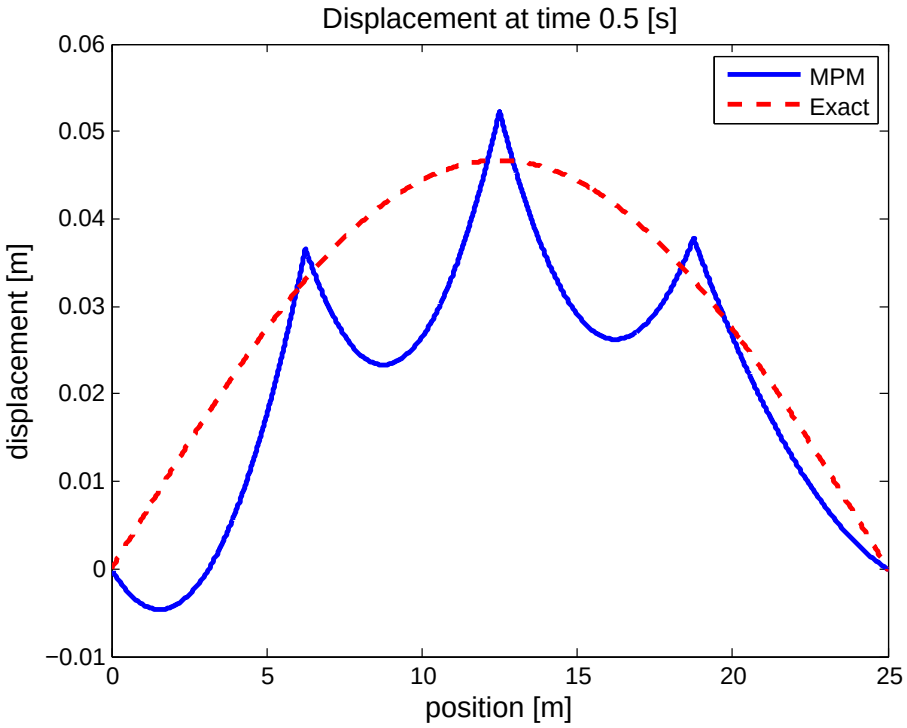

Figure 5.5: Displacement at time $t = 0.5$ s with 4 PPC and a time step size of $\Delta t = 1 \cdot 10^{-3}$ s. 4 elements are defined to discretize the spatial domain. Lagrange based quadratic basis functions are used to approximate the exact solution.

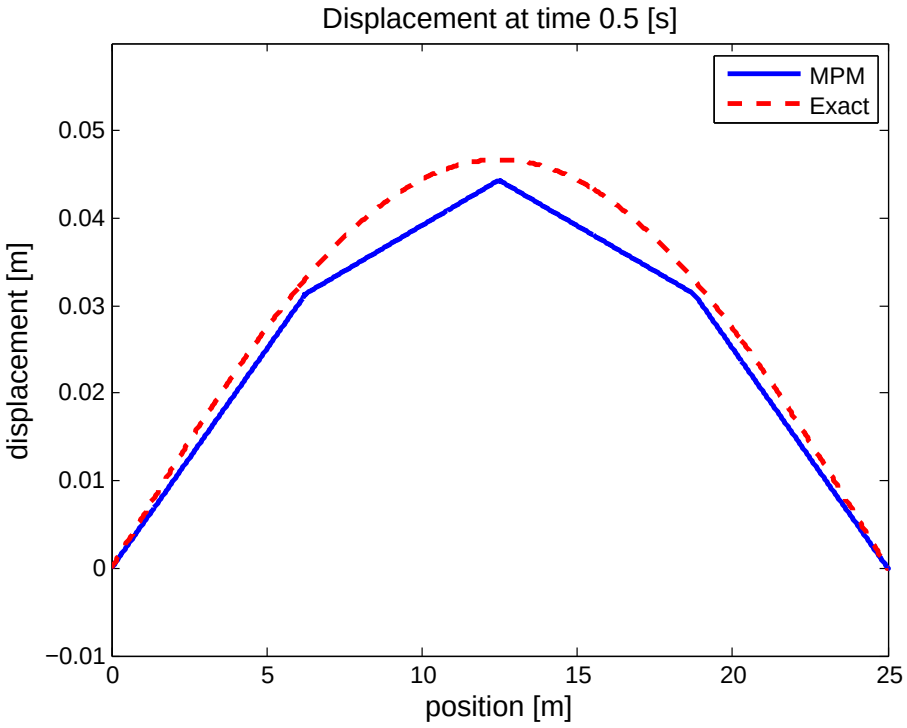

Figure 5.6: Displacement at time $t = 0.5$ s with 4 PPC and a time step size of $\Delta t = 1 \cdot 10^{-3}$ s. 4 elements are defined to discretize the spatial domain. Linear basis functions are used to approximate the exact solution.

6

CONCLUDING REMARKS

In this report the material point method was introduced as an alternative numerical method to solve partial differential equations which involve large deformations. A 1D MPM code has been implemented with Matlab which uses linear basis functions to project the quantities of interest from the nodes to the material points and vice versa.

The projection from the material points to the nodes at the beginning of a time step can be seen as a numerical integration rule. The material points in an element are used as integration points to approximate the integrals over this element. Since the quality of this projection decreases when material points become unevenly distributed, an alternative projection was presented in Chapter 4.

In this report no results are presented of the material point method with the use of this alternative projection. During the rest of this master project the quality of the material point method with this alternative projection will be investigated when using both linear and higher order basis functions.

For the material point method presented in Chapter 2 results were obtained for different benchmark problems while using linear basis functions. In Chapter 3 the numerical problems due to the discontinuity of the basis functions were illustrated in different ways.

As shown in Chapter 5 the use of Lagrange based quadratic basis functions leads to new problems. Due to the negative values obtained by the basis functions, negative entries arrise in the mass matrix leading to stability problems. Furthermore, lumping the mass matrix is no longer possible due to these negative entries leading to longer computation times.

In [9] the use of higher order B-spline basis functions is proposed. During the next months a 1D material point method will be implemented which makes use of quadratic B-spline basis functions. It is expected that the use of these basis functions reduces the problems described in this thesis.

A

APPENDIX

A.1. ASSEMBLY PROCEDURE

In the material point method the mass matrix and force vectors are determined by assembling the element matrices and element vectors. The assemble procedure for mass matrix **M** and force vector **F** are defined by the following formulas:

$$\begin{aligned} \mathbf{M} &= \sum_{el=1}^{n_e} (\mathbf{T}_e)^T \mathbf{M}_e \mathbf{T}_e, \\ \mathbf{F} &= \sum_{el=1}^{n_e} (\mathbf{T}_e)^T \mathbf{F}_e, \end{aligned}$$

where $\mathbf{T}_e$ denotes a Boolean matrix.

We will illustrate the assembly procedure with an example in 1D where linear basis functions are used. Consider a domain consisting of three elements. This implies the existence of three mass matrices, each associated with an element. As an example, we take

$$\mathbf{M}_1 = \begin{bmatrix} 5 & 4 \\ 2 & 7 \end{bmatrix}, \qquad \mathbf{M}_2 = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}, \qquad \mathbf{M}_3 = \begin{bmatrix} 7 & 3 \\ 5 & 2 \end{bmatrix}.$$

By defining the Boolean matrices

$$\mathbf{T}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \qquad \mathbf{T}_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \qquad \mathbf{T}_3 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

we obtain the following global mass matrix

$$\mathbf{M} = \begin{bmatrix} 5 & 4 & 0 & 0 \\ 2 & 8 & 2 & 0 \\ 0 & 2 & 10 & 3 \\ 0 & 0 & 5 & 2 \end{bmatrix}.$$

Since the dimension of the element matrices and vectors depend on the number of nodes per element also the Boolean matrices will differ when using higher-order basis functions.

A.2. LUMPED MATRIX

The lumped mass matrix $\mathbf{M}^L$ is obtained from the consistent mass matrix $\mathbf{M}$ by putting all the weight of a row on the diagonal. Hence, $\mathbf{M}^L$ is defined by

$$\mathbf{M}^L_{(i,i)} = \sum_{j=1}^{n_n} \mathbf{M}_{(i,j)},$$

where n_n denotes the number of nodes. Given the mass matrix $\mathbf{M}$ from the example of the assemble procedure on the previous page, the lumped mass matrix becomes

$$\mathbf{M}^L = \begin{bmatrix} 9 & 0 & 0 & 0 \\ 0 & 12 & 0 & 0 \\ 0 & 0 & 15 & 0 \\ 0 & 0 & 0 & 7 \end{bmatrix}.$$

Recall from Section 2.2.4 that the nodal accelerations are obtained by solving the equations of motion, leading to the following expression:

$$\mathbf{a}^t = (\mathbf{M}^t)^{-1}\mathbf{F}^t.$$

When a lumped mass matrix is used, the inverse of $\mathbf{M}^L$ can easily be obtained and the acceleration of node n can be determined by the following equation:

$$\mathbf{a}_n = \frac{\mathbf{F}_n}{\mathbf{M}^L_{(n,n)}}.$$

In the material point method the particles move through the background mesh. Hence, the mass matrix has to be determined at the beginning of every time step. Therefore, the use of a lumped mass matrix reduces the amount of computations significantly.

The lumping procedure described above is however not used in our code of the one-dimensional material point method. Instead of lumping the consistent mass matrix, a procedure called *direct lumping* is adopted. This procedure makes use of the partition of unity property of the shape functions:

$$\sum_{j=1}^{n_{ne}} N_j(x) = 1,$$

for all $x \in \Omega$. The sum of an arbitrary row i of an element matrix corresponding to element e_k is then given by

$$\sum_{j=1}^{n_{ne}} \int_{e_k} N_i(x)\rho(x)N_j(x)\mathrm{d}x = \int_{e_k} N_i(x)\rho(x)\mathrm{d}x.$$

Hence, when using linear basis functions and applying direct lumping, the element matrix corresponding to element e_k becomes

$$\mathbf{M}_{e_k} = \begin{bmatrix} \int_{e_k} N_1(x)\rho(x)N_1(x)\mathrm{d}x & \int_{e_k} N_1(x)\rho(x)N_2(x)\mathrm{d}x \\ \int_{e_k} N_2(x)\rho(x)N_1(x)\mathrm{d}x & \int_{e_k} N_2(x)\rho(x)N_2(x)\mathrm{d}x \end{bmatrix} = \begin{bmatrix} \int_{e_k} N_1(x)\rho(x)\mathrm{d}x & 0 \\ 0 & \int_{e_k} N_2(x)\rho(x)\mathrm{d}x \end{bmatrix}.$$

A.3. NUMERICAL INTEGRATION IN MPM

In order to obtain an approximation of a definite integral different numerical integration rules can be used. Well known examples of such rules are the *Trapezoidal rule* and the *Midpoint rule*.

In the material point method, the material points are used as integration points. The weight ω_p of each integration point equals the volume of the particle:

$$\omega_p = V_p^t.$$

To illustrate this, consider an arbitrary entry (i, j) of the global mass matrix:

$$\mathbf{M}_{(i,j)} = \sum_{k=1}^{n_e} \int_{e_k} N_i(x)\rho(x)N_j(x)\mathrm{d}x.$$

Each integral over an element e_k is then approximated in the following way:

$$\int_{e_k} N_i(x)\rho(x)N_j(x)\mathrm{d}x \approx \sum_{p=1}^{n_{p,e_k}} \omega_p N_i(x_p^t)\rho(x_p^t)N_j(x_p) = \sum_{p=1}^{n_{p,e_k}} V_p^t N_i(x_p^t)\rho(x_p^t)N_j(x_p^t) = \sum_{p=1}^{n_{p,e_k}} m_p N_i(x_p^t)N_j(x_p^t).$$

The entries of the global force vectors are determined in a similar way. We will illustrate this by considering an arbitrary element of $\mathbf{F}^{\text{grav}}$:

$$\mathbf{F}_i^{\text{grav}} = \sum_{k=1}^{n_e} \int_{e_k} N_i(x)\rho(x)g\mathrm{d}x.$$

Again, the integrals over an element e_k are approximated by using the material points as integration points:

$$\int_{e_k} N_i(x)\rho(x)g\mathrm{d}x \approx \sum_{p=1}^{n_{p,e_k}} \omega_p N_i(x_p^t)\rho(x_p^t)g = \sum_{p=1}^{n_{p,e_k}} V_p^t N_i(x_p^t)\rho(x_p^t)g = \sum_{p=1}^{n_{p,e_k}} m_p g N_i(x_p^t) = \sum_{p=1}^{n_{p,e_k}} f_p^{\text{trac}} N_i(x_p^t).$$

A.4. VALIDATION UPDATED LAGRANGIAN FEM

In order to validate the updated Lagrangian FEM, we considered the oedometer problem, with application of a load p_0 and a graviational acceleration equal to zero. For this case, an analytical solution is available described below.

When considering small deformations, the relation between the applied load p_0 and the strain ϵ is given by

$$\epsilon = \frac{p_0}{E},$$

where E is the Young's modulus. With our standard FEM code, we should be able to obtain the same relation. The strain of the column can be calculated with the following formula:

$$\epsilon = \frac{\Delta H}{H},$$

where H is the initial height of the column and ΔH the change in height.

Under different loads, the strain was determined for the node situated at $y = 0.5$ m. The results can be found in Figure A.1. As expected, a linear relation between load and strain is observed with small-deformation FEM. The slope of the line is equal to the Young's modulus.

For large deformations, the relation between load and strain is given by:

$$\epsilon = \ln\left(1 + \frac{p_0}{E}\right).$$

The strain at an integration point i located at x_i can be determined with the following equation:

$$\epsilon_i^{t+\Delta t} = \epsilon_i^t + \Delta\epsilon_i^{t+\Delta t}.$$

In order to determine the strain for the updated Lagrangian FEM, a damping force was added. The damping force at node i is given by

$$\mathbf{F}_i^{\text{damp},t} = -\alpha \, \text{sgn}(\mathbf{v}_i)|\mathbf{F}_i^{\text{grav},t} + \mathbf{F}_i^{\text{trac},t} - \mathbf{F}_i^{\text{int},t}|,$$

where $\alpha \in [0,1]$ is the damping factor, $|\cdot|$ the absolute value and sgn the singnum function:

$$\text{sgn}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}.$$

By adding this damping force, an equilibrium solution for the column is retrieved for both small and large deformations. A value of 0.7 was chosen for α and the strain was determined under different loads. The relation between the strain and load obtained with the updated Lagrangian FEM code is shown in Figure A.1. As can be observed, there is hardly any difference between the observed and theoretical relation between load and strain.

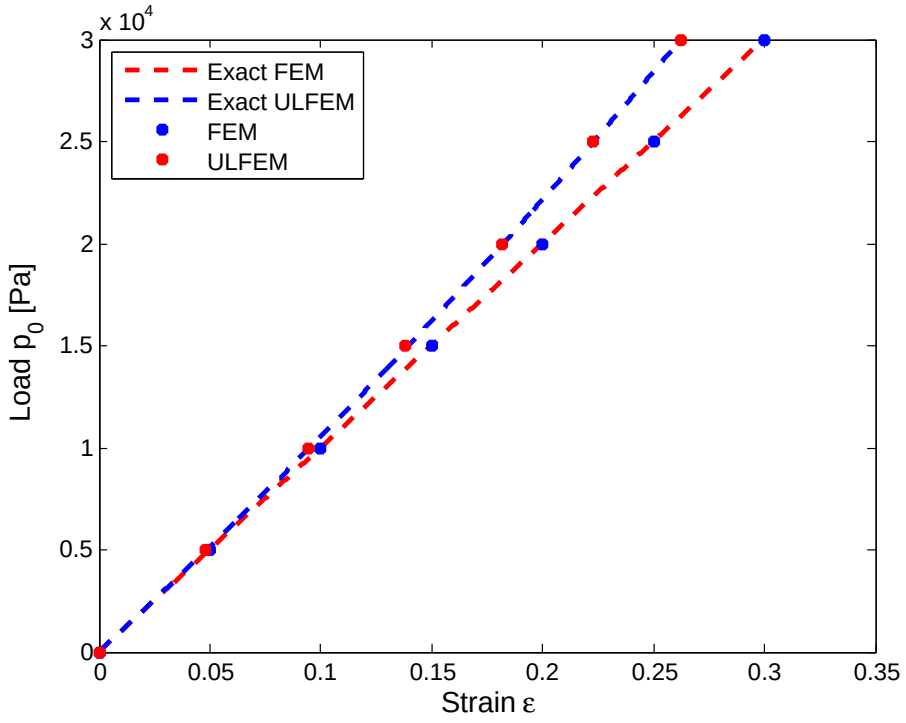

Figure A.1: Relation between load p_0 and strain ϵ with FEM and ULFEM.

A.5. COMPARISON ULFEM AND MPM

For a large deformation problem the solution obtained with MPM is compared with the updated Lagrangian FEM solution. Therefore, the parameter values presented in A.1 will be used to model the second benchmark.

Quantity	Symbol	Value	Unit
Density	ρ	$1 \cdot 10^3$	kg/m^3
Young's modulus	E	$1 \cdot 10^5$	Pa
Gravitational acceleration	g	-9.81	m/s^2
Column height	H	1	m

Table A.1: Parameters used to model the soil column subjected to self-weight.

Figure A.2 shows the position of the node initially situated at $y = 0.5$ m obtained with ULFEM and the displacement of the particle just situated beneath this node with MPM. The grid consist of 30 elements and a time step size was used of $\Delta t = 1 \cdot 10^{-3}$. Since we deal with large deformations, a high number of grid crossing can be expected. In order to reduce the effect of grid crossing, 50 particles per cell are defined when performing the simulation with MPM.

The figure shows that the material point method overestimates the position of the particle. Furthermore, a phase difference between both solutions occurs over time. The period of both solutions was obtained by determining the time between two succesive maxima of the potential energy. For both MPM as ULFEM the period is constant. The period found with MPM is approximately 0.374 s, where the period of the ULFEM solution is around 0.364 s. An overestimation of the exact period of the material point method was also observed in [12].

Both observations seem to imply that with the material point method, the stiffness of the material is underestimated.

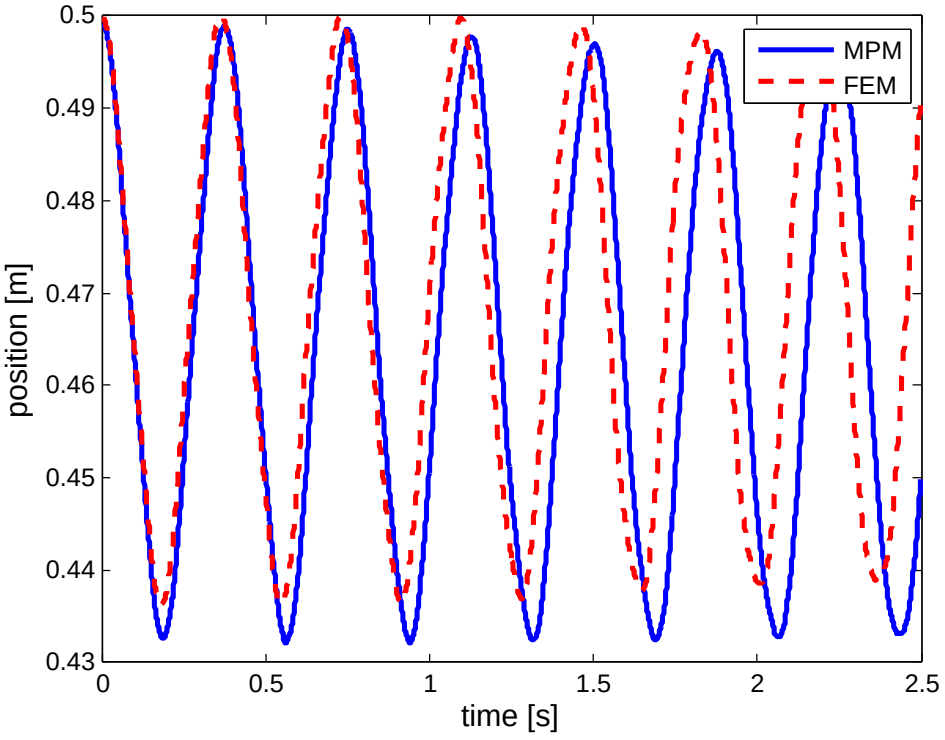

Figure A.2: Solutions obtained with MPM and ULFEM with 30 elements and a time step of $\Delta t = 1 \cdot 10^{-3}$ s.

A.6. DERIVATIVE OF QUADRATIC FORM

To determine the optimal choice of coefficients $\mathbf{a}$, the following quadratic form has to be minimized:

$$J = (\mathbf{Pa}-\mathbf{u})^T\mathbf{W}(\mathbf{Pa}-\mathbf{u}).$$

We can write this quadratic form as

$$J = \mathbf{y}^T\mathbf{Ay},$$

where $\mathbf{y}$ is a vector and $\mathbf{A}$ a symmetric matrix. It must be noted that only $\mathbf{y}$ depends on $\mathbf{a}$. Before we determine the derivative of J, some notation has to be introduced.

We denote the derivative of the vector $\mathbf{y} \in \mathbb{R}^{n\times 1}$ with respect to the vector $\mathbf{a} \in \mathbb{R}^{m\times 1}$ by the matrix $\frac{\partial \mathbf{y}}{\partial \mathbf{a}} \in \mathbb{R}^{n\times m}$, where entry (i,j) corresponds to the derivative of component i of $\mathbf{y}$ with respect to component j of $\mathbf{a}$. This results in the following notation:

$$\frac{\partial \mathbf{y}}{\partial \mathbf{a}} = \begin{pmatrix} \frac{\partial y_1}{\partial a_1} & \frac{\partial y_1}{\partial a_2} & \cdots & \frac{\partial y_1}{\partial a_m} \\ \frac{\partial y_2}{\partial a_1} & \frac{\partial y_2}{\partial a_2} & \cdots & \frac{\partial y_2}{\partial a_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_n}{\partial a_1} & \frac{\partial y_n}{\partial a_2} & \cdots & \frac{\partial y_n}{\partial a_m} \end{pmatrix}.$$

The derivative of a scalar function J with respect to a vector $\mathbf{a} \in \mathbb{R}^{m\times 1}$ is then denoted by the vector $\frac{\partial J}{\partial \mathbf{a}}$:

$$\left(\frac{\partial J}{\partial \mathbf{a}}\right)^T = \begin{pmatrix} \frac{\partial J}{\partial a_1} \\ \frac{\partial J}{\partial a_2} \\ \vdots \\ \frac{\partial J}{\partial a_m} \end{pmatrix}.$$

To determine the derivative of this quadratic form J, we need the following proposition.

Proposition: *Let the scalar function J be defined by:*

$$J = \mathbf{y}^T\mathbf{x},$$

where $\mathbf{y}$ and $\mathbf{x} \in \mathbb{R}^{n\times 1}$. If $\mathbf{y}$ and $\mathbf{x}$ are both functions of the vector $\mathbf{a}$, then the derivative of J with respect to $\mathbf{a}$ is given by:

$$\frac{\partial J}{\partial \mathbf{a}} = \mathbf{x}^T\frac{\partial \mathbf{y}}{\partial \mathbf{a}} + \mathbf{y}^T\frac{\partial \mathbf{x}}{\partial \mathbf{a}}.$$

Proof. When we write J as

$$J = \sum_{i=1}^{n} x_i y_i,$$

the derivative of J with respect to a_k for all $k \in \{1,\ldots,n\}$ is given by

$$\frac{\partial J}{\partial a_k} = \sum_{i=1}^{n} x_i\frac{\partial y_i}{\partial a_k} + y_i\frac{\partial x_i}{\partial a_k} = \mathbf{x}^T\frac{\partial \mathbf{y}}{\partial a_k} + \mathbf{y}^T\frac{\partial \mathbf{x}}{\partial a_k}.$$

Hence, we obtain

$$\frac{\partial J}{\partial \mathbf{a}} = \mathbf{x}^T\frac{\partial \mathbf{y}}{\partial \mathbf{a}} + \mathbf{y}^T\frac{\partial \mathbf{x}}{\partial \mathbf{a}}.$$

□

Using this proposition we determine the derivative of the quadratic form encountered with the MLS approach.

Proposition: *Let the quadratic form J be defined by:*

$$J = \mathbf{y}^T \mathbf{A}\mathbf{y},$$

where $\mathbf{y} \in \mathbb{R}^{n\times 1}$ and $\mathbf{A} \in \mathbb{R}^{n\times n}$. If $\mathbf{y}$ is a function of the vector $\mathbf{a}$ and $\mathbf{A}$ is symmetric, then the derivative of J with respect to $\mathbf{a}$ is given by:

$$\frac{\partial J}{\partial \mathbf{a}} = 2\mathbf{y}^T \mathbf{A}\frac{\partial \mathbf{y}}{\partial \mathbf{a}}.$$

Proof. Let $\mathbf{w}^T = \mathbf{y}^T\mathbf{A}$. We obtain for the quadratic form J:

$$J = \mathbf{w}^T\mathbf{y}.$$

By making use of the previous proposition, we can write for derivative of J with respect to $\mathbf{a}$:

$$\begin{aligned}
\frac{\partial J}{\partial \mathbf{a}} &= \mathbf{y}^T\frac{\partial \mathbf{w}}{\partial \mathbf{a}} + \mathbf{w}^T\frac{\partial \mathbf{y}}{\partial \mathbf{a}}, \\
&= \mathbf{y}^T\frac{\partial \mathbf{A}^T\mathbf{y}}{\partial \mathbf{a}} + \mathbf{y}^T\mathbf{A}\frac{\partial \mathbf{y}}{\partial \mathbf{a}}, \\
&= \mathbf{y}^T\mathbf{A}^T\frac{\partial \mathbf{y}}{\partial \mathbf{a}} + \mathbf{y}^T\mathbf{A}\frac{\partial \mathbf{y}}{\partial \mathbf{a}}, \\
&= 2\mathbf{y}^T\mathbf{A}\frac{\partial \mathbf{y}}{\partial \mathbf{a}}.
\end{aligned}$$

□

BIBLIOGRAPHY

[1] D. Sulsky, Z. Chen, and H. L. Schreyer, *A particle method for history-dependent materials,* Computer methods in applied mechanics and engineering **118**, p. 179 (1994).

[2] M. Gong, *Improving the Material Point Method*, Ph.D. thesis, The University of New Mexico (2015).

[3] I. AL-Kafaji, *Formulation of a dynamic material point method (MPM) for geomechanical problems*, Ph.D. thesis, Institut für Geotechnik der Universität Stuttgart (2013).

[4] F. Harlow, *The particle-in-cell computing method for fluid dynamics,* Methods for Computational Physics **3**, p. 319 (1964).

[5] J. Brackbill and H. Ruppel, *Flip: a method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions.* Journal of Computational Physics **65**, p. 314 (1994).

[6] B. J. Sulsky, D. and D. Burgess, *Mass matrix formulation of the flip particle-in-cell method,* .

[7] D. Sulsky and H. Schreyer, *Axisymmetric form of the material point method with applications to upsetting and taylor impact problems.* Computer Methods in Applied Mechanics an Engineering **139**, p. 409 (1996).

[8] M. M. J. Mieremet, *Numerical stability for velocity-based 2-phase formulation for geotechnical dynamic analysis*, Tech. Rep. (Delft Institute of Applied Mathematics, 2015).

[9] M. Steffen, R. Kirby, and M. Berzins, *Analysis and reduction of quadrature errors in the material point method (mpm).* International journal for numerical methods in engineering **76**, p. 922 (2008).

[10] R. Haberman, *Applied Partial Differential Equations with Fourier Series and Boundary Value Problems* (Prentice Hall, 2004).

[11] S. Bardenhagen and M. Kober, *The generalized material point method.* Computer Modeling in Engineering and Sciences **5**, p. 477 (2004).

[12] S. Andersen and L. Andersen, *Analysis of spatial interpolation in the material-point method.* Computers and Structures **88**, p. 506 (2010).