TUDelft

Delft University of Technology

Algorithmic improvements of the material-point method and Taylor least-squares function reconstruction

Wobbes, Lisa

DOI

10.4233/uuid:e4b0ddb1-26d8-4ba6-8c4d-4894cd77b2be

Publication date
2019

Document Version
Final published version

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

ALGORITHMIC IMPROVEMENTS OF THE MATERIAL-POINT METHOD AND TAYLOR LEAST-SQUARES FUNCTION RECONSTRUCTION

ALGORITHMIC IMPROVEMENTS OF THE MATERIAL-POINT METHOD AND TAYLOR LEAST-SQUARES FUNCTION RECONSTRUCTION

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus prof. dr. ir. T.H.J.J. van der Hagen
chair of the Board for Doctorates
to be defended publicly on
Wednesday 18 December 2019 at 10:00 o'clock

by

Elizaveta Dmitrievna WOBBES

Master of Science in Applied Mathematics, Delft University of Technology, the
Netherlands
born in Ryazan, the Union of Soviet Socialist Republics

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus	chairman
Prof. dr. ir. C. Vuik	Delft University of Technology, promotor
Dr. M. Möller	Delft University of Technology, copromotor

Independent members:

Prof. dr. M. Berzins	The University of Utah, the United States of America
Prof. dr. A. M. Pandolfi	Polytechnic University of Milan, Italy
Prof. dr. C. Augarde	Durham University, the United Kingdom
Prof. dr. M. Hicks	Delft University of Technology
Prof. dr. ir. A. W. Heemink	Delft University of Technology

An electronic version of this dissertation is available at
`http://repository.tudelft.nl/`.

CONTENTS

SUMMARY

The material-point method (MPM) is a continuum-based numerical tool to simulate problems that involve large deformations. Within MPM, a continuum is discretized by defining a set of Lagrangian particles, called material points, which store all relevant material properties. The method adopts an Eulerian background grid, where the equations of motion are solved at every time step. The solution on the background grid is used to subsequently update all material-point properties, such as displacement, velocity, and stress. In this way, MPM incorporates both Eulerian and Lagrangian descriptions. Similarly to other combined Eulerian-Lagrangian techniques, MPM attempts to avoid the numerical difficulties arising from nonlinear convective terms associated with an Eulerian problem formulation, while preventing grid distortion, typically encountered within meshbased Lagrangian formulations.

Over the years, MPM has been successfully applied to many complex problems from engineering and computer graphics. Despite its impressive performance for these applications, the method still suffers from several numerical shortcomings, such as stability issues, inaccurate mapping of the material-point data, and unphysical oscillations that arise when material points travel from one element to another, the so-called grid crossing errors. This dissertation provides an overview of the existing literature that addresses these drawbacks, and introduces new mathematical techniques that improve the performance of MPM.

Previous studies have indicated that the use of higher-order B-spline basis functions within MPM mitigates the grid-crossing errors, thereby improving the accuracy of the method. This thesis combines the B-spline approach, known as BSMPM, with an alternative technique to project the information from material points to the background grid. The mapping technique is based on cubic-spline interpolation and Gauss quadrature. The numerical results show that the proposed approach further increases the accuracy of the method and leads to higher-order convergence. Moreover, the extension of BSMPM to unstructured grids using Powell-Sabin splines is discussed.

After that, this dissertation compares MPM to the optimal transportation meshfree (OTM) method. Both MPM and the OTM method have been developed to efficiently solve partial differential equations that arise from the conservation laws in continuum mechanics. However, the methods are derived in a different fashion and have been studied independently of one another. This thesis provides a direct step-by-step comparison of the MPM and OTM algorithms. Based on this comparison, the conditions, under which the two approaches can be related to each other, are derived, thereby bridging the gap between the MPM and OTM communities. In addition, the thesis introduces a novel unified approach that combines the design principles from BSMPM and the OTM method. The proposed approach is significantly cheaper and more robust than the standard OTM method and allows for the use of a consistent mass matrix without stability issues that are typically encountered in MPM computations.

Finally, this thesis introduces a novel function reconstruction technique that combines the well-known least-squares method with local Taylor basis functions, called Taylor least squares (TLS). The technique reconstructs functions from scattered data, while preserving their integral values. In conjunction with MPM or a related method, the TLS technique locally approximates quantities of interest, such as stress and density, and when used with a suitable quadrature rule, conserves the total mass and linear momentum after transferring the material-point information to the grid. The integration of the technique into MPM, dual domain material-point method (DDMPM), and BSMPM significantly improves the results of these methods. For the considered one-dimensional examples, the TLS function reconstruction technique resembles the approximation properties of the highly-accurate cubic-spline reconstruction, while preserving the physical aspects of the standard algorithm.

I

INTRODUCTION AND BACKGROUND

1

INTRODUCTION

This chapter gives a brief introduction to this dissertation. Section 1.1 provides the motivation for the conducted research. Section 1.2 presents its main objectives. Section 1.3 describes the structure of the dissertation.

1

1.1. MOTIVATION

The material-point method (MPM) [1, 2] is a numerical tool to simulate problems that involve large deformations. In MPM, the material is represented by a set of particles that move through a fixed background grid. These particles are referred to as material points. They carry the physical properties of the continuum such as the mass, strain, and stress. Similarly to the finite element method (FEM) [3], the discretized governing equations are assembled and subsequently solved at the background grid. Essentially, within each time step throughout the simulation, MPM performs the following solution procedure. First, the information available at the material points is projected to the nodes of the background grid. Then, the discretized governing equations are assembled and solved on the grid nodes. Finally, the obtained information is mapped back to update the material points.

MPM has proven to be successful in solving complex engineering problems that involve large strains, multi-phase interactions, and history-dependent material bahavior. Over the years, MPM has been applied to a wide range of applications, including modeling of failure phenomena in single- and multi-phase media [4–9], pile installation [10], crack growth [11, 12], fluid-structure and fluid-membrane interaction [13–16], snow and ice dynamics [17–19], explosion and shock waves [20, 21], metal forming [22], and soft tissue damage [23]. Figure 1.1 provides application examples of MPM.
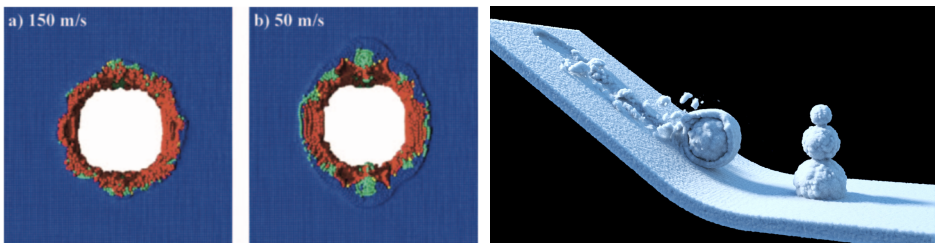
Figure 1.1: Modeling of soft tissue damage [23] (left) and snow dynamics [17] (right) using MPM.

Despite its impressive performance for a variety of computer graphics and engineering problems, MPM still suffers from several shortcomings. For example, when material points travel from one cell to another, they generate unphysical oscillations in the forces, frequently referred to as grid-crossing error [24]. This is due to the use of piecewise-linear basis functions, whose gradients are discontinuous on element boundaries. The direct mapping of the scattered material-point data to the background grid can also introduce significant numerical inaccuracies [25, 26]. In addition, MPM contains FEM-type type errors originating from mass-lumping and interpolation, as well as time-stepping errors [27, 28].

Due to the complex interplay between the background grid and particles within the method, the application of theories developed for closely related FEM is problematic. Therefore, MPM lacks a generic framework to analyze its numerical artifacts. Nevertheless, much research has been conducted to overcome the numerical shortcomings of MPM. For instance, methods such as the generalized interpolation material-point

1

(GIMP) method [24], the convected particle-domain interpolation (CPDI) method [29], the dual domain material-point method (DDMPM) [30], and the B-spline material-point method (BSMPM) [28, 31] have been designed to overcome the grid-crossing error.

1.2. DISSERTATION OBJECTIVES

This dissertation has the following objectives:

- *Overview of the existing development trends within MPM.* A significant part of existing research on MPM is dedicated to further development and improvement of the method. It includes studies on time integration within MPM, its formulation, and reduction of grid-crossing errors. However, to the author's best knowledge, no detailed overview of such studies has been provided in the literature yet. Therefore, this thesis aims to identify the main development courses within MPM research and categorize the existing studies accordingly.

- *Connection to meshfree methods.* MPM originates from particle methods, but is frequently classified as a meshfree method (see Chapter 2 for more details). This dissertation highlights the connection between MPM and meshfree methods by showing a close relation between MPM and the optimal transportation meshfree (OTM) method [32], which is a genuinely meshfree method. This study bridges the gap between the MPM and OTM communities. In addition, it proposes a possible way to unify BSMPM and the OTM method.

- *Further development of BSMPM.* BSMPM provides a fundamental solution to the grid-crossing issue in MPM by replacing piecewise-linear basis functions with at least C^1-continuous higher-order B-spline basis functions. This dissertation further improves the accuracy of BSMPM by combining the method with advanced techniques for mapping the material-point information to the degrees of freedom (DOFs). Moreover, it discusses the extension of BSMPM to unstructured grids, which typically originate from complex real-life problems.

- *Development of an advanced technique to reconstruct functions from scattered data points and its application to MPM and related methods.* As was mentioned previously, numerical shortcomings of MPM are partially caused by the direct mapping of material-point data to the background grid. Although a modified mapping generally improves the accuracy of the MPM solution, the standard reconstruction techniques might lead to the loss of physical properties of the method. In fact, while MPM projection conserves the mass and linear momentum of the system, most standard mapping techniques do not guarantee that. For this reason, this thesis introduces a novel technique to reconstruct functions from scattered data points, which preserves the a priori known integral values. This allows for the conservation of the mass and linear momentum of the system when material-point information is mapped to the background grid within MPM. It should be noted that despite being developed within the MPM framework, the proposed technique can be applied as a conservative projection scheme to problems unrelated to MPM or meshfree methods.

1

1.3. DISSERTATION OUTLINE

This dissertation consists of four parts and is divided into seven chapters:

Part I: Introduction and background

Chapter 1: Introduction
An introuction to the dissertation is provided.
Chapter 2: Overview of the material-point method
MPM is introduced and an overview of studies that further develop and improve the method is given.

Part II: Methodological improvements of MPM

Chapter 3: Further development of the B-spline material-point method
First, BSMPM is combined with an advanced technique to map the material-point data to the background grid. After that, the extension of the method to unstructured grids is discussed.
Chapter 4: Comparison and unification with optimal transportation mesh-free method
The OTM method is introduced. Conditions, under which the MPM and OTM algorithms can be related, are identified. A possible manner to unify the methods is proposed.

Part III: Taylor least-squares function reconstruction and its application to MPM

Chapter 5: Conservative Taylor least-squares reconstruction with application to material-point methods
A novel function reconstruction technique, called Taylor least squares (TLS), is proposed. The TLS technique is applied within one-dimensional MPM computations.
Chapter 6: Discussion: Extension of Taylor least-squares function reconstruction technique to two-dimensional problems and its application to the material-point method
The TLS technique is extended to two-dimensional problems. Issues with its application within two-dimensional MPM simulations are discussed.

Part IV: Closure

Chapter 7: Conclusions
The conclusions are drawn and outlook to further research is provided.

The notation used throughout this dissertation is introduced in the first part of Chap-

ter 2. For an advanced reader, who is familiar with BSMPM, Chapter 3 provides additional insights into it, but is not essential for the understanding of other chapters. Furthermore, Parts II and III of the dissertation can be considered independently of one another.

1

2

OVERVIEW OF THE
MATERIAL-POINT METHOD

This chapter introduces the material-point method (MPM) and gives the details of its single-phase formulation. Although multi-phase formulations are generally used for the simulation of complex materials, such as saturated soils, the single-phase formulation provides sufficient complexity to analyze the method from the mathematical point of view. The chapter is structured as follows. Section 2.1 explains the main physical assumptions behind MPM, places it in the context of meshfree methods, and gives a short description of its predecessors. Section 2.2 introduces the governing equations that can capture the deformation of a one-phase material. Section 2.3 discretizes these equations in space and time. Section 2.4 provides the original MPM algorithm as well as a modified version. Section 2.5 describes the basis functions that are typically adopted for MPM computations. Finally, Section 2.6 attempts to categorize the fundamental research on MPM in a number of groups. Sections 2.6.1 to 2.6.4 are devoted to the main development trends in the MPM community, while Section 2.6.5 describes a number of techniques that combine different research areas as well as improvement strategies that do not fall within the general research trends from the previous sections. Although this chapter discusses some applications of MPM and related methods, this topic forms a voluminous subject and is outside the scope of this thesis.

2.1. Background

2.1.1. Continuum-based framework

As was mentioned previously, the material-point method is a numerical method that was designed to simulate materials that undergo large deformations. Although in reality, any matter is composed of discrete molecules and atoms, the physical model of MPM is based on the assumption that matter is distributed continuously throughout the occupied space. This hypothetical material is called a *continuum*. Alternatively, matter can be treated discretely as a collection of separate blocks or particles. Discrete modeling techniques, such as the molecular dynamics [33, 34] method and the discrete element method [35], are particularly suitable for simulations where the problem size and discontinuities of the material are on a comparable scale. However, discrete models of macroscopic problems are computationally expensive. Therefore, continuum-based methods are typically used for large-scale simulations.

The deformation of a continuum is captured by means of its *kinematical description* (e.g., [36]). Two classical descriptions frequently used in continuum mechanics are the *Lagrangian* and *Eulerian* descriptions. The Eulerian description fixes attention on a given region of space and considers spatial coordinates and time as independent variables. In the Lagrangian description, the independent variables are time and material coordinates. This implies that an individual particle of the continuum is followed through space and time. Traditionally, this is envisioned by a boat drifting down a river. In the Lagrangian description, the flow is observed from the boat, while in the Eulerian description the observation point is a fixed place on the river bank. Furthermore, depending on the frame of reference, two formulations can be developed in the Lagrangian framework. The *total Lagrangian* formulation describes the motion with respect to the initial material coordinates, whereas the *updated Lagrangian* formulation uses the updated material coordinates as the frame of reference.

When numerical methods rely on the computational mesh (or grid) and are based on the purely Lagrangian description, each node of the mesh follows the corresponding material particle during the motion. Such techniques are mainly applied to problems in structural mechanics. The Lagrangian description allows these methods to impose boundary conditions and track free surfaces and interfaces between materials in a straightforward manner. They also facilitate the treatment of history-dependent materials [37]. The main disadvantage of these so-called *meshbased* or *meshdependent* Lagrangian methods is that they require frequent remeshing to follow very large deformations of the continuum. This shortcoming is avoided by meshbased methods based on the Eulerian description, because they fix the computational grid, while the continuum moves through it. However, the treatment of boundary conditions and the definition of free surfaces and interfaces between materials become significantly more complex. Moreover, due to the relative motion between the computational mesh and deforming continuum, governing equations of Eulerian methods contain convective terms. According to Donea et al. [38], the non-symmetric structure of the convection operators can lead to numerical difficulties. Despite this, Eulerian methods are widely used to model fluid dynamics.

While most continuum-based numerical techniques can be categorized as either Lagrangian or Eulerian methods, some techniques, such as MPM and the arbitrary La-

grangian-Eulerian (ALE) methods [39], combine the Eulerian and Lagrangian approaches in an attempt to incorporate their best features and minimize their drawbacks. The main idea behind the ALE methods is that the nodes of the computational mesh may move together with the continuum as in purely Lagrangian methods, but they can also remain fixed following the Eulerian approach or even be moved in an arbitrarily specified way. For instance, the nodes belonging to the boundary of the material can be moved together with it, while the nodes describing the internal part of the continuum might move arbitrarily to optimize the shapes of the elements. The ALE approach is adopted in the context of finite-difference, finite-volume and finite-element methods (e.g., [40–42]). Since the ALE methods strongly rely on a mesh, they belong to the family of meshdependent methods. At the same time, MPM combines a fixed Eulerian background mesh and a set of Lagrangian point masses, called *material points* (or particles). Although it is not derived directly from a classical *meshfree* or *meshless* method, MPM is described as a meshfree approach and is actively discussed within the meshfree community due to its similarities, in terms of advantages and challenges, with typical meshfree methods.

2.1.2. MESHFREE METHODS

This section provides a brief introduction to meshfree methods. For a detailed overview the reader is referred to, for example, the work of Li and Liu [43], Belytschko et al.[44], and Chen et al. [45]. The conventional meshbased methods enable efficient and reliable solutions to a large number of computational problems, but for some processes of practical engineering interest, such as the simulation of land- and flowslides or metal forming processes, their underlying structure is not well suited. In general, meshbased methods do not cope adequately with problems that involve large deformations, moving discontinuities, fracture, irregular domains, etc. The use of meshdependent methods for such problems is expensive and frequently results in slow convergence, deterioration of accuracy, or even breakdown of the computation. For such cases, meshfree methods offer an acceptable alternative to meshbased schemes.

The development of meshfree methods started about 40 years ago with the smoothed particle hydrodynamics (SPH) method [46, 47]. SPH is a particle method that uses an integral representation of variables, or kernel approximation [48, 49], to solve the governing equations. It was originally introduced for modeling astrophysical phenomena, but was later extended to problems in solid mechanics, which could not be solved easily with meshbased techniques. Nowadays, there exist a wide variety of meshfree schemes: element free Galerkin [50–52], generalized finite-element [53, 54], and optimal transportation mehsfree [32] methods are just a few examples. The estimation in meshfree methods can be based on different concepts, such as moving least squares, partition of unity, kernels, and maximum entropy [44, 45]. The common feature of meshfree methods is that they approximate unknowns based on scattered points without mesh connectivity [45].

2.1.3. PARTICLE-IN-CELL AND FLUID-IMPLICIT PARTICLE METHODS

MPM is an adaptation of the Fluid-Implicit Particle (FLIP) [55, 56] method to solid mechanics. FLIP belongs to the large family of *particle* methods and descends from the Particle-In-Cell (PIC) [57] method, which can be tracked back to the middle of the 1950s

2

[58]. PIC was originally developed for fluid-dynamics problems by the group of Harlow at Los Alamos National Laboratory in the USA. The success of PIC arose from the fact that it offered an alternative to meshbased techniques by simply and effectively combining Lagrangian and Eulerian approaches. In PIC, the Lagrangian particles are used to advect all transported quantities of the material, while an Eulerian computational grid is used to solve the governing equations. However, the early PIC schemes suffered from significant dissipation caused by frequent interpolation of the grid values to the particles [55]. This issue is addressed by the Fluid-Implicit Particle method. In contrast to PIC, FLIP transfers only the increments of the grid values to the particles. Various techniques based on PIC and FLIP are applied for computer-graphics calculations, fluid dynamics, and plasma modeling (e.g., [59–63]).

2.2. PHYSICAL MODEL

The application of the principles from continuum mechanics requires respecting the conservation laws. For the problems studied with MPM, it suffices to consider the conservation laws related to thermomechanical systems. These laws are the conservation of mass, the conservation of linear and angular momentum, and the conservation of energy. Their comprehensive derivation can be found, for instance, in the work of Malvern [36]. Although the conservation laws should be valid for all continua, they have to be extended by constitutive (or material) models to describe the mechanical behavior of a particular material. Constitutive relations form a voluminous research subject, but are not a focus of this thesis. For a detailed information on constitutive modeling the reader is referred to, for example, Spencer [64]. This section provides only a short overview of the conservation equations and gives the description of the constitutive models that are relevant to this thesis. Furthermore, after Sulsky et al. derived an MPM formulation for single-phase problems, several multi-phase formulations (e.g., [6, 65, 66]) have been proposed to further expand its applicability range. A summary of these formulations is given in Section 2.6.1.

We start by introducing some notation from continuum mechanics. It is assumed that the considered one-phase continuum occupies the domain $\Omega^0 \subseteq \mathbb{R}^3$ at the initial time t^0 and domain $\Omega^t \subseteq \mathbb{R}^3$ at any later time $t > t^0$. The initial position of the material is denoted by $\mathbf{x}^0 = [x_1^0, x_2^0, x_3^0]^\mathsf{T}$, while the position at time t is $\mathbf{x} = [x_1, x_2, x_3]^\mathsf{T}$. In the abstract formulation, the deformation mapping is defined as $\varphi : \mathbb{R}^3 \times [t^0, T] \to \mathbb{R}^3$, where T is the final time. For a fixed time t, the mapping $\Omega^0 \mapsto \Omega^t$ can be considered as the 'push forward' operator, which needs to be bijective, so that the current domain can also be 'pulled back' to the initial one via $\varphi^{-1}(\mathbf{x}, t) : \Omega^t \to \Omega^0$. Likewise, it links the initial and current positions:

$$\mathbf{x} = \varphi(\mathbf{x}^0, t), \quad \mathbf{x}^0 = \varphi^{-1}(\mathbf{x}, t). \tag{2.1}$$

The displacement, velocity, and acceleration vectors are, denoted by

$$\mathbf{u} = [u_1, u_2, u_3]^\mathsf{T}, \quad \mathbf{v} = [v_1, v_2, v_3]^\mathsf{T}, \quad \mathbf{a} = [a_1, a_2, a_3]^\mathsf{T}, \tag{2.2}$$

respectively. The displacement at time t is defined as the difference between the current and initial positions:

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{x} - \mathbf{x}^0. \tag{2.3}$$
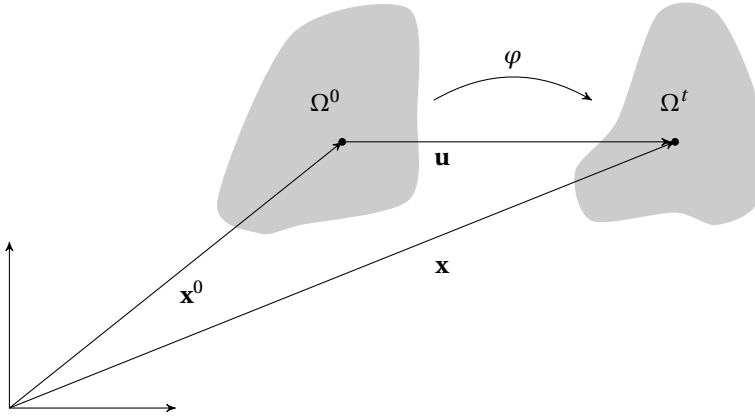
Figure 2.1: Schematic representation of the deformation of a continuum.

This is illustrated in Figure 2.1. The velocity and acceleration are defined by means of the material time derivative. Using $\nabla = \left[\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_3}\right]^{\mathsf{T}}$, the material time derivative can be written as

$$\frac{\mathrm{d}}{\mathrm{d}\, t} = \frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla. \tag{2.4}$$

Since the convective effects can be neglected in Lagrangian computations (e.g., Donea et al.[38]), the velocity and acceleration are obtained from

$$\mathbf{v}(\mathbf{x}, t) \quad = \quad \frac{\mathrm{d}}{\mathrm{d}\, t}\mathbf{u}(\mathbf{x}, t) = \frac{\partial}{\partial t}\mathbf{u}(\mathbf{x}, t), \tag{2.5}$$

$$\mathbf{a}(\mathbf{x}, t) \quad = \quad \frac{\mathrm{d}}{\mathrm{d}\, t}\mathbf{v}(\mathbf{x}, t) = \frac{\partial}{\partial t}\mathbf{v}(\mathbf{x}, t). \tag{2.6}$$

For the description of the constitutive relation, some models rely on the rate of deformation tensor, while other models are based on the deformation gradient tensor. The deformation gradient tensor $\mathbf{F}$ is defined as

$$\mathbf{F} = \mathbf{I} + \frac{\partial \mathbf{u}}{\partial \mathbf{x}^0} = \mathbf{I} + \left\{\frac{\partial u_k}{\partial x_l^0}\right\}, \tag{2.7}$$

where $\mathbf{I}$ is the identity matrix. The rate of deformation tensor $\mathbf{D}$ is the symmetric part of the velocity gradient:

$$\mathbf{D} = \frac{1}{2}\left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}} + \left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right)^{\mathsf{T}}\right). \tag{2.8}$$

In this thesis, the deformation gradient tensor is used to present the constitutive laws, whereas the rate of deformation tensor is required to describe the conservation of energy.

2.2.1. GOVERNING EQUATIONS

CONSERVATION OF MASS

The conservation of mass states that when mass is not entering or leaving the domain Ω^t (i.e., the domain contains no source or sink terms), the change of mass with time should be equal to zero. In the updated Lagrangian form adopted within MPM, this can be expressed as follows:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \tag{2.9}$$

where ρ is the mass density of the material.

CONSERVATION OF MOMENTUM

The conservation of linear momentum represents the equation of motion of a continuum body. In other words, it is equivalent to Newton's second law of motion. Denoting the Cauchy stress tensor by $\boldsymbol{\sigma}$ and a body force by $\mathbf{b}$, the conservation of linear momentum can be written as

$$\frac{\partial (\rho \mathbf{v})}{\partial t} - \nabla \cdot \boldsymbol{\sigma} - \rho \mathbf{b} = 0 \tag{2.10}$$

The conservation of angular momentum implies that the stress tensor is symmetric:

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^\mathsf{T}. \tag{2.11}$$

Since the conservation of angular momentum does not add a new equation, the conservation of linear momentum is usually referred to as the momentum equation.

CONSERVATION OF ENERGY

Assuming that mechanical work is the only source of energy, the conservation of energy can be written as

$$\rho \frac{\partial r}{\partial t} = \mathrm{tr}\,(\mathbf{D}\boldsymbol{\sigma})\,. \tag{2.12}$$

Here, r is the internal energy per unit mass.

CONSTITUTIVE MODELS

In this thesis, only *linear elastic* and *neo-Hookean* (or *hyperelastic*) constitutive relations are considered. A multi-dimensional linear elastic model can be described as

$$\boldsymbol{\sigma} = \lambda \mathbf{I}\, \mathrm{tr}\left(\frac{1}{2}\left(\mathbf{F} + \mathbf{F}^\mathsf{T}\right) - \mathbf{I}\right) + 2\mu\left(\frac{1}{2}\left(\mathbf{F} + \mathbf{F}^\mathsf{T}\right) - \mathbf{I}\right), \tag{2.13}$$

where λ and μ are Lamé's first parameter and the shear modulus, respectively. Denoting the determinant of $\mathbf{F}$ by J, the neo-Hookean material model is given by

$$\boldsymbol{\sigma} = \frac{\lambda \ln(J)}{J}\mathbf{I} + \frac{\mu}{J}\left(\mathbf{F}\mathbf{F}^\mathsf{T} - \mathbf{I}\right). \tag{2.14}$$

For the mathematical research that is presented in this thesis, it is sufficient to consider simple elastic constitutive equations. However, for many applications of MPM, these models are typically replaced by (visco-)elasto-plastic constitutive equations in order to

properly represent the behavior of complex materials. For instance, an elasto-plastic Mohr-Coulomb model is used to simulate the failure of a dam [6], predict the run-outs of landslides in brittle soils [67], and test the bearing capacity of a deep foundation in cohesive soil [68].

2

2.3. DISCRETIZATION

For the discretization, MPM simplifies Equation (2.10) by assuming that the density is constant in time and solves the simplified equation on the fixed background grid by adopting a variational formulation. To obtain the weak form of the momentum equation, it is first multiplied with a so-called test function and integrated over the domain Ω^t. Letting ω denote an element of the test space that consists of all functions, which are sufficiently smooth and zero on the part of the boundary where essential boundary conditions are prescribed, the following expression is obtained:

$$\int_{\Omega^t} w\rho \frac{\partial \mathbf{v}}{\partial t} \, d\Omega^t = \int_{\Omega^t} w(\nabla \cdot \boldsymbol{\sigma}) \, d\Omega^t + \int_{\Omega^t} w\rho \mathbf{b} \, d\Omega^t. \tag{2.15}$$

Applying integration by parts and Gauss' theorem yields the weak form of the momentum equation:

$$\int_{\Omega^t} w\rho \frac{\partial \mathbf{v}}{\partial t} \, d\Omega^t = \int_{\partial \Omega^t} w\mathbf{n} \cdot \boldsymbol{\sigma} \, d\Gamma^t - \int_{\Omega^t} \nabla w \cdot \boldsymbol{\sigma} \, d\Omega^t + \int_{\Omega^t} w\rho \mathbf{b} \, d\Omega^t. \tag{2.16}$$

The first integral on the right-hand side is over the boundary $\Gamma^t = \partial \Omega^t$ of the domain Ω^t and $\mathbf{n}$ is the outward unit normal vector.

2.3.1. SPATIAL DISCRETIZATION

Following the standard approach in FEM, Equation (2.16) is discretized in space by approximating the acceleration by a linear combination of basis functions ϕ_j:

$$\mathbf{a}(\mathbf{x}, t) \approx \mathbf{a}_h(\mathbf{x}, t) = \sum_{j=1}^{\mathcal{N}} \mathbf{a}_j(t)\phi_j(\mathbf{x}) \tag{2.17}$$

with subscript h indicating a spatially discretized form and $\mathcal{N}$ being the total number of nodes on the background grid. By substituting Equation (2.17) into Equation (2.16), interchanging the order of summation and integration, and choosing the test function w equal to ϕ_i, the following expression is obtained for all $i = 1, \ldots, \mathcal{N}$:

$$\sum_{j=1}^{\mathcal{N}} \left(\int_{\Omega^t} \phi_i \rho \phi_j \, d\Omega^t \right) \mathbf{a}_j = \int_{\partial \Omega^t} \phi_i \mathbf{n} \cdot \boldsymbol{\sigma} \, d\Gamma^t - \int_{\Omega^t} \nabla \phi_i \cdot \boldsymbol{\sigma} \, d\Omega^t + \int_{\Omega^t} \phi_i \rho \mathbf{b} \, d\Omega^t. \tag{2.18}$$

For each direction x_k, Equation (2.18) can be written in the matrix-vector form:

$$\mathbf{M}\bar{\mathbf{a}}_k = \bar{\mathbf{f}}_k, \tag{2.19}$$

where $\bar{\mathbf{a}}_k = \left[a_{k,1}, a_{k,2}, \ldots, a_{k,\mathcal{N}} \right]^{\mathsf{T}}$ is the unidirectional vector of the unknown coefficients[1], $\mathbf{M} = \left[M_{ij} \right] \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$ is the consistent mass matrix, and $\bar{\mathbf{f}}_k = \left[f_{k,1}, f_{k,2}, \ldots, f_{k,\mathcal{N}} \right]^{\mathsf{T}}$ is the unidirectional force vector. The entries of the mass matrix and force vector are given by

$$M_{ij} \quad = \quad \int_{\Omega^t} \phi_i \rho \phi_j \, d\Omega^t, \tag{2.20}$$

$$f_{k,i} \quad = \quad \int_{\partial \Omega^t} \phi_i \tau_k \, d\Gamma^t - \int_{\Omega^t} \sum_{l=1}^{3} \frac{\partial \phi_i}{\partial x_l} \sigma_{lk} \, d\Omega^t + \int_{\Omega^t} \phi_i \rho b_k \, d\Omega^t, \tag{2.21}$$

respectively. Here, $\boldsymbol{\tau}(\mathbf{x}, t)$ is the prescribed traction at the boundary. In general, MPM can be implemented combining Equation (2.19) for each k into one linear system. The implementation procedure of a multi-dimensional MPM can be found, for instance, in the thesis of Kafaji [37].

In MPM computations, the consistent mass matrix is typically replaced by the row-sum *lumped* mass matrix $\mathbf{M}^L$, which can be obtained by summing the off-diagonal entries of $\mathbf{M}$ in each row, adding them to the diagonal entry, and subsequently setting the off-diagonal entries to zero:

$$M_{ii}^L = \sum_{j=1}^{\mathcal{N}} M_{ij}. \tag{2.22}$$

If the basis functions maintain a partition of unity within the domain, that is

$$\sum_{j=1}^{\mathcal{N}} \phi_j(\mathbf{x}) = 1 \qquad \forall \, \mathbf{x} \in \Omega^t, \tag{2.23}$$

the diagonal entry of the lumped mass matrix can be expressed as follows:

$$M_{ii}^L = \sum_{j=1}^{\mathcal{N}} M_{ij} = \sum_{j=1}^{\mathcal{N}} \int_{\Omega^t} \phi_i \rho \phi_j \, d\Omega^t = \int_{\Omega^t} \phi_i \rho \sum_{j=1}^{\mathcal{N}} \phi_j \, d\Omega^t = \int_{\Omega^t} \phi_i \rho \, d\Omega^t. \tag{2.24}$$

This implies that the diagonal entries can be computed directly from the final integral of Equation (2.24), which significantly increases the efficiency of the simulation by avoiding the generation of the off-diagonal terms.

In MPM, material points represent the material and carry all physical information about it (e.g., the mass, strain, and stress). While most material-point properties vary in time, the mass is time independent. This assures that the conservation of mass, Equation (2.9), is satisfied. Furthermore, throughout a simulation, the integrals in Equations (2.20) and (2.21) are approximated by projecting the material-point information onto the background grid. Let the continuum be discretized by $\mathcal{M}$ material points, then the integral of an arbitrary vector-valued function $g(\mathbf{x})$ is approximated by

$$\int_{\Omega} g(\mathbf{x}) d\Omega = \sum_{p=1}^{\mathcal{M}} V_p g(\mathbf{x}_p), \tag{2.25}$$

---

[1]We remark that the unknown coefficients correspond to the classical "nodes" located at the vertices of the cells for linear Lagrangian finite element basis functions but can also be of modal type for B-spline basis functions presented in Chapter 3. To simplify the presentation for the reader who is more familiar with nodal basis functions, we will stick to the terminology of "nodes" unless stated otherwise.

where V_p and $\mathbf{x}_p$ represent the material-point volume and position, respectively. The information obtained by solving Equation (2.19) is then mapped back to update the material points.

2.3.2. TEMPORAL DISCRETIZATION

In the considered version of MPM, the Euler-Cromer time-stepping scheme [69] is used for temporal discretization. The scheme applies the forward Euler method to advance the velocity, but employs the backward Euler method for the displacement:

$$\mathbf{v}_i^{s+1} = \mathbf{v}_i^s + \left(t^{s+1} - t^s \right) \mathbf{a}_i^s, \tag{2.26}$$

$$\mathbf{u}_i^{s+1} = \mathbf{u}_i^s + \left(t^{s+1} - t^s \right) \mathbf{v}_i^{s+1}, \tag{2.27}$$

where $s = 0, 1, \ldots, N-1$ is the time-step counter. The Euler-Cromer scheme is energy conservative for oscillatory problems (see [70] for more details). Its combination with finite-element discretization results in a conditionally stable solution algorithm [71]. Moreover, assuming a constant time-step size, the Euler-Cromer scheme is second-order accurate in the displacement and first-order accurate in the velocity. To the author's knowledge this has not been mentioned previously in the MPM literature. Therefore, a short explanation is provided below.

First, note that it follows from Equation (2.27) that

$$\mathbf{v}_i^s = \frac{1}{\Delta t} \left(\mathbf{u}_i^s - \mathbf{u}_i^{s-1} \right), \tag{2.28}$$

where Δt denotes the constants time-step size. Therefore, the displacement at time step $s + 1$ can be rewritten as

$$\begin{aligned}
\mathbf{u}_i^{s+1} &= \mathbf{u}_i^s + \Delta t \left(\mathbf{v}_i^s + \Delta t \, \mathbf{a}_i^s \right) \\
&= \mathbf{u}_i^s + \Delta t \left(\frac{1}{\Delta t} \left(\mathbf{u}_i^s - \mathbf{u}_i^{s-1} \right) + \Delta t \, \mathbf{a}_i^s \right) \\
&= 2\mathbf{u}_i^s - \mathbf{u}_i^{s-1} + \Delta t^2 \mathbf{a}_i^s.
\end{aligned} \tag{2.29}$$

The above expression is obtained by first substituting Equation (2.26) into Equation (2.27), and then using Equation (2.28). Equation (2.29) corresponds to the central difference scheme for the second derivative of the displacement:

$$\mathbf{a}_i^s = \frac{1}{\Delta t^2} \left(\mathbf{u}_i^{s+1} - 2\mathbf{u}_i^s + \mathbf{u}_i^{s-1} \right), \tag{2.30}$$

which is known to be second-order accurate.

At the beginning of each time step, the material-point information is projected from the particles to the nodes of the background grid, where the discretized governing equations are assembled and subsequently solved. The obtained information is then mapped back to update the material points. A schematic representation of the MPM solution strategy that is followed within each time step is provided in Figure 2.2.

2
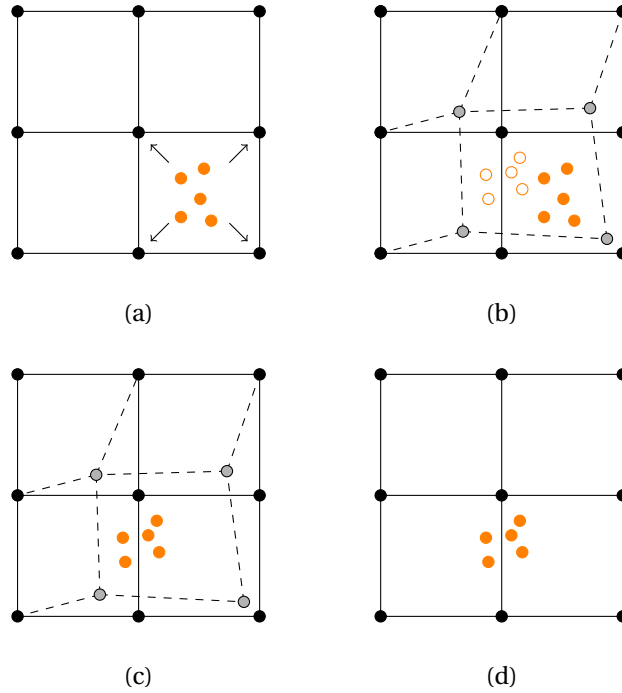
Figure 2.2: Schematic representation of the MPM solution strategy within each time step: (a) projection of the material-point information onto the background grid; (b) solution of the governing equations at the grid nodes; (c) update of the material-point information; (d) update of the material-point positions with respect to the grid.

2.3.3. REMARKS ON CONSERVATION PROPERTIES

From the above derivation of the scheme, it follows that MPM conserves the mass and momentum by design, but does not enforce the conservation of energy. Much research has been conducted with regard to this issue (e.g., Bardenhagen [72], Nairn [11], Love and Sulsky [27, 73]). It has been shown that the conservation of energy strongly depends on the version of the MPM algorithm [72]. This is further discussed in Section 2.6.3. Moreover, a consistent mass matrix is required to conserve energy while mapping the information from the material points to the background grid and vice versa [27]. However, the employment of a consistent mass matrix can significantly reduce the stability of an MPM simulation and is computationally more expensive.

2.4. ALGORITHMS

This section presents two versions of the MPM algorithm: the original version [1], which is also known as update stress last (USL), and a modified one [2], modified update stress last (MUSL) or, alternatively, update velocity first (UVF). The algorithms adopt the notation used previously for the unidirectional acceleration and force vectors, $\bar{\mathbf{a}}_k$ and $\bar{\mathbf{f}}_k$,

for global nodal vectors (e.g., the nodal velocity vector in the direction x_k is denoted by $\bar{\mathbf{v}}_k = [v_{k,1}, v_{k,2}, \ldots, v_{k,\mathcal{N}}]^\top$). Furthermore, it is assumed that only a constant body force is acting externally. It should be noted that Algorithms 1 and 2 contain consistent mass matrices. However, as was already mentioned in Section 2.3.1, MPM typically uses the lumped mass matrix.

The names USL and MUSL refer to the fact that the material-point stress is updated at the last stage within a time step. The main difference between the USL and MUSL algorithms is that USL computes the nodal velocities directly from the nodal accelerations, while MUSL obtains them from the material-point velocities. Nairn [11] explains in detail that assuming that both algorithms adopt a lumped mass matrix, the modified version can avoid extremely small nodal masses that lead to ill-conditioned mass matrices in the original version of the algorithm. This greatly improves the stability of the scheme. Furthermore, Bardenhagen [72] explains that MUSL slowly dissipates energy, but still recommends its use, because it damps out numerical artifacts and the damping is consistent with the accuracy of the solution. In addition to USL and MUSL, there exist the update stress first algorithm [72] and update stress averaged scheme [11] that are described in Section 2.6.3.

Input: Nodal coordinates $\mathbf{x}_i^0$, material-point coordinates $\mathbf{x}_p^0$, velocities $\mathbf{v}_p^0$, volumes V_p^0, densities ρ_p^0, masses m_p, deformation gradients $\mathbf{F}_p^0$, stresses $\boldsymbol{\sigma}_p^0$, body forces $\mathbf{b}_p$

1 Set $s = 0$

2 while $s < N$ **do**

3 $\quad$ Compute basis functions $\phi_i^0\left(\mathbf{x}_p^s\right)$ and gradients $\nabla\phi_i^0\left(\mathbf{x}_p^s\right)$ from initial nodal set $\left\{\mathbf{x}_i^0\right\}$ and advected material-point set $\left\{\mathbf{x}_p^s\right\}$

4 $\quad$ Compute mass matrix $\mathbf{M}^s$, linear momentum vector $\bar{\mathbf{q}}_k^s$, and force vector $\bar{\mathbf{f}}_k^s$:

$$M_{ij}^s = \sum_{p=1}^{\mathcal{M}} \phi_i^0\left(\mathbf{x}_p^s\right) m_p \phi_j^0\left(\mathbf{x}_p^s\right) \tag{2.31}$$

$$q_{k,i}^s = \sum_{p=1}^{\mathcal{M}} \phi_i^0\left(\mathbf{x}_p^s\right) m_p v_{k,p}^s \tag{2.32}$$

$$f_{k,i}^s = \sum_{p=1}^{\mathcal{M}} \left(\sum_{l=1}^{3} \frac{\partial \phi_i^0}{\partial x_l}\left(\mathbf{x}_p^s\right) \sigma_{lk,p}^s + \phi_i^0\left(\mathbf{x}_p^s\right) \rho_p^s b_{k,p}\right) V_p^s \tag{2.33}$$

5 $\quad$ Compute nodal accelerations:

$$\bar{\mathbf{a}}_k^s = \left(\mathbf{M}^s\right)^{-1} \bar{\mathbf{f}}_k^s \tag{2.34}$$

6 $\quad$ Compute nodal velocities and incremental nodal displacements:

2

$$\bar{\mathbf{v}}_k^s = \left(\mathbf{M}^s\right)^{-1} \bar{\mathbf{q}}_k^s \tag{2.35}$$

$$\bar{\mathbf{v}}_k^{s+1} = \bar{\mathbf{v}}_k^s + \left(t^{s+1} - t^s\right) \bar{\mathbf{a}}_k^s \tag{2.36}$$

$$\Delta\bar{\mathbf{x}}_k^{s+1} = \left(t^{s+1} - t^s\right) \bar{\mathbf{v}}_k^{s+1} \tag{2.37}$$

7   Update material-point coordinates:

$$\mathbf{x}_p^{s+1} = \mathbf{x}_p^s + \sum_{i=1}^{\mathcal{N}} \phi_i^0\left(\mathbf{x}_p^s\right) \Delta\mathbf{x}_i^{s+1} \tag{2.38}$$

8   Update material-point velocities:

$$\mathbf{v}_p^{s+1} = \mathbf{v}_p^s + \left(t^{s+1} - t^s\right) \sum_{i=1}^{\mathcal{N}} \phi_i^0\left(\mathbf{x}_p^s\right) \mathbf{a}_i^s \tag{2.39}$$

9   Update material-point deformation gradients:

$$\nabla\mathbf{v}_p^{s+1} = \sum_{i=1}^{\mathcal{N}} \nabla\phi_i^0\left(\mathbf{x}_p^s\right) \mathbf{v}_i^{s+1} \tag{2.40}$$

$$\mathbf{F}_p^{s+1} = \left(\mathbf{I} + \left(t^{s+1} - t^s\right) \nabla\mathbf{v}_p^{s+1}\right) \mathbf{F}_p^s \tag{2.41}$$

10   Update material-point volumes:

$$V_p^{s+1} = \det\left(\mathbf{F}_p^{s+1}\right) V_p^0 \tag{2.42}$$

11   Update material-point densities:

$$\rho_p^{s+1} = \frac{m_p}{V_p^{s+1}} \tag{2.43}$$

12   Compute material-point stresses $\boldsymbol{\sigma}_p^{s+1}$ from $\mathbf{F}_p^{s+1}$ using the constitutive equation (e.g., Equation 2.13 or 2.14)

13   Set $s = s + 1$

end

    Algorithm 1: Update stress last (USL) version of material-point method.

_____

_____

Input: Nodal coordinates $\mathbf{x}_i^0$, material-point coordinates $\mathbf{x}_p^0$, velocities $\mathbf{v}_p^0$,
 volumes V_p^0, densities ρ_p^0, masses m_p, deformation gradients $\mathbf{F}_p^0$, stresses
 $\boldsymbol{\sigma}_p^0$, body forces $\mathbf{b}_p$

2

1 Set $s = 0$

2 **while** $s < N$ **do**

3   Compute basis functions $\phi_i^0\left(\mathbf{x}_p^s\right)$ and gradients $\nabla\phi_i^0\left(\mathbf{x}_p^s\right)$ from initial nodal set
      $\left\{\mathbf{x}_i^0\right\}$ and advected material-point set $\left\{\mathbf{x}_p^s\right\}$

4   Compute mass matrix $\mathbf{M}^s$ and force vector $\bar{\mathbf{f}}_k^s$:

$$M_{ij}^s = \sum_{p=1}^{\mathscr{M}} \phi_i^0\left(\mathbf{x}_p^s\right) m_p \phi_j^0\left(\mathbf{x}_p^s\right) \tag{2.44}$$

$$f_{k,i}^s = \sum_{p=1}^{\mathscr{M}} \left(\sum_{l=1}^{3} \frac{\partial \phi_i^0}{\partial x_l}\left(\mathbf{x}_p^s\right) \sigma_{lk,p}^s + \phi_i^0\left(\mathbf{x}_p^s\right) \rho_p^s b_{k,p} \right) V_p^s \tag{2.45}$$

5   Compute nodal accelerations:

$$\bar{\mathbf{a}}_k^s = \left(\mathbf{M}^s\right)^{-1} \bar{\mathbf{f}}_k^s \tag{2.46}$$

6   Update material-point velocities:

$$\mathbf{v}_p^{s+1} = \mathbf{v}_p^s + \left(t^{s+1} - t^s\right) \sum_{i=1}^{\mathscr{N}} \phi_i^0\left(\mathbf{x}_p^s\right) \mathbf{a}_i^s \tag{2.47}$$

7   Compute linear momentum vector $\bar{\mathbf{q}}_k^s$:

$$q_{k,i}^s = \sum_{p=1}^{\mathscr{M}} \phi_i^0\left(\mathbf{x}_p^s\right) m_p v_{k,p}^{s+1} \tag{2.48}$$

8   Update nodal velocities:

$$\bar{\mathbf{v}}_k^{s+1} = \left(\mathbf{M}^s\right)^{-1} \bar{\mathbf{q}}_k^s \tag{2.49}$$

9   Compute incremental nodal displacements:

$$\Delta\bar{\mathbf{x}}_k^{s+1} = \left(t^{s+1} - t^s\right) \bar{\mathbf{v}}_k^{s+1} \tag{2.50}$$

10   Update material-point coordinates:

$$\mathbf{x}_p^{s+1} = \mathbf{x}_p^s + \sum_{i=1}^{\mathscr{N}} \phi_i^0\left(\mathbf{x}_p^s\right) \Delta\mathbf{x}_i^{s+1} \tag{2.51}$$

11   Update material-point deformation gradients:

$$\nabla\mathbf{v}_p^{s+1} = \sum_{i=1}^{\mathscr{N}} \nabla\phi_i^0\left(\mathbf{x}_p^s\right) \mathbf{v}_i^{s+1} \tag{2.52}$$

$$\mathbf{F}_p^{s+1} = \left(\mathbf{I} + \left(t^{s+1} - t^s\right) \nabla \mathbf{v}_p^{s+1}\right) \mathbf{F}_p^s \tag{2.53}$$

12   Update material-point volumes:

$$V_p^{s+1} = \det\left(\mathbf{F}_p^{s+1}\right) V_p^0 \tag{2.54}$$

13   Update material-point densities:

$$\rho_p^{s+1} = \frac{m_p}{V_p^{s+1}} \tag{2.55}$$

14   Compute material-point stresses $\boldsymbol{\sigma}_p^{s+1}$ from $\mathbf{F}_p^{s+1}$ using the constitutive
       equation (e.g., Equation 2.13 or 2.14)
15   Set $s = s + 1$
   end
Algorithm 2: Modified update stress last (MUSL) version of material-point method.

2.5. BASIS FUNCTIONS

This section completes the description of standard MPM by presenting the piecewise-linear Lagrange (P1) basis functions. Since the multivariate P1 basis functions can be built from the univariate ones by taking the tensor product, this section presents only the univariate P1 basis functions. To simplify the notation x_1 is replaced by x for one-dimensional descriptions throughout this dissertation.

The univariate P1 basis functions are defined using a set of nodes $\{x_i\}_{i=1}^{\mathcal{N}}$:

$$\phi_i = \begin{cases} 0 & \text{if } x < x_{i-1}, \\ (x - x_{i-1}) / (x_i - x_{i-1}) & \text{if } x_{i-1} \leq x < x_i, \\ 1 - (x - x_i) / (x_{i+1} - x_i) & \text{if } x_i \leq x < x_{i+1}, \\ 0 & \text{if } x \geq x_{i+1}. \end{cases} \tag{2.56}$$

The corresponding derivatives are equal to

$$\frac{d\phi_i}{dx} = \begin{cases} 0 & \text{if } x < x_{i-1}, \\ 1/(x_i - x_{i-1}) & \text{if } x_{i-1} \leq x < x_i, \\ -1/(x_{i+1} - x_i) & \text{if } x_i \leq x < x_{i+1}, \\ 0 & \text{if } x \geq x_{i+1}. \end{cases} \tag{2.57}$$

The univariate P1 basis functions and their derivatives are illustrated in Figure 2.3.

On the one hand, P1 basis functions have several advantages:

P. 1. They satisfy the *partition of unity* property:

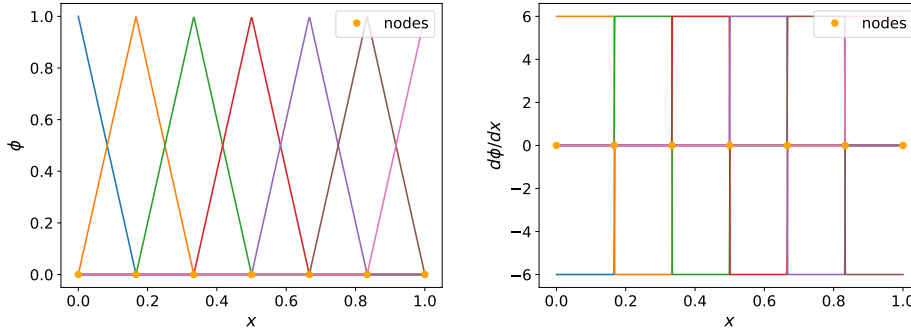$$\sum_{i=1}^{\mathcal{N}} \phi_i(x) = 1 \quad \forall x \in \Omega.$$

Figure 2.3: Univariate piecewise-linear Lagrange basis functions and their derivatives.

P. 2. Each ϕ_i has *compact support*. For $i = 2, 3, \ldots, \mathcal{N} - 1$, ϕ_i is supported by element $i - 1$ and i. The compact supports of basis functions ϕ_1 and $\phi_{\mathcal{N}}$ are contained in element 1 and $\mathcal{N} - 1$, respectively.

P. 3. Each ϕ_i is non-negative over its entire support.

P. 4. They allow for row-sum mass lumping due to *P. 3*. (see Equation 2.22).

P. 5. They allow for direct mass lumping due to *P. 1*. (see Equation 2.24).

Furthermore, P1 basis functions can be implemented in a straightforward manner. On the other hand, the gradients of the piecewise-linear basis functions, just as the gradients of all other C^0-continuous basis functions, are discontinuous on the element boundaries, which can lead to unphysical oscillations in the internal forces when material points cross those boundaries.

2.6. DEVELOPMENT

2.6.1. FORMULATIONS

After Sulsky et al. derived an MPM formulation for single-phase problems, several multi-phase formulations have been proposed to further expand the applicability range of MPM. We distinguish between two categories of multi-phase formulations. The first category consists of the so-called *one-point* formulations that assign all considered phases to the same material point. For example, the behavior of saturated soil is modeled by a two-phase one-point formulation where each material point represents both solid and liquid phases. Zhang et al. [74] suggested a two-phase one-point formulation with the velocity of fluid and hydraulic pressure as primary variables to model dynamic responses of saturated soil subjected to contact or impact of solid bodies. Jassim et al. [75] developed a coupled dynamic one-point formulation with the velocities of solid and liquid phases as primary variables and applied it to simulate the pore pressure development under wave attack on a sea dike. Zabala and Alonso [6] proposed a two-phase form of governing equations whereby the solid acceleration and the pressure increment were

computed at the background grid, while the velocity of the liquid phase was obtained at the material points. Their formulation was applied to model the progressive failure of the Aznalcóllar dam. More recently, Yerro et al. [76] introduced a three-phase one-point formulation by including a gas phase in a one-point formulation and applied it to study unsaturated soils.

The second category of multi-phase MPM formulations uses a different set of material points for each considered phase. So far, this strategy has only been applied to two-phase problems. A so-called *double-point* formulation allows for relative motion of the two phases, while conserving their masses [77]. Zhang et al. [78] used a double-point formulation to simulate failure phenomena in multi-phase porous media. Shin [79] and Mackenzie-Helnwein et al. [80] proposed a two-phase double-point formulations to model solid-fluid mixtures. Abe et al. [65] developed a coupled double-point formulation based on the Biot's mixture theory. His approach was further improved by Bandara and Soga [66] by including the relative acceleration of fluid with respect to the solid skeleton. Despite a high number of existing MPM formulations, their development remains a principal topic in the MPM community. One of the reasons for further research is that many of the formulations are only designed for small deformations [66]. In addition, double-point formulations can be highly computationally expensive and require great care when interface regions that separate saturated soil from dry soil or free water are modeled [77].

2.6.2. TIME INTEGRATION

Typically, MPM advances the solution in time using explicit integration schemes (e.g., [5, 18, 81–84]). Explicit time integration is well-suited for simulations involving impacts at high velocities and fast transient problems. Although originally MPM was introduced with the Euler-Cromer scheme, computations with the centered-difference, Runge-Kutta, Adams-Bashforth-Moulton, and Predictor-Corrector Newmark methods have been performed as well [18, 84]. After studying different integration schemes, Wallstedt and Guilkey [84] concluded that the choice of the scheme has an impact on the overall accuracy of a simulation, but the spatial errors remain dominant in the explicit version of MPM.

Several studies have adopted MPM with implicit time integration (e.g., [85–89]). The implicit formulation can be applied to problems with, for instance, quasi-static loading, or small rate of deformation. Sulsky and Kaul [86] derived an implicit version of MPM based on the combination of Newton's method with the conjugate gradient (CG) method and generalized minimum residual (GMRES) method. Guilkey and Weiss [85] proposed an implicit approach that combined Newton's method and Newmark integration scheme. It has been shown that the results of implicit MPM are in close agreement with the results obtained using implicit FEM [85]. In addition, the implicit version of MPM has been demonstrated to be significantly more accurate than explicit MPM [85, 86].

2.6.3. ALGORITHMS

As was mentioned in Section 2.4, next to the USL and MUSL algorithm, there exist the update stress first (USF) algorithm and update stress averaged (USAVG) scheme. Both USF and USAVG attempt to improve the energy conservation within MUSL. USF and

MUSL are closely related: the only algorithmic difference between them is that USF updates the stress at the beginning of a time step, based on the material-point velocity at time level s, whereas MUSL does that at the end of the time step using the velocity at time level $s+1$. In contrast to MUSL that slowly dissipates energy, USF tends to slowly increase the energy [11]. For this reason, USAVG is designed as an average of the USF and MUSL schemes [11]. Bardenhagen [72] points out that USF and USL algorithms provide different results only for unresolved modes, that is when the oscillation-mode wavelength of the considered problem is comparable to the element size. The advantage of MUSL is that it damps out those unresolved modes, and its damping is consistent with the accuracy of the solution [72].

2.6.4. RELATED METHODS

Methods described here are used to reduce or completely eliminate the so-called *grid(-cell)-crossing instabilities* [24]. Grid-crossing instabilities arise when material points travel from one element to another and are due to the use of piecewise-linear basis functions, whose gradients are discontinuous on element boundaries. These instabilities generate unphysical oscillations in the forces and lead to inaccurate results mainly in terms of the stress, but can also significantly influence velocity and displacement.

GENERALIZED INTERPOLATION MATERIAL-POINT METHOD

Bardenhagen and Kober [24] introduced a generalization of MPM, the generalized interpolation material-point (GIMP) method, where particles are represented by *particle-characteristic* functions. GIMP assumes that the information mapping from material-points to the grid in MPM takes the following form:

$$g_i = \sum_{p=1}^{\mathcal{M}} g\left(\mathbf{x}_p\right)\phi_i\left(\mathbf{x}_p\right). \tag{2.58}$$

This can be rewritten as

$$g_i = \sum_{p=1}^{\mathcal{M}} g\left(\mathbf{x}_p\right)\frac{\int_{\Omega^t}\phi_i(\mathbf{x})\delta\left(\mathbf{x}-\mathbf{x}_p\right)\,d\Omega^t}{\int_{\Omega^t}\delta\left(\mathbf{x}-\mathbf{x}_p\right)\,d\Omega^t}, \tag{2.59}$$

where $\delta\left(\mathbf{x}-\mathbf{x}_p\right)$ is the Dirac delta distribution centered at the particle position $\mathbf{x}_p$. GIMP then replaces δ with a general particle-characteristic function $\chi_p\left(\mathbf{x}-\mathbf{x}_p\right)$ that is centered at $\mathbf{x}_p$. In other words, Equation (2.58) is changed to

$$g_i = \sum_{p=1}^{\mathcal{M}} g\left(\mathbf{x}_p\right)\breve{\phi}_i\left(\mathbf{x}_p\right), \tag{2.60}$$

where $\breve{\phi}_i$ is a weighting function, given by

$$\breve{\phi}_i\left(\mathbf{x}_p\right) = \frac{\int_{\Omega^t}\phi_i(\mathbf{x})\chi_p\left(\mathbf{x}-\mathbf{x}_p\right)\,d\Omega^t}{\int_{\Omega^t}\chi_p\left(\mathbf{x}-\mathbf{x}_p\right)\,d\Omega^t}. \tag{2.61}$$

Moreover, the gradient of the basis function is modified as follows:

$$\widetilde{\nabla \phi}_i \left(\mathbf{x}_p \right) = \frac{\int_{\Omega^t} \nabla \phi_i(\mathbf{x}) \chi_p \left(\mathbf{x} - \mathbf{x}_p \right) \, d\Omega^t}{\int_{\Omega^t} \chi_p \left(\mathbf{x} - \mathbf{x}_p \right) \, d\Omega^t}. \tag{2.62}$$

One-dimensional particle-characteristic functions are frequently chosen to be piecewise-constant. The piecewise-constant particle-characteristic functions can be defined as a combination of step functions $\mathscr{H}$:

$$\chi_p = \mathscr{H} \left(x - \left(x_p - \frac{l_p}{2} \right) \right) - \mathscr{H} \left(x - \left(x_p + \frac{l_p}{2} \right) \right). \tag{2.63}$$

Here, l_p is the measure of the current particle shape. The particle shape, which is also frequently called particle or material-point domain, describes the spatial extent occupied by the particle. In one dimension, the measure of the particle shape can be defined as the time-dependent material-point volume V_p.

Similarly to the piecewise-linear basis functions, multivariate GIMP basis functions can be constructed as the tensor product of the univariate GIMP basis functions. In the multivariate case, the main question is how to define the measure of the material-point shape after the shape starts to deform. One option is to use the initial shape regardless of the time level. Sadeghirad et al. [29] refer to this version as the uniform GIMP (uGIMP) method. Another option is to obtain the particle shape in each direction from the diagonal components of the material-point deformation gradient tensor. Sadeghirad et al. [29] denote this method as the contiguous-particle GIMP (cpGIMP) method. The cpGIMP method ensures that the particle shapes remain rectangular or cuboid throughout the simulation. Finally, the GIMP method can also be combined with a structured mesh-refinement technique in order to adjust the particle shape [90]. To update the material-point shape this refinement technique places massless tracking particles at the corner of the initial shapes and advects them with the grid velocity.

The GIMP methods have been applied to many engineering problems such as modeling of landslides [4], the densification of real foam microstructures [91], dynamic crack growth [92], and seepage failure [93]. However, it should be noted that the standard GIMP and uGIMP methods are only suited for problems that involve small deformations [24, 84]. For relatively accurate simulations of large deformations, the use of the more advanced cpGIMP method and the GIMP method with the structured mesh-refinement technique is essential [24, 29, 84]. The main disadvantages of the latter versions is that they neglect shear distortion and employ tracking techniques to update material-point shapes [29, 30, 84].

CONVECTED PARTICLE-DOMAIN INTERPOLATION METHOD

The convected particle-domain interpolation (CPDI) [29] method modifies the GIMP method in order to capture material-point shapes after shearing. Assuming that particle shapes are parallelograms and the deformation gradient is constant over each shape, the shape deformation can be expressed as

$$\mathbf{l}_{p,1}^{s+1} = \mathbf{F}_p^{s+1} \mathbf{l}_{p,1}^0, \tag{2.64}$$

$$\mathbf{l}_{p,2}^{s+1} = \mathbf{F}_p^{s+1} \mathbf{l}_{p,2}^0, \tag{2.65}$$

where $(\mathbf{l}_{p,1}^0, \mathbf{l}_{p,2}^0)$ and $(\mathbf{l}_{p,1}^{s+1}, \mathbf{l}_{p,2}^{s+1})$ are the vectors defining material-point shapes at the initial and updated configurations. In addition, the CPDI method proposes to use alternative basis functions within Equations (2.61) and (2.62). The CPDI basis functions are constructed by interpolating the standard piecewise-linear basis functions at the four corners of each particle domain. The P1 and CPDI basis functions are different in the interior of the material-point shape, but are the same at the particle corners.

Compared to the GIMP family of methods, the CPDI method has higher accuracy and better stability properties [29]. Nevertheless, the CPDI2 [94] method has been proposed to further increase the accuracy of the method. The material-point shapes in the CPDI2 method are described as quadrilaterals in two dimensions and hexahedra in three dimensions. Recently, the CPDI method has also been extended to arbitrary grids [95, 96] significantly increasing its range of applications. It has been reported that the method has convergence issues and produces pressure spikes within shock wave simulations when the number of material points per element is relatively high [21]. At the same time, the method has been successfully used for complex problems including modeling of ceramics [96], pile driving [97], and fully saturated porous media [98].

B-SPLINE MATERIAL-POINT METHOD

The B-spline material-point method (BSMPM) [28, 31] replaces the piecewise-constant Lagrange basis functions by higher-order B-spline basis functions (see Chapter 3 for a detailed description of B-spline basis functions). Two approaches have been proposed to construct the B-spline basis functions.

First of all, it is possible to define the B-spline basis functions separately for each node. That implies that to construct the univariate quadratic B-spline basis function for the grid node i with $1 < i < \mathcal{N} - 1$, the *knot vector* (i.e., a sequence of ordered nondecreasing points in $\mathbb{R}$ called *knots*) $\{x_{i-3/2}, x_{i-1/2}, x_{i+1/2}, x_{i+3/2}\}$ is adapted, where $x_{i+1/2}$ is defined as $\frac{1}{2}(x_i + x_{i+1})$. For example, an internal zero-centered quadratic B-spline basis function is given by

$$\phi(x) = \begin{cases} \frac{1}{2h^2}x^2 + \frac{3}{2h}x + \frac{9}{8} & \text{if } -\frac{3}{2}h \leq x \leq -\frac{1}{2}h, \\ -\frac{1}{h^2}x^2 + \frac{3}{4} & \text{if } -\frac{1}{2}h \leq x \leq \frac{1}{2}h, \\ \frac{1}{2h^2}x^2 - \frac{3}{2h}x + \frac{9}{8} & \text{if } \frac{1}{2}h \leq x \leq \frac{3}{2}h, \\ 0 & \text{if } \text{otherwise,} \end{cases} \tag{2.66}$$

where h is the grid spacing. For $i = 1$, the knot vector becomes $\{x_0, x_{i-1/2}, x_{i+1/2}, x_{i+3/2}\}$, whereas for $i = 0$, it is equal to the sum of two B-splines defined by the knot vectors $\{x_i, x_i, x_{i+1}, x_{i+3/2}\}$ and $\{x_i, x_i, x_i, x_{i+1/2}\}$. A similar strategy is applied to $i = \mathcal{N} - 1$ and $i = \mathcal{N}$. The tensor product of univariate B-spline basis functions is used to obtain the multivariate B-spline basis functions.

A different approach is to construct all B-spline basis functions from a single knot vector that consists of the knots from the whole domain. This technique is described in detail in Chapter 3. Regardless of the construction approach, the B-spline basis functions satisfy the partition of unity property, have a compact support, are non-negative, and have at least C^0-continuous gradients. However, only the first construction technique guarantees that for a uniform grid, the peaks of the basis functions are centered

over the grid nodes. On the other hand, B-spline basis functions created from a single knot vector are easier to implement. The use of higher-order B-spline basis functions has been shown to not only lead to lower errors, but also to significantly improve the convergence behavior of MPM [31]. In addition, it outperforms the GIMP and CPDI methods for plate impact problems [99].

DUAL DOMAIN MATERIAL-POINT METHOD

The dual domain material-point method (DDMPM) [30] method uses piecewise-linear basis functions, but replaces their gradients by smoother ones. For the construction of the new gradients, the method introduces a weight function α and the gradient $\widetilde{\nabla \phi_i}$ defined as

$$\widetilde{\nabla \phi_i}(\mathbf{x}) = \sum_{j=1}^{\mathcal{N}} \frac{\phi_j(\mathbf{x})}{V_j} \int_\Omega \phi_j(\mathbf{x}) \nabla \phi_i(\mathbf{x}) d\Omega,$$

where V_j is the volume associated with node j and given by $V_j = \int_\Omega \phi_j(\mathbf{x}) d\Omega$. The weight function is required to be zero on the cell boundaries, but is not uniquely specified [30]. In Chapter 5, where DDMPM is applied to one-dimensional problems, the following expression for the weight function is adopted:

$$\alpha(x) = 2 \sum_{i=1}^{\mathcal{N}} \phi_i(x) \phi_{i+1}(x).$$

DDMPM substitutes the gradients of the piecewise-linear basis functions by

$$\overline{\nabla \phi}_i(\mathbf{x}) = \alpha(\mathbf{x}) \nabla \phi_i(\mathbf{x}) + (1 - \alpha(\mathbf{x})) \widetilde{\nabla \phi_i}(\mathbf{x}) \quad \forall i = \{1, \dots, \mathcal{N}\},$$

An example of the DDMPM gradients is given in Figure 2.4. From Section 2.4, it follows that DDMPM modifies the computation of the forces and velocity gradients.
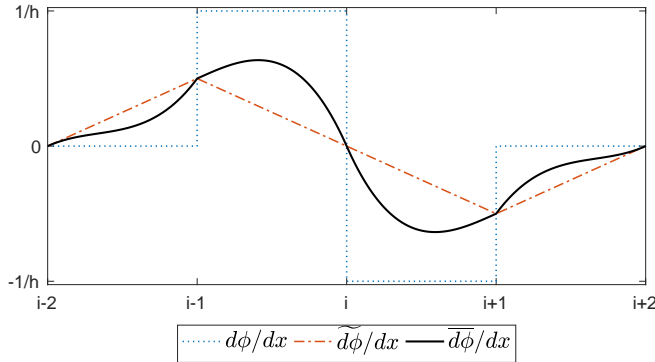

Figure 2.4: The derivatives of univariate basis functions that are used in DDMPM. Here, h represents the element length.

A version of DDMPM with an alternative strategy for numerical integration has been proposed by Dhakal and Zhang [21]. This version allows for a lower number of material points per cell and, hence, lower computational costs compared to the original

DDMPM method. Although from the mathematical point of view, the combination of P1 basis functions with smoother gradients can be considered as less fundamental than the GIMP, CPDI, and BSMPM methods, DDMPM has been shown to be effective within different applications (e.g., shock wave propagation [21] and ductile void growth [100]). In addition, it has been pointed out that, for shock wave simulations, DDMPM is a viable alternative to MPM, the GIMP method, and the CPDI method [21].

2.6.5. OTHER IMPROVEMENTS

Although Sections 2.6.1 to 2.6.4 attempt to provide a summary of fundamental research categories in the MPM community, there exist a significant number of highly valuable research studies that either do not entirely fit the categories of the aforementioned sections or combine multiple research areas. This section describes a small fraction of such contributions.

Much research has been performed to reduce the grid-crossing instabilities in multi-phase simulations. For example, Liu et al. [93] re-derived the two-phase double-point formulation of Bandara and Soga [66] for GIMP, while Zheng et al. [98] combined the two-phase formulation introduced by Zhang et al. [74] with CPDI. At the same time, the double-point formulation derived by Bandara and Soga was used in conjunction with an implicit time scheme [101]. Furthermore, an important contribution to mutli-phase formulations was made by Zhang et al. [102], where they proposed an improved approach for multi-phase calculations that ensured that the continuity condition (i.e., the volume fractions of all considered materials sum to one) was satisfied.

From the mathematical point of view, significant development of MPM was made by Gritton and Berzins [103], who proposed a null-space filter based on a singular-value-decomposition technique. A null-space filter can be required to ensure that non-zero material-point values do not result in zero nodal values after being mapped to the grid. Moreover, Cortis et al. [104] worked on the application of essential boundary conditions in MPM, focusing on the cases when the boundary of the problem domain does not co-incide with the grid-element edges. On the other hand, Bing et al. [105] investigated B-spline based Dirichlet and Neumann boundary conditions. Furthermore, Sulsky and Gong [25, 26] looked at function reconstruction within MPM and improved the accuracy of the method by adopting higher-order reconstruction techniques. The volumetric locking issue in MPM was addressed by Love and Sulsky [73] as well as Coombs et al. [106], while the choice of computational mesh and spatial discretization order are discussed by Andersen and Andersen [107].

While some studies unite different research directions within the MPM community (e.g., [93, 98, 101, 108]), a number of studies extend the use of MPM and related methods by combining them with other numerical tools for solving partial differential equations. For instance, Lian et al. [109, 110] and Chen et al. [111] couple MPM with FEM. Alternatively, MPM can be combined with other meshfree methods or even discrete modeling techniques. Raymond et al. [112] and He et al. [113] couple MPM with SPH, whereas Lu et al. [114] use MPM in conjunction with molecular dynamics. Moreover, Dhakal et al. propose the joint application of molecular dynamics with DDMPM [115]. Table 2.1 provides an overview of the research studies discussed in Section 2.6. Although a numerical comparison of existing methods would be highly valuable for their further evaluation, it

is not a focus of this thesis.

Table 2.1: Overview research areas in MPM.

Research area	Subdivision	Examples of existing research
Formulation	one-point one-phase	[1, 2]
	one-point two-phase	[6, 74, 75, 98]
	one-point three-phase	[76]
	double-point	[65, 66, 78–80, 93]
Time integration	explicit	[5, 18, 81–84]
	implicit	[85–89, 101, 108]
Algorithm	USL	[1]
	MUSL	[2, 11, 72]
	USF	[11, 72]
	USAVG	[11]
Related method	GIMP	[4, 24, 84, 90–93]
	uGIMP	
	cpGIMP	
	CPDI	[29, 94–96, 98]
	CPDI2	
	BSMPM	[28, 31, 99, 116]
	DDMPM	[21, 30, 100, 115]
Other	–	[25, 26, 93, 98, 101–115]

II

METHODOLOGICAL IMPROVEMENTS OF THE MATERIAL-POINT METHOD

3

FURTHER DEVELOPMENT OF THE B-SPLINE MATERIAL-POINT METHOD

This chapter considers the version of B-spline material-point method (BSMPM), where the basis functions are constructed from a single knot vector. It introduces a mapping technique that allows for the projection of the material-point information to the background grid using cubic-spline interpolation and Gauss quadrature rules. In addition, a possible manner to extend BSMPM to unstructured grids is presented. The chapter is structured as follows. Section 3.1 discusses the construction and properties of B-spline basis functions. Section 3.2 introduces the proposed mapping strategy. It presents the general concept of the technique and its individual components. Section 3.3 gives a short description of MPM with Powell-Sabin splines. Section 3.4 introduces the benchmark problems and provides the obtained results. Finally, Section 3.5 draws the conclusions.

3.1. B-SPLINE BASIS FUNCTIONS

A B-spline basis function is a piecewise-polynomial function of a certain degree on its support interval that spans a particular spline space. Figure 3.1 provides examples of linear, quadratic, and cubic univariate B-spline basis functions and their derivatives. Clearly, linear B-spline basis functions are identical to the P1 basis functions described in Section 2.5. A linear combination of B-spline basis functions defines a B-spline curve. B-splines form building blocks for non-uniform rational B-splines (NURBS) [119]. Both B-splines and NURBS are commonly employed to model and discretize geometries in computer-aided design (CAD) and solve partial differential equation in isogeometric analysis, a generalization of the standard, polynomial-based, FEM introduced by Hughes et al. [119], [120–123].

3.1.1. CONSTRUCTION

Generally, a B-spline basis function is defined in the parametric space, based on a knot vector. As was mentioned in Section 2.6.4, a knot vector in one dimension is a sequence of ordered nondecreasing coordinates. It is typically denoted as $\Xi = \left\{ \xi_1, \xi_2, \ldots, \xi_{n_{\bar{b}} + \bar{d} + 1} \right\}$, where $\xi_j \in \mathbb{R}$ is the jth knot, $n_{\bar{b}}$ is the total number of basis functions, and $\bar{d}$ is the polynomial order. The knot vector is *uniform* when its knots are distributed equidistantly. The knots are called *repeated* when more than one knot is positioned at the same coordinate in the parametric space. An *open* knot vector contains the first and last knots $\bar{d} + 1$ times ensuring that the resulting basis functions are interpolatory at the boundaries of the domain. A non-empty knot interval $[\xi_j \ \xi_{j+1})$ is referred to as a *knot span*. For an open uniform knot vector, the number of spans is equal to $n_{\bar{b}} - \bar{d}$.

The Cox-de Boor formula [124] defines B-spline basis functions recursively, starting with piecewise constants (no repeated knots, i.e., $\bar{d} = 0$):

$$\phi_j^0(\xi) = \begin{cases} 1 & \text{if } \xi_j \leq \xi < \xi_{j+1}, \\ 0 & \text{otherwise.} \end{cases}$$

For $\bar{d} > 0$, the basis functions are given by

$$\phi_j^{\bar{d}}(\xi) = \frac{\xi - \xi_j}{\xi_{j+\bar{d}} - \xi_j} \phi_j^{\bar{d}-1}(\xi) + \frac{\xi_{j+\bar{d}+1} - \xi}{\xi_{j+\bar{d}+1} - \xi_{j+1}} \phi_{j+1}^{\bar{d}-1}(\xi) \qquad \xi \in \hat{z},$$

where $\hat{z}$ is the parametric domain. The derivatives of the B-spline basis functions can be computed from [124]:

$$\frac{d\phi_j^{\bar{d}}(\xi)}{d\xi} = \frac{\bar{d}}{\xi_{j+\bar{d}} - \xi_j} \phi_j^{\bar{d}-1}(\xi) - \frac{\bar{d}}{\xi_{j+\bar{d}+1} - \xi_{j+1}} \phi_{j+1}^{\bar{d}-1}(\xi).$$

It should be noted that in the considered implementation of BSMPM, the parametric and physical domains are the same.

3.1.2. PROPERTIES

B-spline basis functions satisfy the following properties:
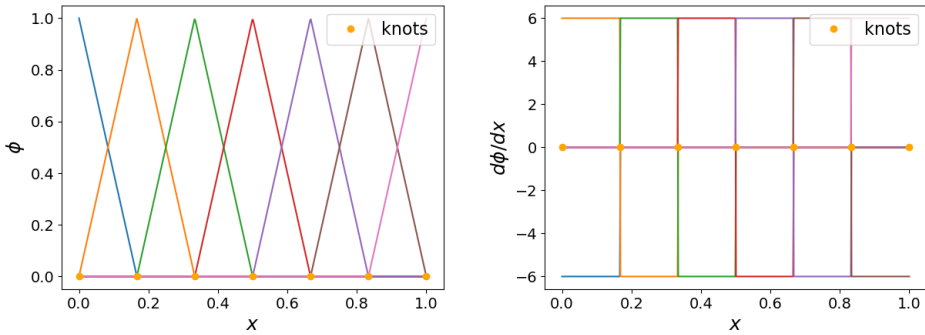
(a) Linear $\left(\bar{d} = 1\right)$

(b) Quadratic $\left(\bar{d} = 2\right)$

(c) Cubic $\left(\bar{d} = 3\right)$

Figure 3.1: Univariate B-spline basis functions and their derivatives.

P. 1. They form a partition of unity:

$$\sum_{j=1}^{n_{\bar{b}}} \phi_j^{\bar{d}}(\xi) = 1 \quad \forall\, \xi \in \hat{z}.$$

P. 2. They have a compact support. The support of each $\phi_j^{\bar{d}}$ is contained within the interval $\left[\xi_j, \xi_{j+\bar{d}+1}\right]$.
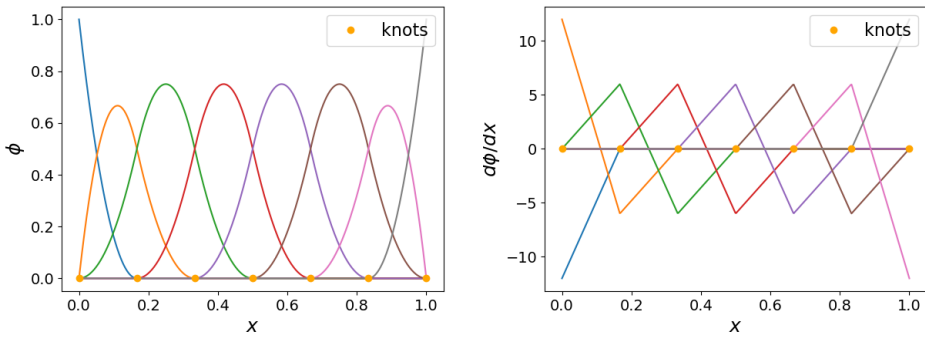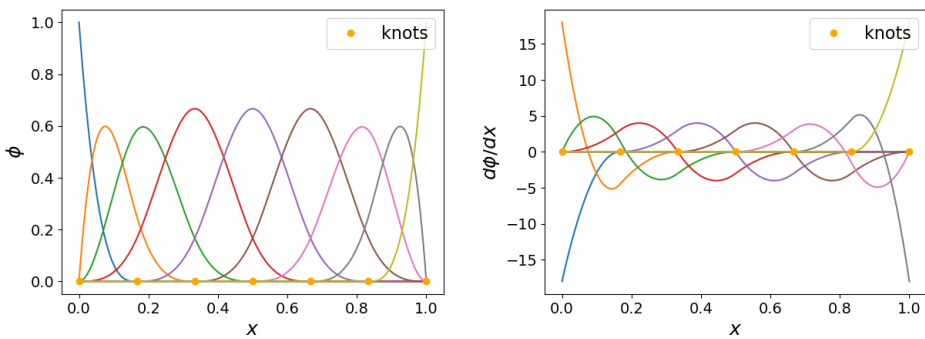
P. 3. Each $\phi_j^{\bar{d}}$ is non-negative in its support:

$$\phi_j^{\bar{d}}(\xi) \geq 0 \quad \forall\, x \in \left[\xi_j, \xi_{j+\bar{d}+1}\right].$$

P. 4. They allow for row-sum mass lumping due to *P. 3.* (see Equation 2.22).

P. 5. They allow for direct mass lumping due to *P. 1.* (see Equation 2.24).

P. 6. They attain $C^{\bar{d}-\kappa_j}$ continuity, where κ_j denotes the multiplicity of the jth knot (i.e., the number of times knot j is duplicated). When the multiplicity of a knot is equal to $\bar{d}$, the basis function is interpolatory at that knot.

Properties *P. 1.*-*P. 5.* are similar to the properties of piecewise-linear basis functions. The main difference lies in property *P. 6.* This property makes B-spline basis functions particularly well suited for MPM simulations, because the smoothness of the basis functions is required to prevent the unphysical oscillations resulting from grid crossing.

3.1.3. APPLICATION TO MPM
This thesis considers only second-order B-splines. These basis functions require a relatively small interval for their support and are sufficiently smooth (i.e., C^1-continuous). In this section and Chapter 5, the B-spline basis functions are used within the MUSL algorithm, described in Algorithm 2.

3.2. MAPPING OF MATERIAL-POINT DATA TO THE BACKGROUND GRID
MPM can be viewed as a version of standard FEM, where the material points serve as the integration points to provide information for the background grid. In this thesis, we refer to this type of particle data mapping as *direct*. Sulsky and Gong [25, 26] point out that the direct mapping procedure reconstructs functions using a particular case of Shepard interpolation [125] that can introduce significant errors. The authors improve the accuracy of the method by adopting higher-order reconstruction techniques. In the modified versions, the velocity, density, and stress fields are reconstructed from the particle data and evaluated either at the nodes or element centres. The MPM-integration is then replaced by a one-point quadrature rule.

Based on the mapping strategy presented by Sulsky and Gong, this section provides an alternative approach to project the material-point information to the background

grid. The approach is presented for one-dimensional problems. The extension to multi-dimensional computations is possible, but is outside the scope of this dissertation. Figure 3.2 shows schematic representations of the direct and modified mappings. The general methodology of the proposed cubic spline (CS) mapping can be described as follows:

- cubic-spline interpolation globally reconstructs certain quantities of interest from scattered material-point information,

- the obtained approximation is evaluated at the integration points (e.g., Gauss points),

- the resulting values are projected to the background grid by means of exact numerical integration (e.g., Gauss quadrature).
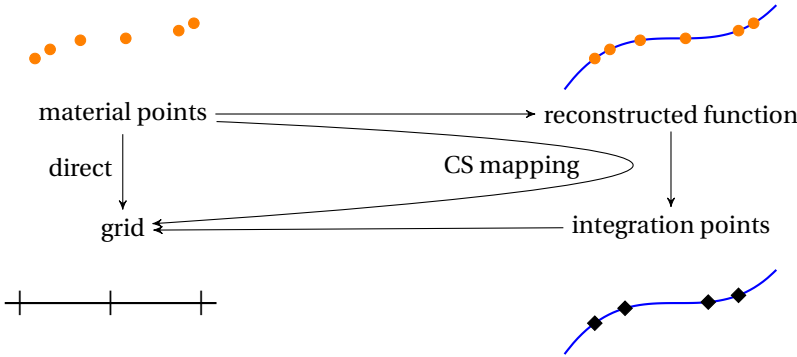
Figure 3.2: Schematic representation of direct data mapping and mapping using cubic-spline (CS) interpolation.

3.2.1. FUNCTION RECONSTRUCTION

Given a limited amount of data describing a real-valued function f (e.g., a set of $\mathcal{M}$ distinct one-dimensional data points, $\{x_p\}_{p=1}^{\mathcal{M}}$, and the function values at these points, $\{f(x_p)\}_{p=1}^{\mathcal{M}}$), function reconstruction techniques determine intermediate values of the function or values outside of the provided data range. In general, a function can be reconstructed by interpolating or approximating the known values. An interpolating technique assures that the values of the reconstructed function z coincide with the known values of f:

$$z(x_p) = f(x_p) \quad \forall p = \{1, \ldots, \mathcal{M}\}. \tag{3.1}$$

Approximation techniques do not necessarily respect this property. Commonly used function reconstruction techniques are linear or higher-order Lagrange interpolation, Hermite interpolation, spline interpolation, and (weighted or moving) least-squares approximation [126, 127].

Cubic-spline interpolation

A *spline* is a piecewise-polynomial curve that is differentiable up to a prescribed order. Assuming that $x_1 < x_2 < \cdots < x_{\mathcal{M}}$, a spline of degree d is defined as

$$z(x) = \begin{cases} z_1(x) & \text{for } x \in [x_1, x_2], \\ z_2(x) & \text{for } x \in [x_2, x_3], \\ \vdots \\ z_{\mathcal{M}-1}(x) & \text{for } x \in [x_{\mathcal{M}-1}, x_{\mathcal{M}}], \end{cases} \tag{3.2}$$

where $z_p(x)$ is a polynomial of degree $\bar{d}$ on $[x_p, x_{p+1}]$, such that $z_p(x)$ and its derivatives are connected smoothly at x_p and x_{p+1} to the neighbouring polynomials. This implies that a cubic spline interpolation has the following properties:

- On each interval $[x_p, x_{p+1}]$, z is a third degree polynomial z_p with $p = \{1, \ldots, \mathcal{M}-1\}$,

- $z(x_p) = f(x_p)$ for $p = \{1, \ldots, \mathcal{M}\}$,

- $z_p^{(q)}(x_{p+1}) = z_{p+1}^{(q)}(x_{p+1})$ for $q = \{0, 1, 2\}$ and $p = \{1, \ldots, \mathcal{M}-2\}$,

- $z_1^{(2)}(x_1) = z_{\mathcal{M}-1}^{(2)}(x_{\mathcal{M}}) = 0$.

It should be noted that the last property is not fundamental: it is required to uniquely define the spline but can be replaced with other conditions that allow for the uniqueness of the unknowns.

In MPM, the material points serve as data points. When large deformations are involved, the material-point distribution can become non-uniform, containing regions sparsely occupied by the material points. For this reason, Figure 3.3 illustrates the cubic-spline reconstruction of $f(x) = \sin(x) + 2$ on $[0, 4\pi]$ with different types of data distribution. Since no data point is located at 4π, the reconstruction involves extrapolation at the right boundary. It is observed that the quality of the reconstruction is high, when the data points are distributed uniformly, but decreases if the distances between the data points become relatively large.
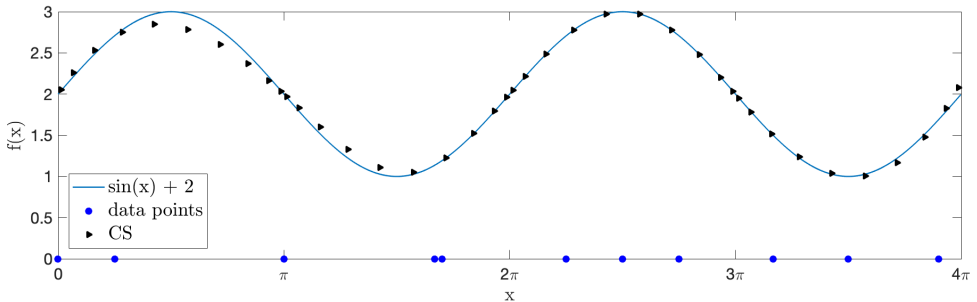

Figure 3.3: Cubic-spline (CS) reconstruction of $f(x) = \sin(x) + 2$ on $[0, 4\pi]$. Data distribution differs within each interval $[\bar{n}\pi, (\bar{n}+1)\pi]$ with $\bar{n} = \{0, 1, 2, 3\}$.

3.2.2. NUMERICAL INTEGRATION

Since the reconstructed function is a third degree polynomial, it allows for an analytical computation of its integral within each element. However, numerical integration can be significantly cheaper and similarly accurate. Essentially, numerical integration approximates a definite integral to a given degree of accuracy employing a finite summation:

$$\int_a^b z(x)\,dx \approx \sum_{g=1}^{n_g} \omega_g z\left(x_g\right). \tag{3.3}$$

Here, a and b (with $a < b$) are the boundaries of the integration domain, n_g is the total number of integration points, ω_g is the weight of an integration point, and x_g is its position. Although there exist many types of numerical integration rules, such as the midpoint and Newton-Cotes formulas, we restrict our attention to Gaussian quadrature.

GAUSS QUADRATURE RULES

Typically numerical integration techniques choose the positions of the integration points first and determine their weights afterwards. Gaussian quadrature determines the positions and weights simultaneously. Gauss quadrature rules are based on the idea that the use of n_g Gauss points should yield the exact integration for polynomials up to degree $2n_g + 1$. In the case of a single integration point, its weight and position are computed by assuming that the Gaussian quadrature is exact for polynomials up to order one:

$$\int_a^b dx = b - a = \omega_g, \tag{3.4}$$

$$\int_a^b x\,dx = \frac{1}{2}\left(b^2 - a^2\right) = \omega_g x_g. \tag{3.5}$$

From this, it follows that $\omega_g = b - a$ and $x_g = \frac{1}{2}(a + b)$, so that the Gauss point is located exactly in the middle of the integral. Conventionally, the general integration domain of integration $[a, b]$ is transformed into $[-1, 1]$. On $[-1, 1]$, the one-point Gauss rule results in $\omega_g = 2$ and $x_g = 0$, whereas two-point Gauss rule prescribes the weight of 1 to both points and locates them at $\pm\sqrt{\frac{1}{3}}$.

3.2.3. APPLICATION TO B-SPLINE MATERIAL-POINT METHOD

The mapping technique proposed in this section can be applied to standard MPM as well as to its related methods, such as BSMPM and DDMPM. This is demonstrated in Chapter 5. Here, a detailed description of the application is provided only for BSMPM. To adopt the technique for other methods, elements should be considered instead of knot spans.

The CS mapping replaces the MPM-integration by Gauss quadrature for the computation of internal forces from Equation (2.45):

$$f_{\text{int},i} = \sum_{p=1}^{\mathcal{M}} \frac{\partial \phi_i^0}{\partial x}\left(x_p\right) \sigma_p V_p \tag{3.6}$$

as well as the nodal velocity (Equation 2.49). First, the proposed technique uses cubic-spline interpolation to reconstruct the stress, density, and momentum fields from the

scattered material-point data. This is done for the complete domain and the reconstructed fields are denoted by $\hat{\sigma}$, $\hat{\rho}$, and $\widehat{\rho v}$ for the stress, density, and momentum, respectively. The obtained fields are subsequently evaluated at the Gauss points. For an accurate approximation of the integrals, four Gauss points are placed within each knot span: two within each half of a knot span. An example of the Gauss-point positions is given in Figure 3.4. After that, the internal force, mass and total momentum are calculated with the two-point Gauss rule in each half of each knot span:

$$f_{\text{int},i} \approx \sum_{k_1=1}^{\mathcal{N}_s} \sum_{k_2=1}^{2} \sum_{g=1}^{2} \frac{\partial \phi_i^0}{\partial x}\left(x_g\right) \hat{\sigma}\left(x_g\right) \omega_g = \sum_{g=1}^{\mathcal{N}_g} \frac{\partial \phi_i^0}{\partial x}\left(x_g\right) \hat{\sigma}\left(x_g\right) \omega_g, \qquad (3.7)$$

$$q_i \approx \sum_{k_1=1}^{\mathcal{N}_s} \sum_{k_2=1}^{2} \sum_{g=1}^{2} \phi_i^0\left(x_g\right) \left(\widehat{\rho v}\right)\left(x_g\right) \omega_g = \sum_{g=1}^{\mathcal{N}_g} \phi_i^0\left(x_g\right) \left(\widehat{\rho v}\right)\left(x_g\right) \omega_g, \qquad (3.8)$$

$$M_{ij} \approx \sum_{k_1=1}^{\mathcal{N}_s} \sum_{k_2=1}^{2} \sum_{g=1}^{2} \phi_i^0\left(x_g\right) \hat{\rho}\left(x_g\right) \phi_j^0\left(x_g\right) \omega_g = \sum_{g=1}^{\mathcal{N}_g} \phi_i^0\left(x_g\right) \hat{\rho}\left(x_g\right) \phi_j^0\left(x_g\right) \omega_g \qquad (3.9)$$

where $\mathcal{N}_s$ is the total number of knot spans, k_2 refers to the number of intervals within each knot span, and $\mathcal{N}_g$ is the total number of Gauss points. From this, the nodal velocity vector can be obtained as

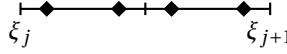$$\mathbf{v} = (\mathbf{M})^{-1}\mathbf{q}. \qquad (3.10)$$

Figure 3.4: Illustration of Gauss points within a knot span.

3.3. EXTENSION TO UNSTRUCTURED TRIANGULAR GRIDS USING POWELL-SABIN SPLINES

On structured rectangular grids, adopting B-spline basis functions within MPM not only eliminates grid-crossing errors but also yields higher-order spatial convergence [28, 31, 99, 128]. Several research studies also demonstrate that BSMPM is a viable alternative to the GIMP, CPDI methods and DDMPM, which have been introduced in Chapter 2 [31, 99, 129, 130]. While the CPDI method and DDMPM can be used on unstructured grids [30, 95, 96], to the best of our knowledge, BSMPM for unstructured grids does not yet exist. This implies that its applicability to real-world problems is limited compared to the CPDI method and DDMPM.

The use of quadratic Powell-Sabin (PS) splines [131] extends BSMPM to unstructured triangulations to combine the benefits of B-splines with the geometric flexibility of triangular grids. The method employs quadratic PS splines and is referred to as PS-MPM. These splines are piecewise higher-order polynomials defined on a particular refinement of any given triangulation and are typically used in computer-aided geometric design and approximation theory [132–135]. PS splines are C^1-continuous and hence overcome the grid-crossing issue within MPM by design. We remark that although this

paper focuses on PS splines, other options such as refinable C^1 splines [136] can most likely be used to extend MPM to unstructured triangular grids. To construct PS-splines on an arbitrary triangulation, a Powell-Sabin refinement is required (i.e., dividing each triangle into sub-triangles). Dierckx et al. [133, 137] provide the details of the construction process.

3.4. NUMERICAL RESULTS

This section provides the numerical results obtained with BSMPM, its combination with the mapping technique that uses cubic-spline basis functions, also referred to as the CS mapping in this thesis, and Powell-Sabin splines on unstructured triangular grids.

3.4.1. BENCHMARKS

Two benchmarks are considered in this section. A one-dimensional vibrating bar is adopted to illustrate the results obtained with BSMPM and its combination with the CS mapping, whereas the performance of Powell-Sabin splines is studied based on a two-dimensional vibrating plate.

BAR WITH FIXED ENDS

This example describes the vibration of a one-phase bar with fixed ends. The motion triggered by an initial velocity that varies along the bar as illustrated in Figure 3.5.
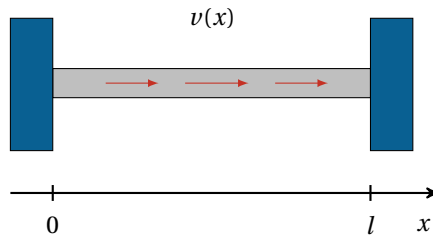

Figure 3.5: Vibrating bar with fixed ends.

Moreover, the gravitational force is neglected. The following initial and boundary conditions are prescribed:

$$u(x^0, 0) = 0, \quad v(x^0, 0) = v^0_{max} \sin\left(\frac{\pi x^0}{l}\right), \quad \sigma(x^0, 0) = 0;$$
$$u(0, t) = 0, \quad u(l, t) = 0.$$

For small strains, the analytical solution in terms of displacement, velocity, and stress

is given by

$$u(x^0, t) = \frac{v^0_{max} l}{\pi \sqrt{E/\rho}} \sin\left(\frac{\pi \sqrt{E/\rho}\, t}{l}\right) \sin\left(\frac{\pi x^0}{l}\right),$$

$$v(x^0, t) = v^0_{max} \cos\left(\frac{\pi \sqrt{E/\rho}\, t}{l}\right) \sin\left(\frac{\pi x^0}{l}\right),$$

$$\sigma(x^0, t) = v^0_{max} \sqrt{E/\rho} \sin\left(\frac{\pi \sqrt{E/\rho}\, t}{l}\right) \cos\left(\frac{\pi x^0}{l}\right).$$

The material-point solutions are considered at the particle positions.

Table 3.1 provides two sets of exemplary parameter values for the vibrating bar benchmark under small deformations. For the simulations performed with the parameters from set 1, the domain is divided into 510 elements for MPM and 510 knot spans for BSMPM with 12 particles per cell (PPC). The domain is discretized by 40 knot spans with 12 PPC, when the parameters from set 2 are adopted.

Table 3.1: Exemplary parameters allowing for small deformations in the vibrating bar problem.

Parameter	Symbol	Value set 1	Value set 2	Unit
Height	l	1.00	1.00	m
Initial density	ρ	$2.00 \cdot 10^3$	$2.00 \cdot 10^3$	kg/m^3
Young's modulus	E	$8.00 \cdot 10^7$	$7.00 \cdot 10^6$	Pa
Max. initial velocity	v^0_{max}	0.60	0.28	m/s^2
Time-step size	Δt	$1.0 \cdot 10^{-5}$	$1.0 \cdot 10^{-7}$	s
Total time	T	$1.0 \cdot 10^{-3}$	$1.9 \cdot 10^{-6}$	s

PLATE UNDERGOING AXIS-ALIGNED DISPLACEMENT

This benchmark [29] considers a two-dimensional neo-Hookean plate that is fixed at the entire boundary. The plate is assumed to be a unit square ($l \times l$ with $l = 1$ m) and its motion is triggered by a body force. An analytical solution for this problem is constructed using the method of manufactured solutions: the analytical solution in terms of displacement is assumed a priori, from which the corresponding body forces are calculated. The displacement is given by

$$u_{x_1}(x_1^0, t) = B \sin\left(\frac{2\pi x_1^0}{l}\right) \sin\left(\frac{c\pi t}{l}\right), \tag{3.11}$$

$$u_{x_2}(x_2^0, t) = B \sin\left(\frac{2\pi x_2^0}{l}\right) \sin\left(\frac{c\pi t}{l} + \pi\right). \tag{3.12}$$

Here, $\rho^0 = 10^3$ kg/m^3, $B = 0.05$ m, and $E = 10^7$ Pa. The corresponding body forces [29] are equal to

$$g_{x_1} = \pi^2 u_{x_1} \left(\frac{4\mu}{\rho^0} - \frac{E}{\rho^0} - 4\frac{\lambda\left[\ln(F_{11}F_{22}) - 1\right] - \mu}{\rho^0 F_{11}^2} \right), \tag{3.13}$$

$$g_{x_2} = \pi^2 u_{x_2} \left(\frac{4\mu}{\rho^0} - \frac{E}{\rho^0} - 4\frac{\lambda\left[\ln(F_{11}F_{22}) - 1\right] - \mu}{\rho^0 F_{22}^2} \right), \tag{3.14}$$

where λ is the Lamé constant, μ is the shear modulus, and F_{11} and F_{22} are the normal components of the deformation gradient. F_{11} and F_{22} are defined as

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \qquad \mu = \frac{E}{2(1+\nu)}, \tag{3.15}$$

$$F_{11} = 1 + 2B\pi\cos\left(2\pi x_1^0\right)\sin\left(\sqrt{E/\rho^0}\,\pi t\right), \tag{3.16}$$

$$F_{22} = 1 + 2B\pi\cos\left(2\pi x_2^0\right)\sin\left(\sqrt{E/\rho^0}\,\pi t + \pi\right). \tag{3.17}$$

It is assumed that Poisson's ratio is equal to $\nu = 0.3$. This problem is simulated with MPM and PS-MPM, using an unstructured triangular grid with material points initialized uniformly over the domain, as shown in Figure 3.6. A time step size and the total simulation time are set to $\Delta t = 2.25 \cdot 10^{-4}$ s and $T = 0.02$ s, respectively.
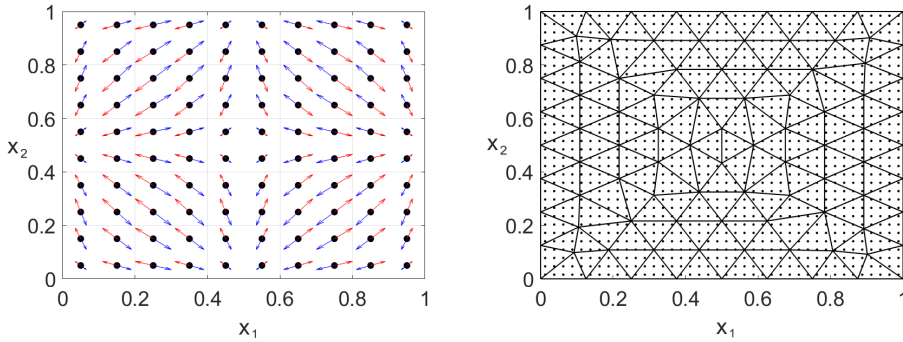
Figure 3.6: The exact solution where particles (marked with dots) move back and forth along the marked vectors (left) and the unstructured grid with the initial particle configuration (right). The axes are given in meters. Both figures were obtained in collaboration with Pascal de Koster.

3.4.2. B-SPLINE MATERIAL-POINT METHOD
The performance of BSMPM is illustrated by means of the one-dimensional vibrating bar with the parameters from set 1 in Table 3.1. Figure 3.7 shows that the method completely removes the unphysical oscillations in the velocity field that causes grid crossing artifacts in standard MPM. In addition, it demonstrates the close agreement of BSMPM with the analytical solution.

3
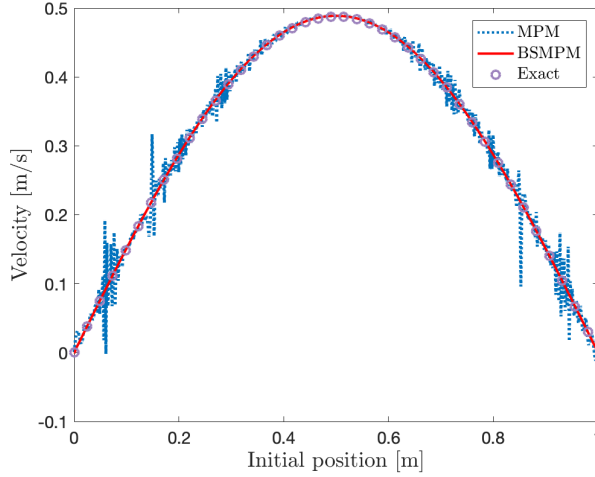
Figure 3.7: Stress distribution in the vibration bar problem with grid crossing. The results are obtained with standard MPM and BSMPM using the parameters from value set 1 in Table 3.1.

BSMPM results are also provided and discussed in detail in Chapter 5. It is shown that in some cases (e.g., Figure 5.10), the use of B-spline basis functions causes oscillations at the boundary of the domain, which prevents the method from achieving higher-order convergence.

3.4.3. MAPPING OF MATERIAL-POINT DATA WITH CUBIC-SPLINE FUNCTION RECONSTRUCTION
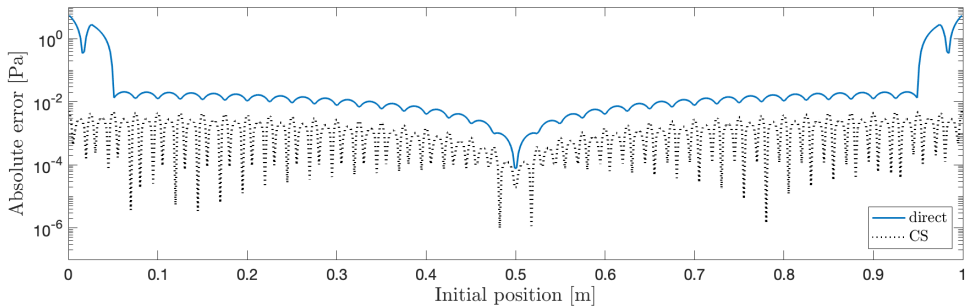
Figure 3.8: Absolute error obtained using BSMPM with and without CS mapping for the stress distribution in the vibrating bar problem without grid crossing. The results are computed using the parameters from set 2.

For BSMPM, replacing direct material-point data projection to the background grid by the mapping with cubic-spline reconstruction can be used to eliminate the oscillations at the domain boundary. This is shown in Figure 3.8. In addition to reducing the absolute error at the boundary, the proposed approach significantly decreases the value

of the error over the complete domain. Finally, for this benchmark, the mapping with cubic-spline reconstruction leads to third-order convergence for displacement and velocity, and second-order convergence for the stress (see Figure 5.9). A lower order of convergence in terms of stress is caused by its computation from the displacement by taking the derivative, instead of discretizing the stress field directly.

3.4.4. APPLICATION TO UNSTRUCTURED TRIANGULAR GRIDS USING POWELL-SABIN SPLINES

Figure 3.9 illustrates that Powell-Sabin splines can be successfully used to extend BSMPM to unstructured triangular grids. The results are provided in terms of the normal stress in the x_1-direction. It is shown that the proposed approach eliminates the grid-crossing errors observed with standard MPM and accurately resembles the analytical solution. Although the method demonstrates excellent results for this benchmark, its application to more complex problems is problematic. Further research is required to assure the robustness of the presented method to computations that involve partially filled elements. De Koster et al. [118] discuss this issue in detail.
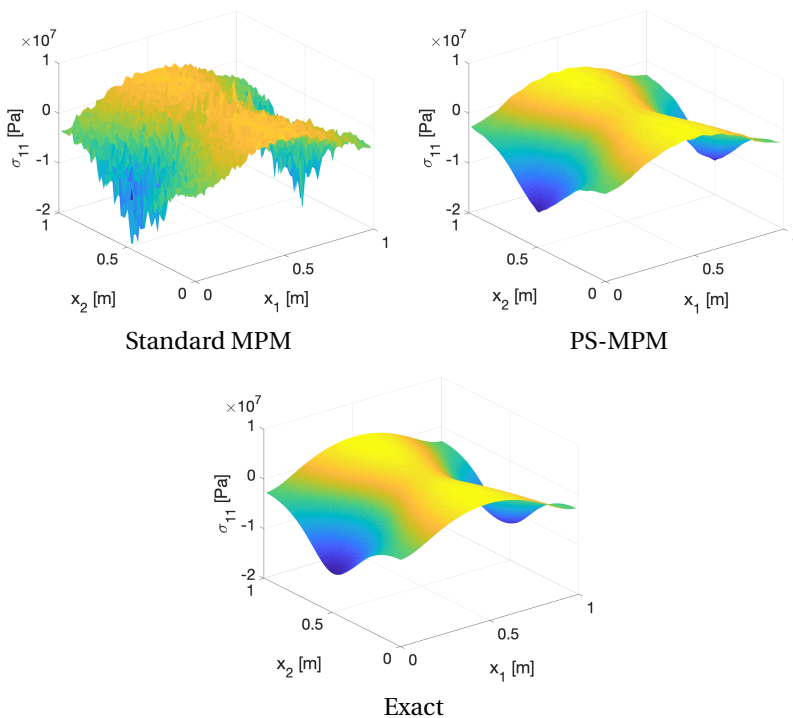
Figure 3.9: The interpolated particle stress in the x-direction at $t = 0.016$ s. The figure was obtained in collaboration with Pascal de Koster.

3.5. CONCLUSIONS

This chapter presents a possible way to use B-spline basis functions within an MPM algorithm. The basis functions are constructed from a knot vector that contains the knots over the complete domain, and are computed using the Cox-de Boor formula. The use of B-spline basis functions in MPM can be combined with a mapping technique that allows for the projection of the material-point data to the background grid using cubic-spline interpolation. The numerical results show that this modified technique provides more accurate results on the boundaries of the domain, yields higher-order convergence, and reduces the absolute error. Furthermore, this section discusses the extension of BSMPM to unstructured triangular grids by employing Powell-Sabin splines. The approach demonstrates promising results for relatively simple problems, but has to be further investigated for its successful application to problems that contain partially filled elements.

4

COMPARISON AND UNIFICATION WITH OPTIMAL TRANSPORTATION MESHFREE METHOD

Both the material-point method (MPM) and optimal transportation meshfree (OTM) method have been developed to efficiently solve partial differential equations that are based on the conservation laws from continuum mechanics. However, the methods are derived in a different fashion and have been studied independently of one another. In this chapter, we provide a direct step-by-step comparison of the MPM and OTM algorithms and introduce a novel unified approach that combines the design principles from B-spline MPM (BSMPM) and OTM methods. This chapter is structured as follows. Section 4.1 gives a short introduction. Section 4.2 describes the OTM scheme and provides its computational algorithm. Section 4.3 gives an overview of local maximum-entropy basis functions, which are typically used within the OTM method. Section 4.4 gives a comparison of the algorithms. After that, Section 4.5 introduces the unified approach. Section 4.6 reports the results that show the difference between the MPM and OTM methods numerically. In addition, it demonstrates the results obtained with our unified approach and gives a comparison with BSMPM. Finally, Section 4.7 summarizes the main conclusions of this study.

4.1. INTRODUCTION

The optimal transportation meshfree (OTM) method [32] has been developed to simulate general solid and fluid flows and applied to a wide range of problems [139–142]. The OTM method uses the concepts from optimal transportation theory (an overview is provided by Villani [143]) to translate the mass and momentum balance equations into a minimization problem for the total action of the solid over a time interval. OTM employs two sets of points: nodal points and material points. Nodal points carry information about the positions, while material points represent the continuum. This updated Lagrangian method is typically used with maximum entropy (maxent) basis functions [144, 145] that are fully defined by the nodal set and the domain of analysis.

Many similarities can be found between MPM and the OTM method. For instance, they both employ the idea of material points that represent the continuum, but are not used directly to compute the solution of the governing equations. Moreover, an alternative derivation of the OTM scheme has been provided by Weißenfels and Wriggers [146], where the method is obtained from the weak form of the equation of motion. Despite this, MPM and the OTM method have evolved and been studied independently from each other. An in-depth analysis and direct comparison of the methods provides a better understanding of their relation, with potential improvements of MPM based on the present knowledge of the OTM method and vice versa.

The presented study consists essentially of two parts. The first part offers new insights into the relation between the MPM and OTM methods by drawing a detailed comparison of their algorithms. Based on this comparison, it identifies the conditions under which the two approaches can be related to each other, and highlights their fundamental differences. The second part of the study presents a novel unified approach that combines the design principles from the B-spline MPM (BSMPM) and OTM methods. The main idea behind the approach is to use the similarities between B-spline and maxent basis functions, which are typically used in conjunction with the OTM method [147]. For example, both of them are non-negative, smooth, and possess the partition of unity property. The proposed approach is significantly cheaper than the standard OTM method and allows for the use of a consistent mass matrix without stability issues that are typically encountered in MPM computations. Although in this chapter, the unified approach is applied to relatively simple examples, all derivations are presented in their general form enabling its straightforward extension to more complex problems.

4.2. OPTIMAL TRANSPORTATION MESHFREE METHOD

The OTM method is a meshless updated Lagrangian method that is based on the concepts from optimal transportation theory. In contrast to MPM, the OTM method explicitly includes the dependence of the mass density on time. For an arbitrary time interval $\left[t^0, T\right]$ it assumes that the density at time t^0 and T are prescribed:

$$\rho\left(\mathbf{x}^0, t^0\right) = \rho^0\left(\mathbf{x}^0\right), \tag{4.1}$$

$$\rho\left(\mathbf{x}, T\right) = \rho^T\left(\mathbf{x}\right), \tag{4.2}$$

where $\mathbf{x}^0$ and $\mathbf{x}$ are given in Equation (2.1). Benamou and Brenier [148] note that the mass and momentum balance equations together with Equations (4.1) and (4.2) can be

translated into a minimization problem for the action of the solid over the time interval. The action over $[t^0, T]$ is given by

$$A(\rho, \mathbf{v}) = \int_{t^0}^{T} K(\rho, \mathbf{v}) \, dt, \tag{4.3}$$

where $K(\rho, \mathbf{v})$ is the kinetic energy, which is equal to

$$K(\rho, \mathbf{v}) = \int_{\Omega^T} \frac{\rho}{2} |\mathbf{v}|^2 \, d\Omega^T. \tag{4.4}$$

Although Equations (4.3) and (4.4) are expressed in terms of mass density and velocity, in the OTM framework the flow is described by means of the deformation mapping φ that is defined in Equation (2.1). The deformation mapping is related to velocity and density in the following way [32]:

$$\mathbf{v}(\mathbf{x}, t) = \frac{\partial \varphi}{\partial t}(\mathbf{x}^0, t), \tag{4.5}$$

$$\rho(\mathbf{x}, t) = \rho^0(\mathbf{x}^0) / \det(\nabla \varphi(\mathbf{x}^0, t)). \tag{4.6}$$

Benamou and Brenier [148] also demonstrate that the deformation mapping that minimizes the action in Equation (4.3) is given in terms of McCann's displacement interpolation [149]:

$$\varphi(\mathbf{x}^0, t) = \frac{T - t}{T - t^0} \mathbf{x}^0 + \frac{t - t^0}{T - t^0} \varphi(\mathbf{x}^0, T). \tag{4.7}$$

Here, $\varphi(\mathbf{x}^0, T)$ is the optimal transportation map of ρ^0 into ρ^T with respect to the cost function [32]:

$$C^{0 \to T}(\gamma) = \int_{\Omega^0} |\gamma(\mathbf{x}^0, T) - \mathbf{x}^0|^2 \rho^0(\mathbf{x}^0) \, d\Omega^0, \tag{4.8}$$

in which $\gamma(\mathbf{x}^0, t)$ is a generic mapping of mass density.

To generate a numerical scheme, Equation (4.3) is discretized in space and time. The time interval $[t^0, T]$ is divided into sub-intervals $[t^s, t^{s+1}]$ with $s = 0, 1, \ldots, N-1$, to which the above theory developed for $[t^0, T]$ is still applicable. The OTM method approximates a flow using the concept of the free energy of the solid U [140]:

$$U(\varphi(\mathbf{x}^s, t)) = \int_{\Omega^s} f(\nabla \varphi(\mathbf{x}^s, t)) \rho^s(\mathbf{x}^s) \, d\Omega^s, \tag{4.9}$$

where f is the local free-energy density per unit volume. Furthermore, the method employs the Wasserstein distance d_W between mass densities at two consecutive time instances [32, 139]:

$$d_W^2(\rho^s(\mathbf{x}^s), \rho^{s+1}(\mathbf{x}^{s+1})) = \inf_{\substack{\gamma: \Omega^s \to \Omega^{s+1} \\ \rho^s = \rho^{s+1} \det(\nabla \gamma(\mathbf{x}^s, t^{s+1}))}} \int_{\Omega^s} |\gamma(\mathbf{x}^s, t^{s+1}) - \mathbf{x}^s|^2 \rho^s(\mathbf{x}^s) \, d\Omega^s$$

$$= \inf_{\substack{\gamma: \Omega^s \to \Omega^{s+1} \\ \rho^s = \rho^{s+1} \det(\nabla \gamma(\mathbf{x}^s, t^{s+1}))}} C^{s \to s+1}(\gamma). \tag{4.10}$$
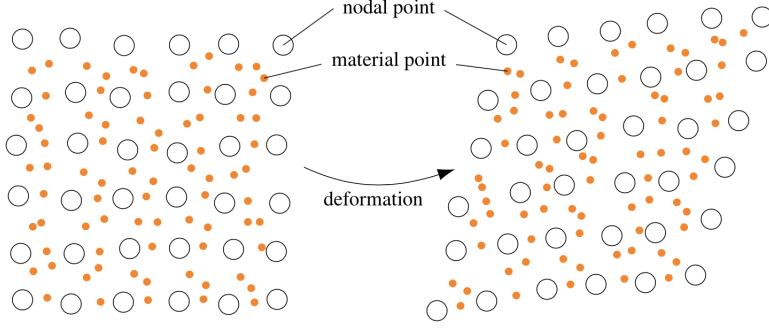
Figure 4.1: Schematic representation of the OTM method. Discretization of the domain by nodes (nodal points) and material points before and after deformation.

4

For elastic materials in unforced systems, the semi-discrete action sum is equal to [139]

$$A^{0 \to N}\left(\varphi^0, \ldots, \varphi^N\right) = \sum_{s=0}^{N-1} \left(\frac{1}{2} \frac{d_W^2\left(\rho^s, \rho^{s+1}\right)}{\left(t^{s+1} - t^s\right)^2} - \frac{1}{2}\left(U\left(\varphi^s\right) + U\left(\varphi^{s+1}\right)\right) \right)\left(t^{s+1} - t^s\right), \quad (4.11)$$

where φ^s is the deformation mapping at time t^s. For the spatial discretization of Equation (4.11), the OTM method employs two sets of points: nodal points and material points (see Figure 4.1). Nodal points carry position information, while material points represent the continuum body and arise from the spatial approximation of the mass densities by point masses:

$$\rho^s(\mathbf{x}) \approx \rho_h^s(\mathbf{x}) = \sum_{p=1}^{\mathcal{M}} m_p \delta\left(\mathbf{x} - \mathbf{x}_p^s\right), \quad (4.12)$$

where $\delta\left(\mathbf{x} - \mathbf{x}_p^s\right)$ is the Dirac delta distribution centered at $\mathbf{x}_p^s$. Material points are convected by the deformation:

$$\mathbf{x}_p^{s+1} = \varphi_h^{s \to s+1}\left(\mathbf{x}_p^s\right), \quad (4.13)$$

in which $\varphi_h^{s \to s+1}(\mathbf{x})$ is the incremental deformation map. Fedeli et al. [141] explain that it can be described by general linear interpolation schemes of the form:

$$\varphi_h^{s \to s+1}(\mathbf{x}) = \sum_{i=1}^{\mathcal{N}} \phi_i^s(\mathbf{x}) \mathbf{x}_i^{s+1}, \quad (4.14)$$

whereby the basis functions are assumed to be *consistent*. Consistent basis functions satisfy the following conditions:

- Partition of unity property:

$$\sum_{i=1}^{\mathcal{N}} \phi_i^s(\mathbf{x}) = 1 \qquad \forall \, \mathbf{x} \in \Omega^s. \quad (4.15)$$

- *Linear completeness*:

$$\sum_{i=1}^{\mathcal{N}} \mathbf{x}_i^s \phi_i^s(\mathbf{x}) = \mathbf{x} \qquad \forall\, \mathbf{x} \in \Omega^s. \tag{4.16}$$

Moreover, material points carry a fixed mass, serve as integration points, and store all local state data. In order to fully discretize Equation (4.11), the Wasserstein distance $d_W^2\left(\rho_h^s, \rho_h^{s+1}\right)$ is expressed as follows:

$$d_W^2\left(\rho_h^s, \rho_h^{s+1}\right) = \inf_{\substack{\gamma:\, \Omega^s \to \Omega^{s+1} \\ \rho_h^s = \rho_h^{s+1}\det(\nabla\gamma(\mathbf{x}^s, t^{s+1}))}} \int_{\Omega^s} \left|\gamma\left(\mathbf{x}^s, t^{s+1}\right) - \mathbf{x}^s\right|^2 \rho_h^s\left(\mathbf{x}^s\right) d\Omega^s \tag{4.17}$$

$$= \int_{\Omega^s} \left|\varphi\left(\mathbf{x}^s, t^{s+1}\right) - \mathbf{x}^s\right|^2 \rho_h^s\left(\mathbf{x}^s\right) d\Omega^s \tag{4.18}$$

$$= \int_{\Omega^s} \left|\varphi\left(\mathbf{x}^s, t^{s+1}\right) - \mathbf{x}^s\right|^2 \sum_{p=1}^{\mathcal{M}} m_p \delta\left(\mathbf{x}^s - \mathbf{x}_p^s\right) d\Omega^s \tag{4.19}$$

$$= \sum_{p=1}^{\mathcal{M}} m_p \int_{\Omega^s} \left|\varphi\left(\mathbf{x}^s, t^{s+1}\right) - \mathbf{x}^s\right|^2 \delta\left(\mathbf{x}^s - \mathbf{x}_p^s\right) d\Omega^s \tag{4.20}$$

$$= \sum_{p=1}^{\mathcal{M}} m_p \left|\varphi\left(\mathbf{x}_p^s, t^{s+1}\right) - \mathbf{x}_p^s\right|^2 \tag{4.21}$$

$$= \sum_{p=1}^{\mathcal{M}} m_p \left|\mathbf{x}_p^{s+1} - \mathbf{x}_p^s\right|^2. \tag{4.22}$$

Here, we used the fact that $\varphi\left(\mathbf{x}^s, t^{s+1}\right)$ is the optimal transportation map of ρ^s into ρ^{s+1} with respect to $C^{s\to s+1}$ for the second equality, while Equation (4.12) for the third equality. Moreover, a similar procedure is followed to approximate the free energy $U\left(\varphi_h^s\right)$:

$$U\left(\varphi_h^s\right) = \int_{\Omega^s} f\left(\nabla\varphi_h^s\right) \rho_h^s d\Omega^s \tag{4.23}$$

$$= \int_{\Omega^s} f\left(\nabla\varphi_h^s\right) \sum_{p=1}^{\mathcal{M}} m_p \delta\left(\mathbf{x} - \mathbf{x}_p^s\right) d\Omega^s \tag{4.24}$$

$$= \sum_{p=1}^{\mathcal{M}} \int_{\Omega^s} f\left(\nabla\varphi_h^s\right) m_p \delta\left(\mathbf{x} - \mathbf{x}_p^s\right) d\Omega^s \tag{4.25}$$

$$= \sum_{p=1}^{\mathcal{M}} m_p f\left(\nabla\varphi_h^s\right). \tag{4.26}$$

Substituting Equations (4.22) and (4.26) into Equation (4.11) leads to the fully discrete action:

$$A_h^{0\to N}\left(\varphi_h^0, \ldots, \varphi_h^N\right) = \sum_{s=0}^{N-1} \sum_{p=1}^{\mathcal{M}} \left(\frac{m_p}{2} \frac{\left|\mathbf{x}_p^{s+1} - \mathbf{x}_p^s\right|^2}{\left(t^{s+1} - t^s\right)^2} - \frac{m_p}{2} \left(f\left(\nabla\varphi_h^s\left(\mathbf{x}_p^s\right)\right)\right.\right.$$
$$\left.\left. + f\left(\nabla\varphi_h^{s+1}\left(\mathbf{x}_p^{s+1}\right)\right)\right)\right)\left(t^{s+1} - t^s\right). \tag{4.27}$$

The OTM algorithm originates from applying the discrete Hamilton's principle [150] to the fully discrete action [32, 151].

4.2.1. ALGORITHM

Input: Nodal coordinates $\mathbf{x}_i^0$, material-point coordinates $\mathbf{x}_p^{-1}$, $\mathbf{x}_p^0$, volumes V_p^0, densities ρ_p^0, masses m_p, deformation gradients $\mathbf{F}_p^0$, stresses $\boldsymbol{\sigma}_p^0$, body forces $\mathbf{b}_p$

1   Set $s = 0$

2   **while** $s < N$ **do**

3   Compute basis functions $\phi_i^s\left(\mathbf{x}_p^s\right)$ and derivatives $\nabla\phi_i^s\left(\mathbf{x}_p^s\right)$ from advected nodal set $\{\mathbf{x}_i^s\}$ and advected material-point set $\{\mathbf{x}_p^s\}$

4   Compute mass matrix $\mathbf{M}^s$, linear momentum vector $\bar{\mathbf{q}}_k^s$, and force vector $\bar{\mathbf{f}}_k^s$:

$$M_{ij}^s = \sum_{p=1}^{\mathcal{M}} \phi_i^s\left(\mathbf{x}_p^s\right) m_p \phi_j^s\left(\mathbf{x}_p^s\right) \tag{4.28}$$

$$q_{k,i}^s = \sum_{p=1}^{\mathcal{M}} \phi_i^s\left(\mathbf{x}_p^s\right) m_p \frac{x_{k,p}^s - x_{k,p}^{s-1}}{t^s - t^{s-1}} \tag{4.29}$$

$$f_{k,i}^s = \sum_{p=1}^{\mathcal{M}} \left(\sum_{l=1}^{3} \frac{\partial\phi_i^s}{\partial x_l}\left(\mathbf{x}_p^s\right) \sigma_{kl,p}^s + \phi_i^s\left(\mathbf{x}_p^s\right) \rho_p^s b_{k,p} \right) V_p^s \tag{4.30}$$

5   Update nodal coordinates:

$$\bar{\mathbf{x}}_k^{s+1} = \bar{\mathbf{x}}_k^s + \left(t^{s+1} - t^s\right)\left(\mathbf{M}^s\right)^{-1}\left(\bar{\mathbf{q}}_k^s + \frac{t^{s+1} - t^{s-1}}{2}\bar{\mathbf{f}}_k^s\right) \tag{4.31}$$

6   Update material-point coordinates:

$$\mathbf{x}_p^{s+1} = \varphi_h^{s \to s+1}(\mathbf{x}_p^s) \tag{4.32}$$

7   Update material-point volumes:

$$V_p^{s+1} = \det\left(\nabla\varphi_h^{s \to s+1}\left(\mathbf{x}_p^s\right)\right) V_p^s \tag{4.33}$$

8   Update material-point densities:

$$\rho_p^{s+1} = \frac{m_p}{V_p^{s+1}} \tag{4.34}$$

9   Update material-point deformation gradients:

$$\mathbf{F}_p^{s+1} = \nabla\varphi_h^{s \to s+1}\left(\mathbf{x}_p^s\right)\mathbf{F}_p^s \tag{4.35}$$

10   Compute material-point stresses $\boldsymbol{\sigma}_p^{s+1}$ from $\mathbf{F}_p^{s+1}$ using the constitutive equation (e.g., Equation 2.13 or 2.14)

12   Set $s = s + 1$

13 **end**

Algorithm 3: Optimal transportation meshfree method.

A basic OTM algorithm for a solid material is summarized in Algorithm 3. For more details on the OTM method, we refer to the work of Li et al. [32] and Habbal [151].

4.3. LOCAL MAXIMUM-ENTROPY BASIS FUNCTIONS

Maxent basis functions were introduced by Sukumar [144] for the construction of polygonial interpolants. Arroyo and Ortiz [145] presented local maxent basis functions and first used them within a meshfree method. After that, local maxent basis functions have been integrated within several meshfree schemes such as point collocation methods [152]. The schemes that combined meshfree methods with maxent basis function have been applied to a variety of problems including simulations of shear-deformable plates [153] and thin-shell analysis [154].

The construction of maxent basis functions combines elements from probability theory and optimization. In fact, within a convex hull of the nodal set $\{\mathbf{x}_i\}$ (i.e., the smallest convex set that contains all nodes), the set of local maxent basis functions $\{\phi_i(\mathbf{x}) \geq 0\}_{i=1}^{\mathcal{N}}$ form the solution of the following constrained optimization problem [155]:

$$\max_{\phi \,\in\, \mathbb{R}_+^{\mathcal{N}}} \; -\sum_{i=1}^{\mathcal{N}} \phi_i(\mathbf{x}) \ln\left(\frac{\phi_i(\mathbf{x})}{w_i(\mathbf{x})}\right) \tag{4.36}$$

subject to

$$\sum_{i=1}^{\mathcal{N}} \phi_i(\mathbf{x}) = 1, \tag{4.37}$$

$$\sum_{i=1}^{\mathcal{N}} \phi_i(\mathbf{x})(\mathbf{x}_i - \mathbf{x}) = 0, \tag{4.38}$$

where $w_i(\mathbf{x})$ is a non-negative weight function or prior estimate of ϕ_i. The solution of this problem is typically found using the method of Lagrange multipliers [156] and can be written as

$$\phi_i(\mathbf{x}) = \frac{Z_i(\mathbf{x}, \boldsymbol{\lambda})}{Z(\mathbf{x}, \boldsymbol{\lambda})} \tag{4.39}$$

with

$$Z_i(\mathbf{x}, \boldsymbol{\lambda}) = w_i(\mathbf{x}) \exp(-\boldsymbol{\lambda} \cdot (\mathbf{x}_i - \mathbf{x})), \tag{4.40}$$

$$Z(\mathbf{x}, \boldsymbol{\lambda}) = \sum_i Z_i(\mathbf{x}, \boldsymbol{\lambda}), \tag{4.41}$$

where $\boldsymbol{\lambda}$ represents the Lagrange multipliers.

In practice, the primal problem of maximization is transferred into the dual problem of minimization [144]. Considering the new formulation, $\boldsymbol{\lambda}$ is expressed as [157, 158]

$$\boldsymbol{\lambda} = \arg\min \ln(Z). \tag{4.42}$$

The Lagrange multipliers are typically found via Newton's method. The solution procedure of this method (e.g., [159]) requires an initial guess for $\boldsymbol{\lambda}$ and both first- and second-order partial derivatives of $\ln(Z)$ with respect to $\boldsymbol{\lambda}$. Figure 4.2 demonstrates an example of the convergence path of the Newton's method for a one-dimensional computation.
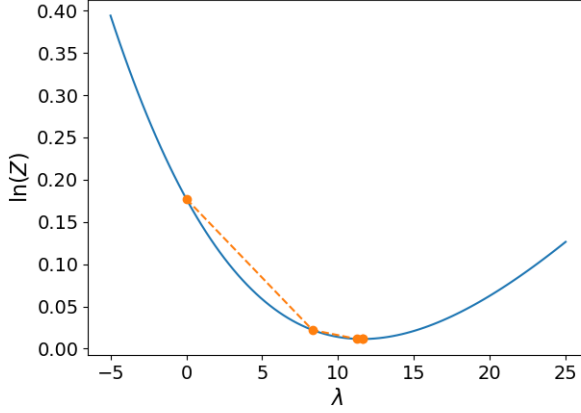
Figure 4.2: Example of a convergence path of Newton's method for a one-dimensional computation.

The first derivatives of the local maxent basis functions are given by (e.g., [160])

$$\nabla \phi_i = \phi_i \left((\mathbf{x} - \mathbf{x}_i) \cdot \left(\mathbf{H}^{-1} - \mathbf{H}^{-1}\mathbf{A} \right) - \frac{\nabla w_i}{w_i} + \sum_{j=1}^{\mathcal{N}} \phi_j \frac{\nabla w_j}{w_j} \right), \qquad (4.43)$$

in which the matrices $\mathbf{H}$ and $\mathbf{A}$ are computed in the following way:

$$\mathbf{H} = \sum_{j=1}^{\mathcal{N}} \phi_j (\mathbf{x} - \mathbf{x}_j) \otimes (\mathbf{x} - \mathbf{x}_j), \qquad (4.44)$$

$$\mathbf{A} = \sum_{j=1}^{\mathcal{N}} \phi_j (\mathbf{x} - \mathbf{x}_j) \otimes \frac{\nabla w_j}{w_j}. \qquad (4.45)$$

Here, $\otimes$ is the dyadic product (i.e., the dyadic product of any two vectors is equal to $\mathbf{a} \otimes \mathbf{b} = \mathbf{a}\mathbf{b}^{\mathsf{T}}$).

4.3.1. WEIGHT FUNCTIONS

In the above description, the weight function remained unspecified due to a large number of viable options (e.g., Gaussian prior, cubic or quartic spline). In general, the prior functions are defined by means of the normalized radius of the support domain, r_i for node i:

$$r_i(\mathbf{x}) = \frac{||\mathbf{x} - \mathbf{x}_i||}{d_i}, \qquad (4.46)$$

where $|| \cdot ||$ is the L^2-norm, and d_i is the size of the domain of support of node i, which is a user-defined parameter. In this thesis, d_i is equal to

$$d_i = d_{\max} \zeta_i, \qquad (4.47)$$

in which $d_{\max}$ is a factor with a typical value between 2.0 and 4.0 (this value is selected by the user), and ζ_i is the distance between node i and its nearest neighboring node.

Alternative definitions of the normalized radius of the support domain can be found, for instance, in the work of Sukumar and Wright [161] and Yaw et al. [160].

Cubic spline weight functions, which are used in this thesis, are given by

$$w_i(\mathbf{x}) = \begin{cases} \frac{2}{3} - 4r_i^2(\mathbf{x}) + 4r_i^3(\mathbf{x}), & \text{for } 0 < r_i(\mathbf{x}) \le \frac{1}{2}, \\ \frac{4}{3} - 4r_i(\mathbf{x}) + 4r_i^2(\mathbf{x}) - \frac{4}{3}r_i^3(\mathbf{x}), & \text{for } \frac{1}{2} < r_i(\mathbf{x}) \le 1, \\ 0, & \text{for } r_i(\mathbf{x}) > 1. \end{cases} \tag{4.48}$$

In one dimension, the derivative of a cubic spline basis function is computed as

$$\frac{dw_i}{dx} = \frac{dw_i}{dr_i}\frac{dr_i}{dx} = \begin{cases} \frac{1}{d_i}\left(-8r_i(x) + 12r_i^2(x)\right)\text{sign}(x - x_i), & \text{for } 0 < r_i(x) \le \frac{1}{2}, \\ \frac{1}{d_i}\left(-4 + 8r_i(x) - 12r_i^2(x)\right)\text{sign}(x - x_i), & \text{for } \frac{1}{2} < r_i(x) \le 1, \\ 0, & \text{for } r_i(x) > 1. \end{cases} \tag{4.49}$$

In two dimensions, the derivatives are obtained from

$$\nabla w_i = \left[\frac{dw_i}{dr_i}\frac{\partial r_i}{\partial x_1}, \frac{dw_i}{dr_i}\frac{\partial r_i}{\partial x_2}\right]^\mathsf{T} \tag{4.50}$$

In the above expression, the partial derivatives of r_i can be written as

$$\frac{\partial r_i}{\partial x_k} = \frac{x_k - x_{k,i}}{d_i \|\mathbf{x} - \mathbf{x}_i\|}. \tag{4.51}$$

Figure 4.3 shows the one-dimensional cubic spline basis functions and their derivatives, as well as the corresponding local maxent basis functions together with the derivatives. Figure 4.5 illustrates the two-dimensional prior functions, local maxent basis functions, and the partial derivatives of the local maxent basis functions for an interior node and a corner node. The domain discretization as well as the evaluation points are shown in Figure 4.4. The Python implementation of the maxent basis functions, which was used in this thesis, is validated by a comparison with the Matlab implementation provided by Ortiz-Bernardin [162].

4.3.2. PROPERTIES

Local maxent basis functions possess many desirable properties for meshfree algorithms. First of all, they are entirely defined by the nodal set and the domain of analysis. They are also non-negative, satisfy the partition of unity property, and provide an exact approximation for affine functions [145]. Furthermore, the local maxent basis functions have the so-called weak Kronecker-delta property [145]. The weak Kronecker-delta property implies that the Kronecker-delta property holds only for the basis functions ϕ_i corresponding to the nodes x_i on the boundary of a convex hull. The Kronecker-delta property can be stated as follows:

$$\phi_i(x_i) = \delta_{ij} = \begin{cases} 1 & \text{for } i = j, \\ 0 & \text{otherwise}, \end{cases} \tag{4.52}$$

In other words, the weak Kronecker-delta property states that the maxent basis functions are interpolatory at the boundary nodes, which allows for direct imposition of Dirichlet boundary conditions [145, 146].
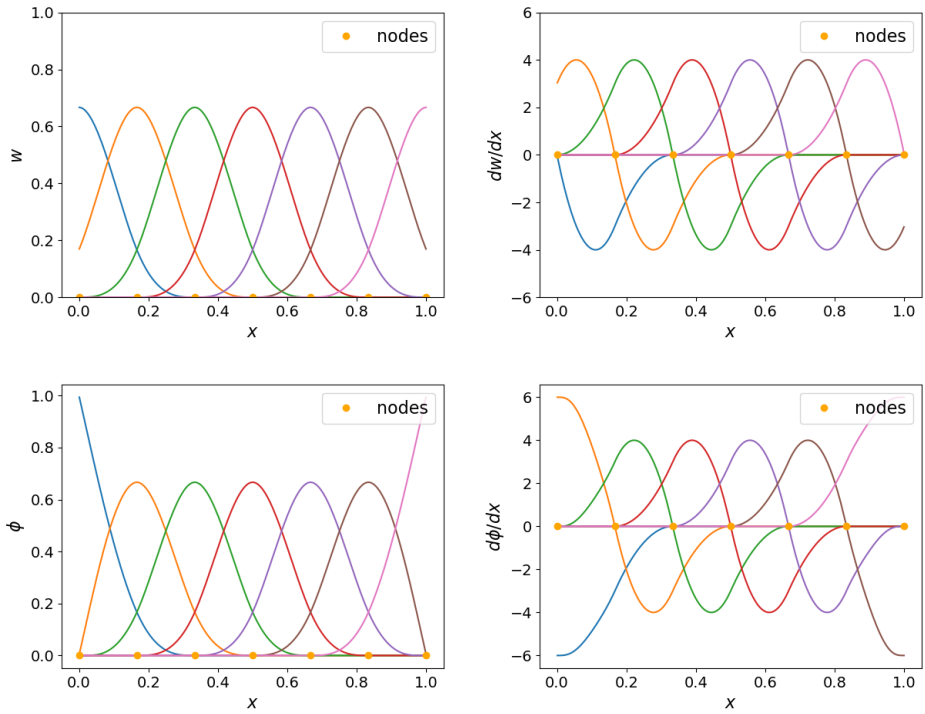
Figure 4.3: 1D cubic spline weight functions, local maxent basis functions, and their derivatives for $d_{\max} = 2.0$.
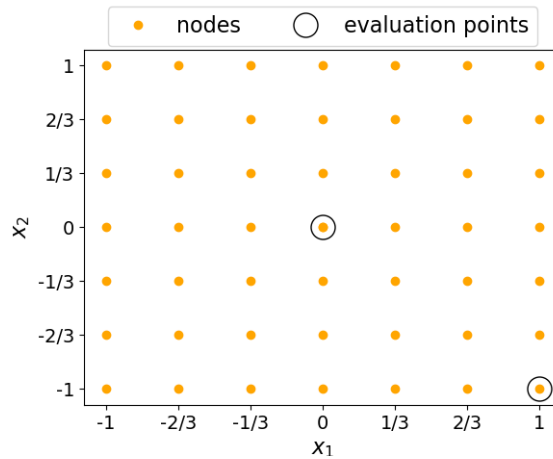

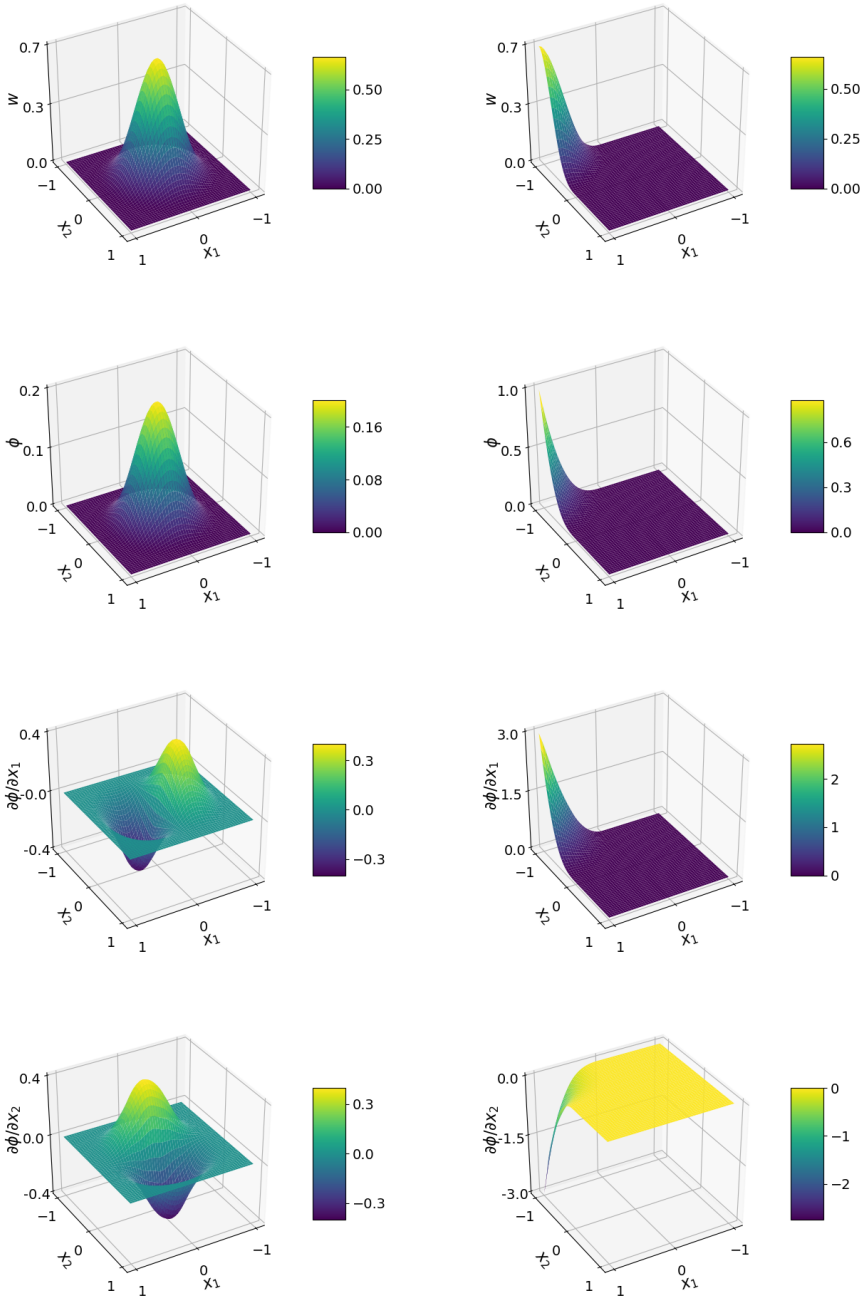Figure 4.4: Example of the discretization of a two-dimensional domain.

Figure 4.5: Examples of 2D weight functions, local maxent basis functions and their derivatives corresponding to cubic spline weight functions with $d_{max} = 2.0$. The domain descritization is shown in Figure 4.4, the basis functions are evaluated at $(0,0)$ (left) and $(1,-1)$ (right).

However, the local maxent basis functions are defined only within a convex hull of the nodal set. If non-convex domains are considered, the basis functions lose the weak Kronecker-delta property at the non-convex parts of the domain [145]. In addition, the calculation of Lagrange multipliers is numerically challenging [151] and frequently requires significant computational costs.

4.4. COMPARISON OF ALGORITHMS

While the derivations of MPM and the OTM method are fundamentally different, the resulting algorithms have many similarities. This section provides a side-by-side comparison of the computational steps from Algorithm 1 (USL-MPM) and 3 (OTM). Based on this comparison, it summarizes the conditions required to further relate the methods and highlights the principal differences between them.

- In the beginning of the simulation, both algorithms intialize the nodal coordinates and material-point properties. While the MPM computation requires the material-point velocity for time step $s = 0$, the OTM method expects the material-point positions to be known at $s = -1$. This difference arises from the explicit definition of the material-point velocity in MPM and its implicit use in the OTM method. This is further explained in the discussion of step 4 of both schemes.

- After the initialization phase, the time step counter s is set to zero, which identifies the start of the solution phase. At the end of each time step, s is increased by one until the maximum number of time steps N is reached.

- In step 3, the schemes compute the basis functions and their derivatives. However, in the OTM method the basis functions are updated in each time step based on the nodal velocities, while in MPM the basis functions remain fixed over time. This is an important difference between the methods. To distinguish between the basis functions, the OTM basis functions are denoted by ϕ_i^s and the MPM basis functions by ϕ_i^0.

- On the other hand, the schemes can be related in step 4. Assuming that the material-point velocity in MPM can be written as

$$\mathbf{v}_p^s = \frac{1}{t^s - t^{s-1}} \left(\mathbf{x}_p^s - \mathbf{x}_p^{s-1} \right). \tag{4.53}$$

A direct substitution of Equation (4.53) into the expression for the linear momentum in the MPM algorithm (Equation 2.32) leads to the linear momentum formula used in the OTM method (Equation 4.29).

- Furthermore, steps 5 and 6 of the USL algorithm are implicitly included in step 5 of the OTM algorithm, where the nodal coordinates at time step $s + 1$ are computed. More precisely, from step 5 in the OTM algorithm, we obtain

$$\Delta \bar{\mathbf{x}}_k^{s+1} = \bar{\mathbf{x}}_k^{s+1} - \bar{\mathbf{x}}_k^s = \left(t^{s+1} - t^s \right) \left(\mathbf{M}^s \right)^{-1} \left(\bar{\mathbf{q}}_k^s + \frac{t^{s+1} - t^{s-1}}{2} \bar{\mathbf{f}}_k^s \right). \tag{4.54}$$

At the same time, the incremental nodal displacement in MPM can be written as

$$\Delta \bar{\mathbf{x}}_k^{s+1} = \left(t^{s+1} - t^s\right)\left(\left(\mathbf{M}^s\right)^{-1} \bar{\mathbf{q}}_k^s + \left(t^{s+1} - t^s\right)\bar{\mathbf{a}}_k^s\right)$$
$$= \left(t^{s+1} - t^s\right)\left(\mathbf{M}^s\right)^{-1}\left(\bar{\mathbf{q}}_k^s + \left(t^{s+1} - t^s\right)\bar{\mathbf{f}}_k^s\right). \tag{4.55}$$

It can be seen that Equation (4.55) is equal to Equation (4.54) for constant time step sizes. From this it can be concluded that the update of nodal positions is identical for both methods when the time-step size is fixed.

- The definition of the incremental transport map (Equation 4.14) implies that, in step 6 of the OTM algorithm, material-point positions are obtained from

$$\mathbf{x}_p^{s+1} = \sum_{i=1}^{\mathcal{N}} \phi_i^s\left(\mathbf{x}_p^s\right)\mathbf{x}_i^{s+1}, \tag{4.56}$$

while step 7 in the MPM algorithm states that

$$\mathbf{x}_p^{s+1} = \mathbf{x}_p^s + \sum_{i=1}^{\mathcal{N}} \phi_i^0\left(\mathbf{x}_p^s\right)\Delta\mathbf{x}_i^{s+1}. \tag{4.57}$$

The equality of the above expressions can be shown if linear completeness (see Equation 4.16) of the MPM basis functions is imposed. That is, if the following condition is satisfied:

$$\sum_{i=1}^{\mathcal{N}} \phi_i^0(\mathbf{x})\mathbf{x}_i^s = \mathbf{x} \qquad \forall\, \mathbf{x} \in \Omega^s. \tag{4.58}$$

In this case, Equation (4.57) can be rewritten as

$$\mathbf{x}_p^{s+1} = \sum_{i=1}^{\mathcal{N}} \phi_i^0\left(\mathbf{x}_p^s\right)\mathbf{x}_i^s + \sum_{i=1}^{\mathcal{N}} \phi_i^0\left(\mathbf{x}_p^s\right)\Delta\mathbf{x}_i^{s+1} = \sum_{i=1}^{\mathcal{N}} \phi_i^0\left(\mathbf{x}_p^s\right)\mathbf{x}_i^{s+1}. \tag{4.59}$$

- Moreover, the OTM scheme avoids a direct update of the material-point velocity by adopting Equation (4.53), whereas MPM performs the update in step 8. Nevertheless, assuming that

$$\sum_{i=1}^{\mathcal{N}} \phi_i^0\left(\mathbf{x}_p^s\right)\mathbf{v}_i^s = \mathbf{v}_p^s, \tag{4.60}$$

it is possible to relate the methods again. Substituting Equation (4.60) into step 8 of MPM gives

$$\mathbf{v}_p^{s+1} = \sum_{i=1}^{\mathcal{N}} \phi_i^0\left(\mathbf{x}_p^s\right)\mathbf{v}_i^s + \left(t^{s+1} - t^s\right)\sum_{i=1}^{\mathcal{N}} \phi_i^0\left(\mathbf{x}_p^s\right)\mathbf{a}_i^s = \sum_{i=1}^{\mathcal{N}} \phi_i^0\left(\mathbf{x}_p^s\right)\left(\mathbf{v}_i^s + \left(t^{s+1} - t^s\right)\mathbf{a}_i^s\right). \tag{4.61}$$

Substituting step 6 of the MPM algorithm yields

$$\mathbf{v}_p^{s+1} = \sum_{i=1}^{\mathcal{N}} \phi_i^0\left(\mathbf{x}_p^s\right)\mathbf{v}_i^{s+1}. \tag{4.62}$$

Therefore, the updated material-point velocity in MPM is equal to

$$\mathbf{v}_p^{s+1} = \frac{1}{t^{s+1} - t^s} \sum_{i=1}^{\mathcal{N}} \phi_i^0\left(\mathbf{x}_p^s\right) \Delta \mathbf{x}_i^{s+1} = \frac{1}{t^{s+1} - t^s} \left(\mathbf{x}_p^{s+1} - \mathbf{x}_p^s\right). \tag{4.63}$$

The above equalities follow from the computation of the incremental nodal displacements in steps 6, and the update of material-point coordinates in step 7 of the MPM algorithm.

The final expression in Equation (4.63) is identical to the implicit material-point velocity update in the OTM algorithm. Although an extra assumption is required to establish a connection between the methods, it does not lead to an essential disparity between them.

- Step 9 of both schemes can be shown to be identical as well. From step 6 in the OTM algorithm and the OTM definition of material-point velocity presented in Equation (4.53), it follows that

$$\nabla \varphi_h^{s \to s+1}\left(\mathbf{x}_p^s\right) = \nabla \mathbf{x}_p^{s+1} = \nabla \left(\mathbf{x}_p^s + \left(t^{s+1} - t^s\right) \mathbf{v}_p^{s+1}\right) = \mathbf{I} + \left(t^{s+1} - t^s\right) \nabla \mathbf{v}_p^{s+1}. \tag{4.64}$$

Therefore, step 9 is the same for the MPM and OTM algorithm.

- Next, the update of the material-point volumes is investigated. MPM performs this update in step 10, while the OTM method computes the volume in step 7. To prove the equivalence of those steps, we need to show that

$$\det\left(\left(\mathbf{I} + \left(t^{s+1} - t^s\right) \nabla \mathbf{v}_p^{s+1}\right) \mathbf{F}_p^s\right) V_p^0 = \det\left(\nabla \varphi_h^{s \to s+1}\left(\mathbf{x}_p^s\right)\right) V_p^s. \tag{4.65}$$

The identity is proved using mathematical induction. For $s = 0$, the MPM update can be written as

$$\det\left(\left(\mathbf{I} + (t^1 - t^0)\nabla \mathbf{v}_p^1\right) \mathbf{F}_p^0\right) V_p^0 \tag{4.66}$$

$$= \det\left(\mathbf{I} + (t^1 - t^0)\nabla \mathbf{v}_p^1\right) \det\left(\mathbf{F}_p^0\right) V_p^0 \tag{4.67}$$

$$= \det\left(\mathbf{I} + (t^1 - t^0)\nabla \mathbf{v}_p^1\right) \det(\mathbf{I}) V_p^0 \tag{4.68}$$

$$= \det\left(\mathbf{I} + (t^1 - t^0)\nabla \mathbf{v}_p^1\right) V_p^0 \tag{4.69}$$

$$= \det\left(\nabla \varphi_h^{0 \to 1}\left(\mathbf{x}_p^s\right)\right) V_p^0, \tag{4.70}$$

where the last equality is based on Equation (4.64). Thus, the statement from Equation (4.65) holds for the base case. Assuming that the desired identity also holds for time step $s \in \mathbb{N}$, the following expression is obtained:

$$\det\left(\mathbf{I} + \left(t^s - t^{s-1}\right) \nabla \mathbf{v}_p^s\right) \det\left(\mathbf{F}_p^{s-1}\right) V_p^0 = \det\left(\nabla \varphi_h^{s-1 \to s}\left(\mathbf{x}_p^{s-1}\right)\right) V_p^{s-1}. \tag{4.71}$$

The substitution of Equation (4.64) leads to the following expression:

$$\det\left(\mathbf{I} + \left(t^s - t^{s-1}\right) \nabla \mathbf{v}_p^s\right) \det\left(\mathbf{F}_p^{s-1}\right) V_p^0 = \det\left(\mathbf{I} + \left(t^s - t^{s-1}\right) \nabla \mathbf{v}_p^s\right) V_p^{s-1}. \tag{4.72}$$

It should be noted that $\det\left(\mathbf{I}+\left(t^s-t^{s-1}\right)\nabla\mathbf{v}_p^s\right)=0$ only if

$$\frac{\partial v_k}{\partial x_l} = \begin{cases} -\frac{1}{t^s-t^{s-1}} & \text{for } k=l, \\ 0 & \text{otherwise.} \end{cases} \tag{4.73}$$

For $\det\left(\mathbf{I}+\left(t^s-t^{s-1}\right)\nabla\mathbf{v}_p^s\right)\neq 0$, Equation (4.72) reduces to

$$\det\left(\mathbf{F}_p^{s-1}\right)V_p^0 = V_p^{s-1}. \tag{4.74}$$

Assuming that $\det\left(\mathbf{F}_p^{s-1}\right)\neq 0$ and $V_p^{s-1}\neq 0$, for time step $s+1$, the left-hand side of Equation (4.65) can be written as

$$\det\left(\left(\mathbf{I}+\left(t^{s+1}-t^s\right)\nabla\mathbf{v}_p^{s+1}\right)\mathbf{F}_p^s\right)V_p^0 \tag{4.75}$$

$$= \det\left(\mathbf{I}+\left(t^{s+1}-t^s\right)\nabla\mathbf{v}_p^{s+1}\right)\det\left(\mathbf{F}_p^s\right)V_p^0 \tag{4.76}$$

$$= \det\left(\mathbf{I}+\left(t^{s+1}-t^s\right)\nabla\mathbf{v}_p^{s+1}\right)\det\left(\mathbf{F}_p^s\right)\frac{V_p^{s-1}}{\det\left(\mathbf{F}_p^{s-1}\right)} \tag{4.77}$$

$$= \det\left(\mathbf{I}+\left(t^{s+1}-t^s\right)\nabla\mathbf{v}_p^{s+1}\right)\frac{V_p^s}{V_p^{s-1}}V_p^{s-1} \tag{4.78}$$

$$= \det\left(\mathbf{I}+\left(t^{s+1}-t^s\right)\nabla\mathbf{v}_p^{s+1}\right)V_p^s \tag{4.79}$$

$$= \det\left(\nabla\varphi_h^{s\to s+1}\left(\mathbf{x}_p^s\right)\right)V_p^s. \tag{4.80}$$

In the above expression, the second equality is obtained from Equation (4.74), the third equality follows from the volume update in the MPM algorithm (i.e., Equation 2.42), and the last equality is derived using Equation (4.64). Therefore, we conclude that by induction the MPM and OTM schemes update the material-point volumes in the same manner.

- The remaining steps are identical for MPM and OTM.

In this section, we have shown that under certain conditions the MPM and OTM algorithms can be related to each other. Namely, assuming a constant time step and the validity of Equations (4.53), (4.58), and (4.60) for the MPM scheme, the only difference between the algorithms emerges from the update of the basis functions and their gradients.

However, this difference is fundamental. Since the nodes are fixed to their initial positions in the MPM basis-function update (step 3 of the USL algorithm), the method is considered to be a combination of Lagrangian and Eulerian approaches. At the same time, the OTM method is an updated Lagrangian particle method. For the implementation of the methods, this implies that the OTM method only discretizes the initial material domain, while MPM discretizes the complete domain, where the material is allowed to move, as well as the initial material domain. Consequently, the OTM method does not include inactive elements.

4.5. UNIFIED APPROACH

The unified approach combines the OTM scheme and B-spline MPM (BSMPM) [28, 31]. Similarly to BSMPM, the proposed approach uses B-spline basis functions as described in Chapter 3. However, it computes them based not only on the updated material-point positions, but also on the advected degrees-of-freedom set. B-spline basis functions of any order possess the partition of unity property, but only first-order B-spline basis functions satisfy the linear completeness property. Therefore, according to the definition in Section 4.2, higher-order basis functions are not consistent. Since Algorithm 2 is designed for consistent basis functions, the unified algorithm is mainly based on Algorithm 1 to ensure the compatibility of the unified algorithm with the higher-order B-spline basis functions. Furthermore, the proposed approach employs the consistent mass matrix. The unified algorithm is presented in Algorithm 4.

Both MPM and OTM provide motivation for the proposed unified method. On the one hand, the addition of the advected nodal points to update the basis functions is supposed to stabilize the computation when BSMPM is combined with a consistent mass matrix. Consistent mass matrices frequently cause stability issues in MPM [27] and BSMPM inherits these issues. For this reason, MPM and BSMPM are generally used with a lumped mass matrix. While mass lumping has little influence on the solution quality of lower order methods, its $\mathcal{O}(h^2)$ approximation of the consistent mass matrix [28] can significantly influence the spatial convergence of higher-order methods such as BSMPM. Moreover, previous studies demonstrate that methods similar to the unified approach can be used successfully for complex simulations. In fact, a method closely related to the unified approach, the so-called moving-mesh MPM, has been applied to model the biological mechanics of cells [163] and the texture evolution in polychrystalline nickel [164].

On the other hand, within the OTM framework, the use of B-spline basis functions is expected to significantly reduce the computational costs. In contrast to maxent basis functions, B-spline basis functions do not require the adoption of iterative methods and have a purely analytical definition. It has been pointed out by Cyron et al. [147] that maxent and higher-order B-spline basis functions have many common properties. For example, they are both smooth, non-negative, have compact support, and satisfy the partition of unity property. Therefore, the use of B-spline basis functions provides a viable alternative to the massively parallel implementation of OTM (pOTM) [165]. In addition, B-spline basis functions do not require the use of search algorithms, frequently added to the standard OTM scheme for stabilization [146, 151]. Finally, the adoption of higher-order B-spline basis functions can lead to higher-order spatial convergence.

We remark that in contrast to the OTM scheme, the proposed unified approach can not be viewed as a meshless method. However, the study from Chater 3 shows the potential of BSMPM on arbitrary grids. The extension of the BSMPM to arbitrary grids in combination with an efficient remeshing technique (e.g., the remeshing strategy for large deformations proposed by Erhart et al. [166]) might bring the unified approach closer to the meshless algorithms. In this chapter, the examples are restricted to relatively simple problems to study the basic properties of the considered methods. Thus, further research is required to evaluate the effect of mesh distortion on the proposed unified approach.

4.5.1. ALGORITHM

Input: Coordinates at degrees of freedom $\mathbf{x}_i^0$, material-point coordinates $\mathbf{x}_p^0$, velocities $\mathbf{v}_p^0$, volumes V_p^0, densities ρ_p^0, masses m_p, deformation gradients $\mathbf{F}_p^0$, body forces $\mathbf{b}_p$

1 Set $s = 0$

2 **while** $s < N$ **do**

3   Compute basis functions $\phi_i^s\left(\mathbf{x}_p^s\right)$ and gradients $\nabla\phi_i^s\left(\mathbf{x}_p^s\right)$ from advected nodal set $\{\mathbf{x}_i^s\}$ and advected material-point set $\{\mathbf{x}_p^s\}$

4   Compute mass matrix $\mathbf{M}^s$, linear momentum vector $\bar{\mathbf{q}}_k^s$, and force vector $\bar{\mathbf{f}}_k^s$:

$$M_{ij}^s = \sum_{p=1}^{\mathcal{M}} \phi_i^s\left(\mathbf{x}_p^s\right) m_p \phi_j^s\left(\mathbf{x}_p^s\right) \tag{4.81}$$

$$q_{k,i}^s = \sum_{p=1}^{\mathcal{M}} \phi_i^s\left(\mathbf{x}_p^s\right) m_p v_{k,p}^s \tag{4.82}$$

$$f_{k,i}^s = \sum_{p=1}^{\mathcal{M}} \left(\sum_{l=1}^{3} \frac{\partial \phi_i^s}{\partial x_l}\left(\mathbf{x}_p^s\right) \sigma_{lk,p}^s + \phi_i^s\left(\mathbf{x}_p^s\right) \rho_p^s b_{k,p} \right) V_p^s \tag{4.83}$$

5   Compute accelerations at degrees of freedom:

$$\bar{\mathbf{a}}_k^s = \left(\mathbf{M}^s\right)^{-1} \bar{\mathbf{f}}_k^s \tag{4.84}$$

6   Compute incremental displacement and updated coordinates for degrees of freedom:

$$\Delta\bar{\mathbf{x}}_k^{s+1} = \left(t^{s+1} - t^s\right)\left(\mathbf{M}^s\right)^{-1} \left(\bar{\mathbf{q}}_k^s + \frac{t^{s+1} - t^{s-1}}{2} \bar{\mathbf{f}}_k^s \right) \tag{4.85}$$

$$\bar{\mathbf{x}}_k^{s+1} = \bar{\mathbf{x}}_k^s + \Delta\bar{\mathbf{x}}_k^{s+1} \tag{4.86}$$

7   Update material-point coordinates:

$$\mathbf{x}_p^{s+1} = \mathbf{x}_p^s + \sum_{i=1}^{\mathcal{N}} \phi_i^s\left(\mathbf{x}_p^s\right) \Delta\mathbf{x}_i^{s+1} \tag{4.87}$$

4

8 | Update material-point velocities:

$$\mathbf{v}_p^{s+1} = \mathbf{v}_p^s + \left(t^{s+1} - t^s\right) \sum_{i=1}^{\mathcal{N}} \phi_i^s\left(\mathbf{x}_p^s\right) \mathbf{a}_i^s \tag{4.88}$$

9 | Update material-point deformation gradients:

$$\mathbf{F}_p^{s+1} = \left(\mathbf{I} + \left(t^{s+1} - t^s\right) \nabla \mathbf{v}_p^{s+1}\right) \mathbf{F}_p^s \tag{4.89}$$

10 | Update material-point volumes:

$$V_p^{s+1} = \det\left(\mathbf{F}_p^{s+1}\right) V_p^s \tag{4.90}$$

11 | Update material-point densities:

$$\rho_p^{s+1} = \frac{m_p}{V_p^{s+1}} \tag{4.91}$$

12 | Compute material-point stresses $\boldsymbol{\sigma}_p^{s+1}$ from $\mathbf{F}_p^{s+1}$ using the constitutive equation (e.g., Equation 2.13 or 2.14)

13 | Set $s = s + 1$

end

Algorithm 4: Unified approach.

4.6. NUMERICAL RESULTS

In this section, three benchmarks are considered to illustrate the performance of the discussed methods. The one-dimensional benchmarks describe the vibration of a bar, but have fundamentally different motion triggers and boundary conditions. In the first benchmark, where both ends of the bar are fixed, the domain contains only filled elements allowing for a straightforward implementation and analysis. The second benchmark, where a traction force is acting at one of the boundaries, contains multiple empty cells throughout the simulation, thereby serving as a representative example for the stability analysis. The last benchmark is two-dimensional, it further extends the numerical analysis of the considered algorithms.

The results are provided for the USL version of the MPM scheme (Algorithm 1) and OTM algorithms, as well as the proposed unified approach. For MPM, piecewise-linear, second-order B-spline, and maxent basis functions are employed. The OTM algorithm is used only with consistent basis functions (i.e., piecewise-linear and maxent basis functions). For the one-dimensional benchmarks, the factor $d_{\max}$ is set to 2.0 to compute the maxent basis functions, while for the two-dimensional problem, its value depends on the considered algorithm.

4.6.1. BAR WITH FIXED ENDS

This benchmark has been presented in Chapter 3, which also provides its analytical solution in terms of displacement, velocity, and stress. Here, the length of the bar is set to

1 m, Young's modulus is set to 4 kPa, while the initial density and amplitude of the velocity v_0 are equal to 1 kg/m^3 and 0.6 m/s, respectively. The total simulation time is set to 0.001 s, while the time-step size is equal to 10^{-5} s. This relatively small time step is required to minimize the contribution of the temporal error to the total one. Moreover, the number of nodes varies between 8 and 512, while the number of material points per element remains equal to 12. The analytical solution in terms of displacement, velocity, and stress can be found in Chapter 3. For the convergence analysis, the Root-Mean-Square (RMS) error in the displacement is computed. RMS error is defined as follows:

$$\sqrt{\frac{1}{\mathcal{M}} \sum_{p=1}^{\mathcal{M}} (u(x_p^0, T) - u_p)}, \tag{4.92}$$

where $u(x_p^0, T)$ and u_p are, respectively, the analytical and numerical solutions at position x_p^0 at time T.
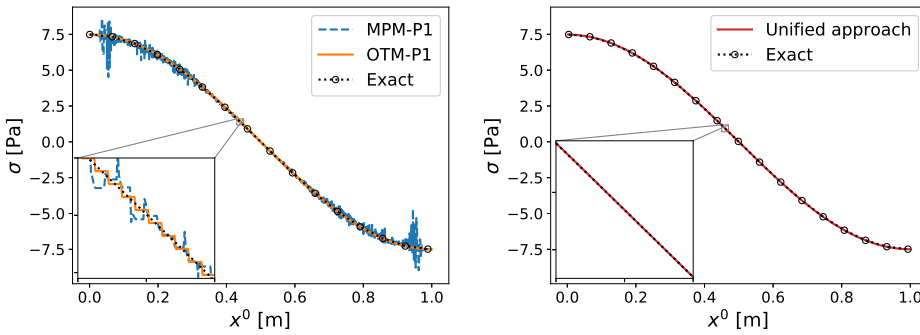

Figure 4.6: Comparison of the results obtained with MPM and the OTM method with piecewise-linear basis functions (left), and the unified approach (right) to the analytical solution. For MPM and the OTM method, the number of nodes is equal to 512, while for the unified approach only 32 nodes were used.

The left part of Figure 4.6 depicts the final stress profiles obtained using MPM and the OTM method with piecewise-linear basis functions. Grid crossing causes severe oscillations in the MPM stress profile. The calculation of the basis functions with the advected nodal coordinates in the OTM method prevents these inaccuracies, significantly improving the results. Although the OTM-P1 method avoids grid crossing errors, it provides only a piecewise-constant approximation of the stress profile due to the gradients of the P1 basis functions. The right part of Figure 4.6 illustrates the performance of the unified approach. The use of second-order B-spline basis functions prevents the grid crossing errors and improves the accuracy of the solution. The results obtained with maxent basis functions are similar to those computed with the unified approach. To avoid repetition, these results are not shown.

Table 4.1 provides the results in terms of computational time obtained with the in-house implementation of the considered methods in Python. We remark that the maxent computations were performed without a search algorithm. The table shows the simulation time of all considered methods for 32 and 512 nodes. Based on this information, it

can be concluded that the simulations with the B-spline basis functions are considerably cheaper compared to those with the maxent basis functions. More precisely, replacing maxent by B-spline basis functions reduces the computational time by at least a factor of 6 for MPM and by at least a factor of 8 for the OTM algorithm.

Table 4.1: Computational time required for the considered methods normalized with respect to the MPM-P1 computational time with the corresponding number of nodes.

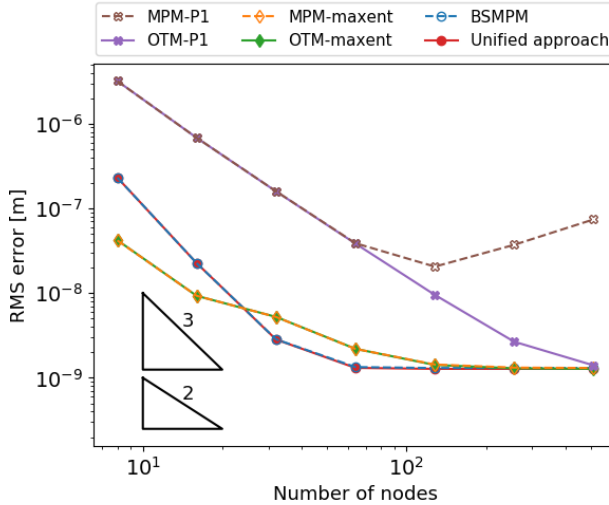| Method	Time [s]	
	32 nodes	512 nodes
OTM - P1	1.12	1.04
MPM - maxent	16.22	6.96
OTM - maxent	26.04	9.07
BSMPM	1.00	1.11
Unified approach	1.23	1.15

Figure 4.7: Convergence behavior of the considered methods.

In addition, Figure 4.7 illustrates the spatial convergence behavior of the considered methods at the end of the simulation. When piecewise-linear basis functions are used, both MPM and the OTM method demonstrate second-order convergence for relatively coarse grids. However, for fine meshes, the methods behave differently. In fact, MPM suffers from grid-crossing errors that result in a loss of the convergence. The OTM-P1 method preserves the second-order convergence until the final refinement, where only first-order convergence is achieved. A sudden decrease of the convergence rate is also observed in the computations when maxent and B-spline basis functions are employed within both MPM and OTM schemes. For this reason, it may be assumed that the loss of the convergence order is unrelated to the choice of the basis functions. Inaccurate

numerical integration, time-integration errors, round-off errors, or a combination of the above can contribute to the reduction of the convergence rate [25, 28, 84]. Furthermore, Figure 4.7 shows that maxent basis functions lead to significantly lower errors than the piecewise-linear basis functions for both MPM and OTM schemes. With maxent basis functions, the convergence of MPM and the OTM method varies between linear and quadratic. It should be noted that the accuracy of MPM and the OTM method with maxent basis functions can be further improved by adapting more advanced implementations [145, 167–169]. The use of B-spline basis functions leads to similar results for BSMPM and the unified approach. These methods have third-order convergence untill the limiting value is reached.

4.6.2. BAR WITH DYNAMIC TRACTION BOUNDARY CONDITIONS

This benchmark describes the motion of a neo-Hookean bar with one free end. The bar is fixed at $x^0 = 0$ and subjected to a traction force at the free end $x^0 = l$. The forcing function is equal to $\tau\left(x^0, t\right) = \delta\left(x^0 - l\right)\sigma\left(x^0, t\right)$. Defining $\omega = \pi/l$, the stress is given by

$$\sigma\left(x^0, t\right) = \begin{cases} 0 & \text{for } t \in \left[0, l - x^0\right), \\ \sin\left(\omega\left(t + x^0\right)\right) & \text{for } t \in \left[l - x^0, l + x^0\right), \\ \sin\left(\omega\left(t + x^0\right)\right) + \sin(\omega(t - x^0)) & \text{for } t \in \left[l + x^0, 3l - x^0\right), \\ \sin\left(\omega\left(t - x^0\right)\right) & \text{for } t \in \left[3l - x^0, 3l + x^0\right), \\ 0 & \text{for } t \in \left[3l + x^0, 4l\right]. \end{cases} \tag{4.93}$$

The initial length of the bar is set to 1 m, the density is equal to 100 kg/m³, and Young's modulus is equal to 100 Pa. The length of the computational domain is set to 1.25 m. A more detailed description that includes an analytical solution for displacement is provided by Steffen et al. [31]. To illustrate the stress profile obtained with different methods, the material domain is discretized by 68 nodes, which results in 85 nodes for the complete domain. The material-domain discretization is sufficient for the unified and OTM methods due to their updated Lagrangian nature, whereas MPM requires the discretization of the complete domain. Each active element initially contains 4 particles. The computational time is set to 0.4 s and the time-step size is equal to 10^{-4} s.

Figure 4.8 depicts the obtained results. It shows that maxent and B-spline basis functions eliminate the grid crossing error in MPM. However, MPM-maxent and BSMPM do not follow the analytical solution at the right edge of the bar. Within BSMPM, these inaccuracies can be significantly reduced by increasing the initial number of particles per elements. This suggests that the errors are caused by insufficient accuracy of the numerical integration in MPM. Thus, advanced numerical integration techniques (e.g., Taylor least squares, which is described in Chapter 5) may improve the BSMPM solution at the boundary. The inaccuracies within MPM-maxent have a different origin. They are most probably caused by the incomplete set of maxent basis functions, which arise from the presence of inactive elements throughout an MPM simulation. Figure 4.8 also shows that the OTM-P1, OTM-maxent, and unified methods provide significantly more accurate solutions than their MPM equivalents.

Similarly to the benchmark discussed in Section 4.6.1, the use of B-spline basis functions instead of maxent basis functions considerably decreases the computational time
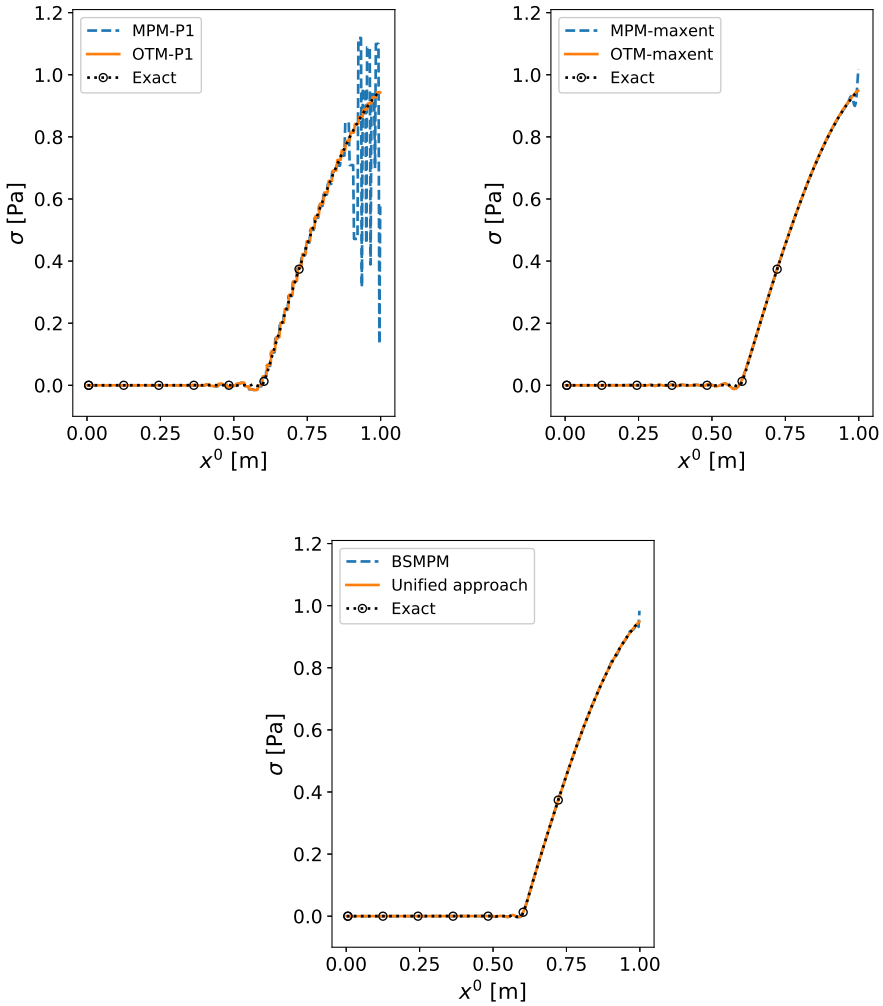
4

Figure 4.8: Comparison of the considered methods to the analytical solution. The material domain is discretized by 68 nodes.

for both MPM and OTM methods. In fact, the unified approach and BSMPM computations are approximately 10 times faster than the OTM and MPM computations with maxent basis functions (without a search algorithm) with the settings used for Figure 4.8.

Furthermore, the unified approach and BSMPM have the lowest RMS error and highest convergence rates compared to the other methods. This is illustrated in Figure 4.9. To minimize quadrature and time integration errors, this figure is obtained placing 12 particles per cell at the beginning of the simulation and reducing the computational time to 0.1 s. In general, the obtained convergence orders of the considered algorithms are

slightly lower than expected. This can be related to the discontinuities in the solution for the stress field.

The main advantage of the unified method over BSMPM arises from its stability properties. When a material point enters an empty element, BSMPM inherits stability issues from MPM. For instance, changing the total number of nodes to 81 results in a termination after 0.3 s. This issue in MPM is discussed in detail by, for example, Kafaji [37] and requires the use of the MUSL algorithm with a lumped mass matrix to circumvent the breakdown.
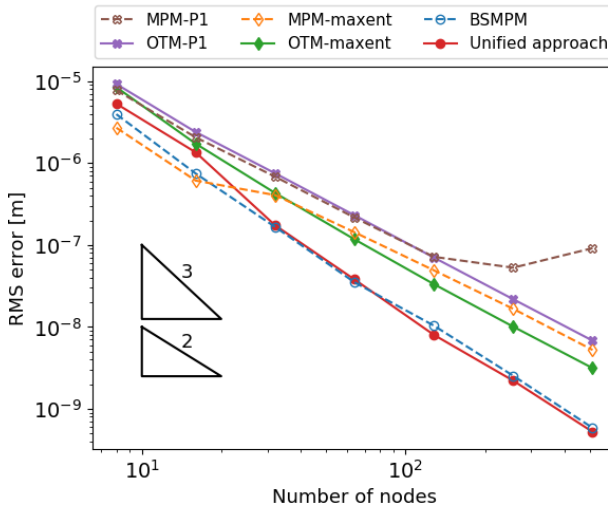

Figure 4.9: Convergence behavior of the considered methods. The number of nodes, in this case, refers to the number of nodes used to discretize the material domain.

4.6.3. PLATE UNDERGOING AXIS-ALIGNED DISPLACEMENT

The final benchmark describes a two-dimensional neo-Hookean plate from Chapter 2. Here, the maximum amplitude of the displacement B is set to 0.005 m, Young's modulus E is equal to 10^7 Pa, and initial mass density ρ^0 is set to $1 \cdot 10^7$ Pa. Furthermore, Poisson's ratio is equal to 0.3. The domain is discretized by 33 nodes in each direction and each element contains initially 16 particles. The computational time is set to $3.5 \cdot 10^{-3}$ s and the time-step size equals 10^{-4} s. The tensor product of the one-dimensional basis functions is adopted for discretization.

Figure 4.10 shows the normal stress in x_1-direction σ_{11} along the plate for material points with $x_2^0 \approx 0.07$ m. Employing piecewise-linear basis functions within both MPM and the OTM method leads only to a piecewise-constant approximation of the stress field. Due to grid-crossing errors, the stress profile obtained using MPM with piecewise-linear basis functions deviates significantly from the analytical solution. The simulations performed with the maxent basis functions show considerably more accurate stress approximations for both methods. To obtain these results with the MPM algorithm the user-defined factor $d_{\max}$ is set to 3.0, while for the OTM method, $d_{\max} = 2.0$ is taken. The
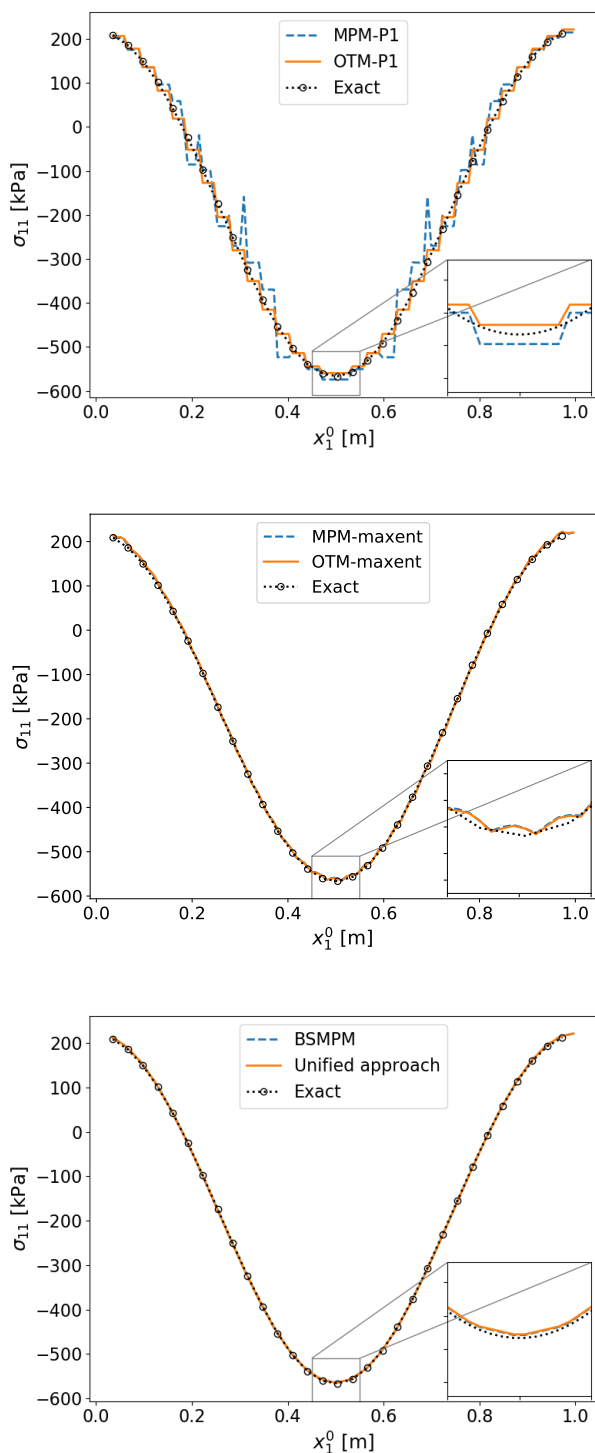
4

Figure 4.10: Comparison of the considered methods to the analytical solution. The material domain is discretized by 33 nodes in each direction. The figures were obtained in collaboration with Roel Tielen.

unified approach and BSMPM lead to an even smoother stress profiles which are in close agreement with the analytical solution.

4.7. CONCLUSIONS

The first part of this study provides a comparison between the MPM and OTM schemes. While the methods were derived in fundamentally different manners, the resulting algorithms are closely related. In fact, assuming a constant time step, the validity of the backward Euler scheme for material-point displacement in MPM, as well as the linear completeness of the MPM basis functions and their ability to translate nodal velocities into material-point velocities, the only difference between the algorithms emerges from the update of the basis functions. However, this difference is fundamental. Since MPM uses initial nodal positions in the basis-function update, it is viewed as a combination of Lagrangian and Eulerian approaches. At the same time, the OTM method is a fully updated Lagrangian method. Moreover, MPM is typically used with piecewise-linear basis functions, whereas the OTM method generally employs maximum-entropy basis functions.

In the second part of the study, a unified approach is proposed. This approach combines BSMPM and OTM methods. Similarly to BSMPM, the proposed approach uses B-spline basis functions. However, it computes the basis functions based not only on the updated material-point positions, but also on the advected degrees-of-freedom set. The obtained numerical results demonstrate that the proposed method preserves the convergence properties of BSMPM and remains stable when a consistent mass matrix is adopted. Furthermore, the unified approach does not contain user-defined parameters and is significantly cheaper and more stable than the standard OTM computation.

III

TAYLOR LEAST-SQUARES FUNCTION RECONSTRUCTION AND ITS APPLICATION TO THE MATERIAL-POINT METHOD

5

CONSERVATIVE TAYLOR LEAST-SQUARES RECONSTRUCTION WITH APPLICATION TO MATERIAL-POINT METHODS

This chapter introduces a novel technique to reconstruct functions from scattered data, to which we refer as the Taylor least-squares (TLS) function-reconstruction technique. The main feature of the TLS technique is that it preserves the a priori known integral values of the reconstructed functions. In this chapter, the TLS reconstruction is presented for one-dimensional problems and applied to the material-point method (MPM) and related methods in order to reduce the spatial errors within them. The chapter is structured as follows. Section 5.1 gives a short introduction to the family of least-squares methods. Section 5.2 describes the building blocks of the TLS approximation and provides an example of its application. Section 5.3 motivates the use of the TLS technique within MPM and related methods, outlines how it can be applied within an MPM algorithm, and gives a mathematical analysis of its conservation properties. Section 5.4 discusses the numerical results obtained for vibrating bar and column compaction problems. Finally, Section 5.5 provides the conclusions.

5.1. INTRODUCTION

As was pointed out in Chapter 3, function reconstruction from scattered data can be achieved by interpolating or approximating the known values. While Chapter 3 focuses on an interpolation technique, this chapter addresses function reconstruction by means of approximation. One of the most widespread approaches to find the best polynomial approximation to scattered data is the least-squares method [171]. Assuming a linear relation for the provided data, the least-squares method finds the line that minimizes the sum of squared distances (deviations) from itself to each data point (see Section 5.2.1 for a detailed mathematical description). A generalization of the least-squares approach can be achieved by assigning unequal weights to the deviations and is known as the weighted least-squares (WLS) method. For smoothing and interpolating data, Lancaster and Salkauskas [172] introduced the so-called moving least-squares (MLS) method, which starts with a WLS formulation for an arbitrary fixed point inside the considered domain, and then moves this point over the domain while computing and evaluating a WLS fit at each point individually. A detailed description of the MLS and WLS methods is provided by, for example, Nealen [127]. The least-squares methods have been applied to problems in a variety of fields including statistical regression (e.g., [173, 174]), computer-aided geometric design and computer graphics (e.g., [175, 176]), image processing (e.g., [177]), computational aerodynamics (e.g., [178]), and meshfree methods (e.g., [26, 179, 180]). It has also been shown that the approximations obtained with the least-squares techniques are highly accurate (e.g., [179, 181]). However, little information is available on the approximation methods that allow for the preservation of an a priory known integral value of the function.

In this chapter, we propose a novel reconstruction technique, called Taylor least squares (TLS), which reconstructs functions from scattered data, while preserving their integral values. TLS combines the least-squares method with the Taylor basis functions [182] to locally approximate quantities of interest. The technique is applied within MPM and related methods to reduce spatial errors arising from the direct mapping of material-point data to the background grid.

5.2. TAYLOR LEAST-SQUARES FUNCTION RECONSTRUCTION

5.2.1. LEAST-SQUARES APPROXIMATION

Given a set of $\mathcal{M}$ distinct one-dimensional data points, $\{x_p\}_{p=1}^{\mathcal{M}}$, and the data values of these points, $\{f(x_p)\}_{p=1}^{\mathcal{M}}$, we assume that $f \in F$, where F is a normed function space on $\mathbb{R}$, and $P = \text{span}\{\psi_i\}_{i=1}^{n_b} \subset F$ is a set of n_b basis functions. The least-squares [171] approximation at a point $x \in \mathbb{R}$ is the value $w^* \in P$ that minimizes, among all $w \in P$, the least-squares error:

$$E = \sum_{p=1}^{\mathcal{M}} \left(w(x_p) - f(x_p) \right)^2.$$

Using the basis-function vector, $\boldsymbol{\psi}(x) = \left[\psi_1(x), \psi_2(x), \ldots, \psi_{n_b}(x)\right]^{\mathsf{T}}$, and the vector of unknown coefficients, $\mathbf{a} = \left[a_1, a_2, \ldots, a_{n_b}\right]^{\mathsf{T}}$, the least-squares approximation can be
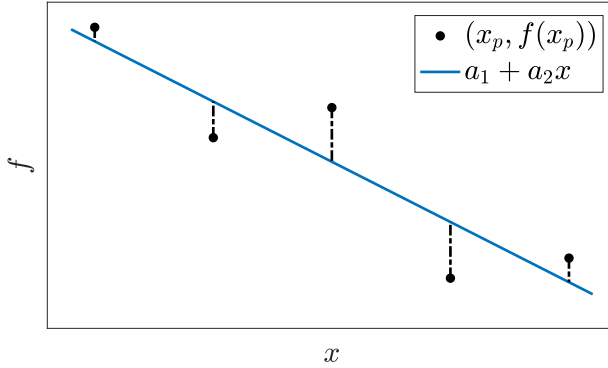
Figure 5.1: Example of least-squares approximation with a standard polynomial basis.

written as

$$w^*(x) = \sum_{i=1}^{n_b} a_i \psi_i(x) = \boldsymbol{\psi}^\top(x)\mathbf{a}.$$

Note that $\boldsymbol{\psi}^\top$ is a row vector, while $\mathbf{a}$ is a column vector. An example of the least-squares approximation with a standard polynomial basis is provided in Figure 5.1. In order to compute the coefficient vector, $\frac{\partial E}{\partial a_i}$ is set to zero for $i = 1, 2, \ldots, n_b$, leading to the normal equations:

$$\sum_{p=1}^{\mathcal{M}} \boldsymbol{\psi}\left(x_p\right)\left(\boldsymbol{\psi}^\top\left(x_p\right)\mathbf{a} - f\left(x_p\right)\right) = \mathbf{0}.$$

From that, we obtain the following expression for the unknown coefficients:

$$\mathbf{a} = \left(\sum_{p=1}^{\mathcal{M}} \boldsymbol{\psi}\left(x_p\right)\boldsymbol{\psi}^\top\left(x_p\right)\right)^{-1} \sum_{p=1}^{\mathcal{M}} \boldsymbol{\psi}\left(x_p\right) f\left(x_p\right). \tag{5.1}$$

Defining the matrices D and B and the function vector U as

$$D = \sum_{p=1}^{\mathcal{M}} \boldsymbol{\psi}\left(x_p\right)\boldsymbol{\psi}^\top\left(x_p\right), \tag{5.2}$$

$$B = \left[\boldsymbol{\psi}(x_1),\ \boldsymbol{\psi}(x_2),\ \ldots,\ \boldsymbol{\psi}(x_{\mathcal{M}})\right], \tag{5.3}$$

$$U = \left[f(x_1),\ f(x_2),\ \ldots,\ f(x_{\mathcal{M}})\right]^\top, \tag{5.4}$$

the least-squares solution is given by

$$w^*(x) = \boldsymbol{\psi}^\top(x)D^{-1}BU. \tag{5.5}$$

It should be noted that the basis for P is not yet specified.

5.2.2. TAYLOR BASIS FUNCTIONS

A viable choice for the basis for P that leads to an overall conservative reconstruction scheme is the local Taylor basis functions [182]. To define these basis functions we introduce the concept of the volume average of a function f over the cell e:

$$\overline{f} = \frac{1}{|\Omega_e|} \int_{\Omega_e} f \, d\Omega_e, \tag{5.6}$$

where $|\Omega_e|$ is the volume of cell e. In one dimension, $\Omega_e = [x_{\min}, x_{\max}]$ with $x_{\max} > x_{\min}$, and $|\Omega_e| = x_{\max} - x_{\min}$.

The first three Taylor basis functions are then given by

$$\psi_1 = 1,$$
$$\psi_2 = \frac{x - x_c}{\Delta x},$$
$$\psi_3 = \frac{(x - x_c)^2}{2\Delta x^2} - \overline{\frac{(x - x_c)^2}{2\Delta x^2}}, \text{ etc.}$$

Here, $x_c = \frac{x_{\max} + x_{\min}}{2}$ is the centroid x_c of the cell e, and $\Delta x = \frac{x_{\max} - x_{\min}}{2}$ (see Figure 5.2). The first three Taylor basis functions are illustrated in Figure 5.3. An important aspect
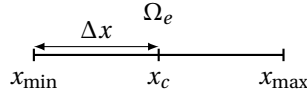

Figure 5.2: Illustration of x_c and Δx for one dimensional elements.

of the Taylor basis that will ensure the conservation property of the reconstruction technique is [182]:

$$\int_{\Omega_e} \psi_i \, d\Omega_e = \begin{cases} |\Omega_e| & \text{if } i = 1, \\ 0 & \text{if } i \neq 1. \end{cases} \tag{5.7}$$
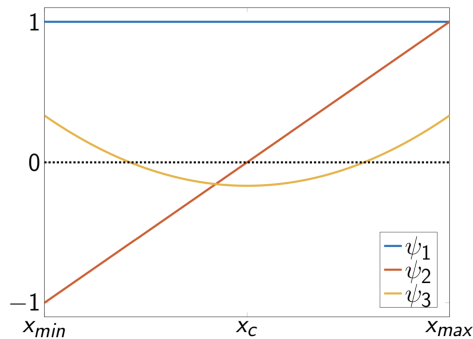

Figure 5.3: First three Taylor basis functions.

5.2.3. EXAMPLES OF TAYLOR LEAST-SQUARES RECONSTRUCTION

The TLS approach uses local Taylor basis functions for the least-squares approximation of a function f:

$$f(x) \approx \hat{f}(x) = \sum_{i=1}^{n_b} a_i \psi_i(x).$$

Suppose that $\int_{\Omega_e} f(x)\, d\Omega_e = c$ with $c \in \mathbb{R}$ should be conserved by the reconstruction. Then, using Equation (5.7), we obtain:

$$\int_{\Omega_e} \hat{f}(x)\, d\Omega_e = \int_{\Omega_e} \sum_{i=1}^{n_b} a_i \psi_i(x)\, d\Omega_e = \sum_{i=1}^{n_b} a_i \int_{\Omega_e} \psi_i(x)\, d\Omega_e = a_1 |\Omega_e|. \tag{5.8}$$

Therefore, the reconstruction procedure preserves the integral quantity if we set

$$a_1 := \frac{c}{|\Omega_e|}. \tag{5.9}$$

It should be noted that Equation (5.9) can be enforced explicitly.

We illustrate this property by reconstructing $f(x) = \sin(x) + 2$ on $[0, 4\pi]$. In this case, the integral is equal to 8π. The domain is divided into four elements of size π and contains 11 data points. Two data points are located at the boundaries of the first element, (i.e., 0 and π). In $[2\pi, 3\pi]$, the data points are distributed uniformly in the interior of the domain. The remaining data points have random positions creating different types of data distribution within an element. It should be noted that the data point at the right boundary of the first element is used for both the first and second elements.
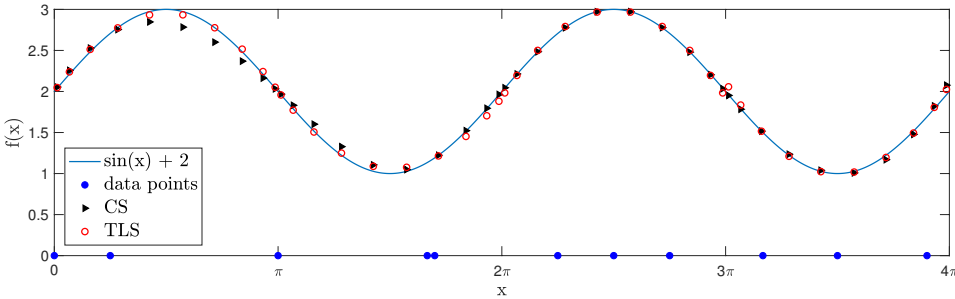

Figure 5.4: Cubic-spline (CS) and TLS reconstructions of $f(x) = \sin(x) + 2$ on $[0, 4\pi]$ for different types of data point distribution within an element.

The TLS approximation is obtained using three Taylor basis functions. We compare its performance with that of the cubic-spline reconstruction in terms of the Root-Mean-Square (RMS) error for function f and the relative error for the integral of f. The RMS error is computed using 100 Gauss points per element, while for the numerical integration, the reconstructed function is evaluated only at two Gauss points within each element. Figure 5.4 visualizes the data point distribution, and the cubic-spline and TLS reconstructions of f for 10 Gauss points per element, whereas Table 5.1 provides the corresponding errors.

Table 5.1: Errors of the TLS and cubic-spline reconstruction for $f(x) = \sin(x) + 2$ and its integral on $[0, 4\pi]$ with four elements and eleven data points.

Error	Cubic-splines	TLS
RMS error for function	$6.0867 \cdot 10^{-2}$	$3.9967 \cdot 10^{-2}$
Relative error for integral	$4.2759 \cdot 10^{-2}$	$2.7903 \cdot 10^{-15}$

Table 5.1 shows that for the considered example, the TLS technique outperforms the cubic-spline reconstruction when the conservation and accuracy properties are considered. In fact, the TLS approach preserves the integral up to machine precision. However, Figure 5.4 shows that the performance of the TLS technique depends on the distribution of the data points within each element. More precisely, the TLS approximation within the interval $[2\pi, 3\pi]$, where the data points are distributed uniformly, is more accurate than, for instance, in the interval $[0, 1\pi]$.

In some rare cases, data distribution can locally decrease the quality of the TLS approximation, but has little influence on the cubic-spline interpolation. An example is provided in Figure 5.5, where $[\pi, 2\pi]$ contains only two data points located at π and $5\pi/3$. This particular data distribution leads to a linear dependence between the columns of matrix D from Equation (5.5) and, hence, distorts the TLS approximation within this interval. It is possible to use the condition number of D to detect the data distributions that decrease the accuracy of the TLS technique. In addition, the quality of the TLS technique in such situations can be improved easily.
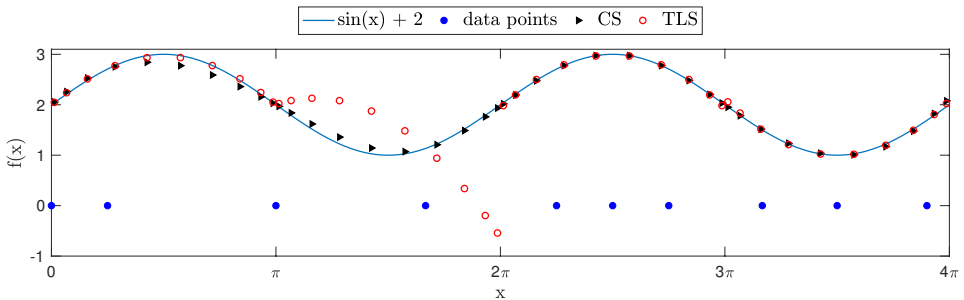

Figure 5.5: Comparison of the cubic-spline and TLS reconstructions of $f(x) = \sin(x) + 2$ on $[0, 4\pi]$ for a challenging local data distribution.

First of all, the singularity of D can be prevented by reducing the number of basis functions used for the reconstruction on $[\pi, 2\pi]$. Although this strategy will preserve the conservative properties of the TLS technique and can be implemented in a straightforward manner, it will lower the accuracy of the method. Therefore, we suggest an alternative approache that uses the information from the neighboring intervals to maintain the high quality of the reconstruction scheme. On the one hand, it is possible to evaluate the TLS approximation of f associated with $[2\pi, 3\pi]$ at 2π and add the obtained value to the set of data points upon which the TLS approximation is based on $[\pi, 2\pi]$. However, this so-called *virtual* data point at 2π should be excluded from the computation of a_1 in

Equation (5.9). Thus, virtual points do not influence the conservative properties of the technique. Figure 5.6 illustrates the improved approximation in this case. On the other hand, the data points from the neighboring intervals can also be used directly. For the considered example, the closest neighboring point to interval $[\pi, 2\pi]$ is located within interval $[2\pi, 3\pi]$. This approach is slightly less accurate than the one with virtual points, but the approximation remains conservative as long as the computation of a_1 preserves the values within each element. The TLS reconstruction that employs neighboring data points is depicted in Figure 5.7.
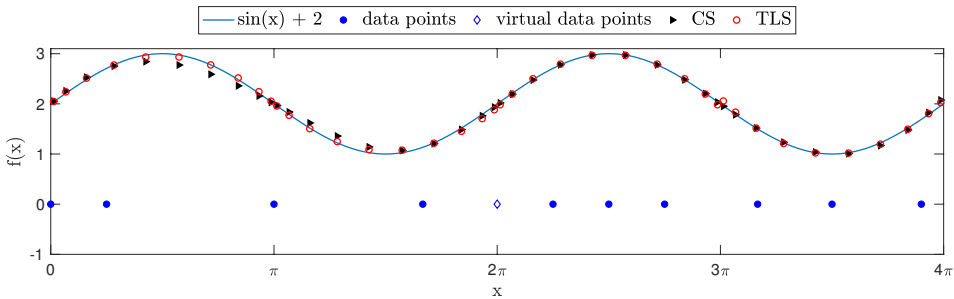

Figure 5.6: Comparison of the cubic-spline and TLS reconstructions of $f(x) = \sin(x) + 2$ on $[0, 4\pi]$ with a virtual data point at 2π.
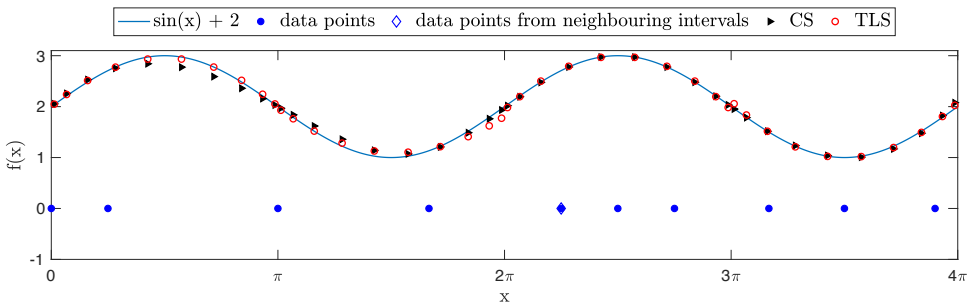
Figure 5.7: Comparison of the cubic-spline and TLS reconstructions of $f(x) = sin(x) + 2$ on $[0, 4\pi]$ with a neighbouring data point.

5.3. APPLICATION OF TAYLOR LEAST-SQUARES TECHNIQUE TO THE MATERIAL-POINT METHODS

Although modified mapping techniques generally reduce the spatial errors that are caused by the direct mapping of material-point data to the background grid, the standard reconstruction techniques, such as spline interpolation presented in Chapter 3, might lead to the loss of physical properties of the material-point methods. MPM, as well as the dual domain material-point method (DDMPM), and B-spline MPM (BSMPM) preserve the

total mass $\mathbb{M}$ and linear momentum $\mathbb{Q}$ of the system. This implies that the methods conserve the mass and momentum while mapping the information from material-points to the background grid. To show this, the row-lumped mass from Equation 2.22 is written in the following way:

$$M_{ij}^{\mathrm{L}\,s} = \begin{cases} m_i \;\text{ with }\; m_i = \sum_{p=1}^{\mathcal{M}} \phi_i^0\left(\mathbf{x}_p^s\right) m_p, & \text{for } i = j, \\ 0, & \text{for } i \neq j. \end{cases} \tag{5.10}$$

In the one-dimensional case, Equation (5.10), Equation (2.49) from Algorithm 2 and the partition of unity property of the basis functions yield:

$$\mathbb{M} = \sum_{i=1}^{\mathcal{N}} m_i = \sum_{i=1}^{\mathcal{N}} \sum_{p=1}^{\mathcal{M}} m_p \phi_i\left(x_p\right) = \sum_{p=1}^{\mathcal{M}} m_p \sum_{i=1}^{\mathcal{N}} \phi_i\left(x_p\right) = \sum_{p=1}^{\mathcal{M}} m_p,$$

$$\mathbb{Q} = \sum_{i=1}^{\mathcal{N}} m_i v_i = \sum_{i=1}^{\mathcal{N}} m_i \frac{1}{m_i} \sum_{p=1}^{\mathcal{M}} m_p \phi_i\left(x_p\right) v_p = \sum_{p=1}^{\mathcal{M}} m_p v_p \sum_{i=1}^{\mathcal{N}} \phi_i\left(x_p\right) = \sum_{p=1}^{\mathcal{M}} m_p v_p.$$

The time index is dropped to simplify the notation.

Integration of the TLS reconstruction within MPM allows for the preservation of these conservative properties of MPM, while improving the accuracy of the method. The idea behind the use of the TLS reconstruction within MPM is similar to that described in Section 3.2.1:

- scattered material-point information is used to reconstruct certain quantities of interest over the material domain,

- the obtained approximation is evaluated at the integration points (such as Gauss points),

- the resulting values are projected to the background grid by means of exact numerical integration.

In contrast to the spline interpolation, the TLS technique reconstructs functions locally, within each element. This implies that the reconstructed function may contain discontinuities at the element boundaries (see Figure 5.8). Therefore, numerical integration is performed separately within each active element as in any element-based discretization.

While the TLS reconstruction can be used autonomously within the MPM algorithm, it can also be combined with techniques like the convected particle-domain interpolation (CPDI) method or BSMPM in order to further reduce the spatial errors. To avoid stability issues, TLS is applied within the MUSL algorithm, presented in Algorithm 2, with a lumped mass matrix. Since previous studies have indicated that DDMPM and BSMPM are viable alternatives not only for MPM, but also for the generalized interpolation material-point (GIMP) method and the CPDI method [21, 30, 99], we have applied the TLS reconstruction technique within MPM, DDMPM, and BSMPM. The methods are tested on two one-dimensional benchmarks describing the deformation of a vibrating bar and column compaction under slowly increasing body force, respectively. The
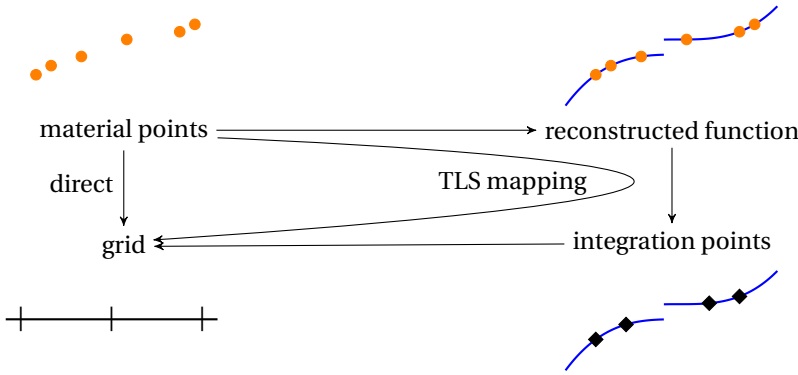
Figure 5.8: Schematic representation of the TLS mapping.

conservation property of the TLS reconstruction technique is verified by computing the total mass and momentum before and after projecting the particle information to the background grid. The accuracy of the material-point methods with TLS approximation is investigated either qualitatively or based on the spatial errors and convergence rates. In addition, the obtained results are compared to those computed with the cubic-spline reconstruction technique.

5.3.1. MAPPING OF PARTICLE DATA

When the TLS reconstruction is considered as part of MPM or a related method, particles serve as data points. To conserve the integral of a certain quantity within each element, the coefficient of the first basis function is specified according to Equation (5.9). The remaining coefficients are calculated from Equation (5.1) excluding ψ_1, thereby leaving the integral value unchanged. When the conservation of the reconstructed quantity is not required, a standard least-square approach is followed. This implies that all coefficients are treated as unknowns and their values are obtained from Equation (5.1).

A TLS reconstruction is applied to replace the MPM-integration for the computation of internal forces:

$$f_{\text{int},i} = \sum_{p=1}^{\mathscr{M}} \frac{\partial \phi_i^0}{\partial x} (x_p) \sigma_p V_p, \tag{5.11}$$

which is part of Equation (2.45), as well as the nodal velocity from Equation (2.49) by an exact method, such as an element-wise Gauss quadrature. We obtain the approximations with a quadratic TLS reconstruction. This implies that only the first three Taylor basis functions are used (i.e, $n_b = 3$). In this case, a two-point Gauss rule within each element leads to an exact integration. The nodal internal forces are computed as follows:

1. Apply a quadratic TLS approach to reconstruct the stress field from the particle data within each active element without specifying the coefficient of the first Taylor basis function:

$$\hat{\sigma}_e = \sum_{i=1}^{n_b} s_i \psi_i, \tag{5.12}$$

where s_i is the coefficient corresponding to Taylor basis function ψ_i. Outside of element e, $\hat{\sigma}_e$ is zero. The global approximation of the stress function, $\hat{\sigma}$, is then equal to

$$\hat{\sigma} = \sum_{e=1}^{\mathcal{N}_e} \hat{\sigma}_e, \tag{5.13}$$

where $\mathcal{N}_e$ is the total number of elements.

2. Integrate the stress approximation using a two-point Gauss quadrature:

$$f_{\text{int},i} \approx \int_\Omega \frac{\partial \phi_i^0}{\partial x}(x)\hat{\sigma}(x,t)\,d\Omega = \sum_{g=1}^{\mathcal{N}_g} \frac{\partial \phi_i^0}{\partial x}\left(x_g\right)\hat{\sigma}\left(x_g\right)\omega_g,$$

where $\mathcal{N}_g$ is the total number of Gauss points, x_g is the global position of a Gauss point, and ω_g is its weight.

The material-point velocities are mapped to the nodes in the following manner:

1. Apply a quadratic TLS approach to reconstruct the density and momentum fields from the particle data within each active element, while preserving the mass and momentum of the element:

$$\hat{\rho}_e = \sum_{i=1}^{n_b} r_i \psi_i \quad \text{with} \quad r_1 = \frac{1}{|\Omega_e|} \sum_{\{p|x_p \in \Omega_e\}} m_p$$

$$(\widehat{\rho v})_e = \sum_{i=1}^{n_b} \gamma_i \psi_i \quad \text{with} \quad \gamma_1 = \frac{1}{|\Omega_e|} \sum_{\{p|x_p \in \Omega_e\}} m_p v_p. \tag{5.14}$$

where r_i and γ_i are the coefficients corresponding to Taylor basis function i. Outside of element e, $\hat{\rho}_e$ and $(\widehat{\rho v})_e$ are equal to zero. The global approximations are then equal to

$$\hat{\rho} = \sum_{e=1}^{\mathcal{N}_e} \hat{\rho}_e \quad \text{and} \quad (\widehat{\rho v}) = \sum_{e=1}^{\mathcal{N}_e} (\widehat{\rho v})_e. \tag{5.15}$$

2. Integrate the approximations using a two-point Gauss quadrature to obtain the linear momentum and the consistent mass matrix:

$$q_i = \sum_{g=1}^{\mathcal{N}_g} \phi_i^0\left(x_g\right)(\widehat{\rho v})\left(x_g\right)\omega_g, \tag{5.16}$$

$$M_{ij} = \sum_{g=1}^{\mathcal{N}_g} \phi_i^0\left(x_g\right)\hat{\rho}\left(x_g\right)\phi_j^0\left(x_g\right)\omega_g. \tag{5.17}$$

It should be noted that the consistent mass matrix may be replaced by a lumped mass matrix without loss of the conservation property of the algorithm.

3. Compute the velocity vector:

$$\mathbf{v} = \mathbf{M}^{-1}\mathbf{q}. \tag{5.18}$$

From Section 5.2.3, it follows that the optimal performance of the TLS technique requires at least three particles in each element at the beginning of the simulation. Since this particle distribution is not preserved under large deformations, virtual data points or material points from neighboring elements may be used to improve the accuracy of the approximation within elements that contain only one or two material points.

5.3.2. CONSERVATION OF MASS AND MOMENTUM

As mentioned in Section 5.3.1, the TLS technique reconstructs the density and linear momentum fields inside each element in the following way:

$$\rho_e \approx \hat{\rho}_e = \sum_{i=1}^{n_b} r_i \psi_i \quad \text{with} \quad r_1 = \frac{1}{|\Omega_e|} \sum_{\{p | x_p \in \Omega_e\}} m_p,$$

$$(\rho v)_e \approx (\widehat{\rho v})_e = \sum_{i=1}^{n_b} \gamma_i \psi_i \quad \text{with} \quad \gamma_1 = \frac{1}{|\Omega_e|} \sum_{\{p | x_p \in \Omega_e\}} m_p v_p.$$

According to Equations (5.8) and (5.9), this preserves the mass $\mathbb{M}_e$ and momentum $\mathbb{Q}_e$ of element e. As a result, the total mass and momentum of the system are conserved after the TLS reconstruction:

$$\mathbb{M} = \sum_{e=1}^{\mathcal{N}_e} \mathbb{M}_e = \sum_{e=1}^{\mathcal{N}_e} \sum_{\{p | x_p \in \Omega_e\}} m_p = \sum_{p=1}^{\mathcal{N}_p} m_p, \tag{5.19}$$

$$\mathbb{Q} = \sum_{e=1}^{\mathcal{N}_e} \mathbb{Q}_e = \sum_{e=1}^{\mathcal{N}_e} \sum_{\{p | x_p \in \Omega_e\}} m_p v_p = \sum_{p=1}^{\mathcal{N}_p} m_p v_p.$$

The mass- and momentum-conservation properties of the mapping obtained using TLS reconstruction and Gauss quadrature can be shown as well.

Since the total mass is equal to the sum of the entries in the mass matrix from Equation (5.17), it can be written as

$$\mathbb{M} = \sum_{i=1}^{\mathcal{N}} \sum_{j=1}^{\mathcal{N}} M_{ij} = \sum_{i=1}^{\mathcal{N}} \sum_{j=1}^{\mathcal{N}} \sum_{g=1}^{\mathcal{N}_g} \phi_i(x_g) \hat{\rho}(x_g) \phi_j(x_g) \omega_g$$

$$= \sum_{g=1}^{\mathcal{N}_g} \hat{\rho}(x_g) \omega_g \sum_{i=1}^{\mathcal{N}} \phi_i(x_g) \sum_{j=1}^{\mathcal{N}} \phi_j(x_g) = \sum_{g=1}^{\mathcal{N}_g} \hat{\rho}(x_g) \omega_g. \tag{5.20}$$

The last equality is derived using the partition of unity property of piecewise-linear and B-spline basis functions. In the remaining part of the proof, we assume that $n_b \leq \frac{2N_g}{N_e}$ and that there are $\frac{\mathcal{N}_g}{\mathcal{N}_e}$ integration points per element (or knot span), so that the Gauss quadrature is exact. Therefore, the following holds:

$$\mathbb{M} = \sum_{g=1}^{\mathcal{N}_g} \hat{\rho}(x_g) \omega_g = \sum_{e=1}^{\mathcal{N}_e} \sum_{\{g | x_g \in \Omega_e\}} \hat{\rho}_e(x_g) \omega_g = \sum_{e=1}^{\mathcal{N}_e} \int_{\Omega_e} \hat{\rho}_e \, d\Omega_e = \sum_{e=1}^{\mathcal{N}_e} \mathbb{M}_e = \sum_{p=1}^{\mathcal{M}} m_p.$$

The last two steps emerge from the conservation of mass per element and Equation (5.19).

For the linear momentum, we also assume that $n_b \leq \frac{2N_g}{N_e}$ and that there are $\frac{N_g}{N_e}$ integration points per element (or knot span). Following the above steps, the total momentum after the mapping can be written as

$$
\begin{aligned}
\mathbb{Q} &= \sum_{i=1}^{\mathcal{N}} q_i = \sum_{i=1}^{\mathcal{N}} \sum_{g=1}^{N_g} \phi_i\left(x_g\right) (\widehat{\rho v})(x_g)\omega_g \\
&= \sum_{g=1}^{N_g} (\widehat{\rho v})(x_g)\omega_g \sum_{i=1}^{\mathcal{N}} \phi_i\left(x_g\right) = \sum_{g=1}^{N_g} (\widehat{\rho v})\left(x_g\right)\omega_g \\
&= \sum_{e=1}^{\mathcal{N}_e} \sum_{\{g \mid x_g \in \Omega_e\}} (\widehat{\rho v})_e\left(x_g\right)\omega_g = \sum_{e=1}^{\mathcal{N}_e} \int_{\Omega_e} (\widehat{\rho v})_e \, d\Omega_e \\
&= \sum_{e=1}^{\mathcal{N}_e} \mathbb{Q}_e = \sum_{p=1}^{\mathcal{M}} m_p v_p.
\end{aligned}
\tag{5.21}
$$

Therefore, we have shown that if the Gauss quadrature is performed using a sufficient number of integration points per element, the mass and momentum balance is satisfied not only by the TLS function reconstruction, but also by its combination with the Gauss quadrature.

5.4. Numerical results

We study the conservation property of the material-point methods by calculating the maximum relative errors in the total mass and momentum over all time steps before and after the computation of the velocity at the nodes or DOFs. For MPM, DDMPM, and BSMPM, the errors in the mass and momentum are bounded by $1 \cdot 10^{-15}$ for the vibrating bar benchmark, and $1 \cdot 10^{-13}$ for the column compaction example. Therefore, the TLS results are only compared to those obtained with the cubic-spline reconstruction.

5.4.1. Bar with fixed ends

This benchmark has been presented in Chapter 3, which also provides its analytical solution in terms of displacement, velocity, and stress. Table 5.2 provides exemplary parameter values for the vibrating bar benchmark under small deformations. For the spatial convergence analysis, we minimize the contribution of temporal errors by using an artificially small time-step size and a short simulation time. In fact, the time-step size and total simulation time of, respectively, $1 \cdot 10^{-7}$ s and $1.9 \cdot 10^{-6}$ s eliminate the contribution of the temporal errors and are significantly increased for the grid-crossing study presented later in this section. Furthermore, the number of elements (knot spans) is varied from 5 to 40, while the initial PPC is fixed to 12. These settings ensure that grid crossing does not occur, and the maximal observed strain is equal to $5.3 \cdot 10^{-7}$ m.

The results in terms of spatial errors are shown in Figure 5.9. As expected, MPM with piecewise-linear basis functions demonstrates second-order convergence in both the displacement and velocity. Since the stress is not discretized directly, but is computed from the displacement by taking the derivative, its convergence rate is one. The application of the TLS reconstruction technique, as well as the cubic-spline interpolation, has

Table 5.2: Exemplary parameters allowing for small deformations in the vibrating bar problem.

Parameter	Symbol	Value	Unit
Length	L	1.00	m
Initial density	ρ	$2.00 \cdot 10^3$	kg/m^3
Young's modulus	E	$7.00 \cdot 10^6$	Pa
Max. initial velocity	v_0	0.28	m/s^2

almost no influence on the stress, but decreases the displacement error by a factor of 1.7. For DDMPM, the application of the reconstruction techniques tends to reduce not only the error in the displacement, but also in the stress. However, velocity results are similar to those obtained for MPM.
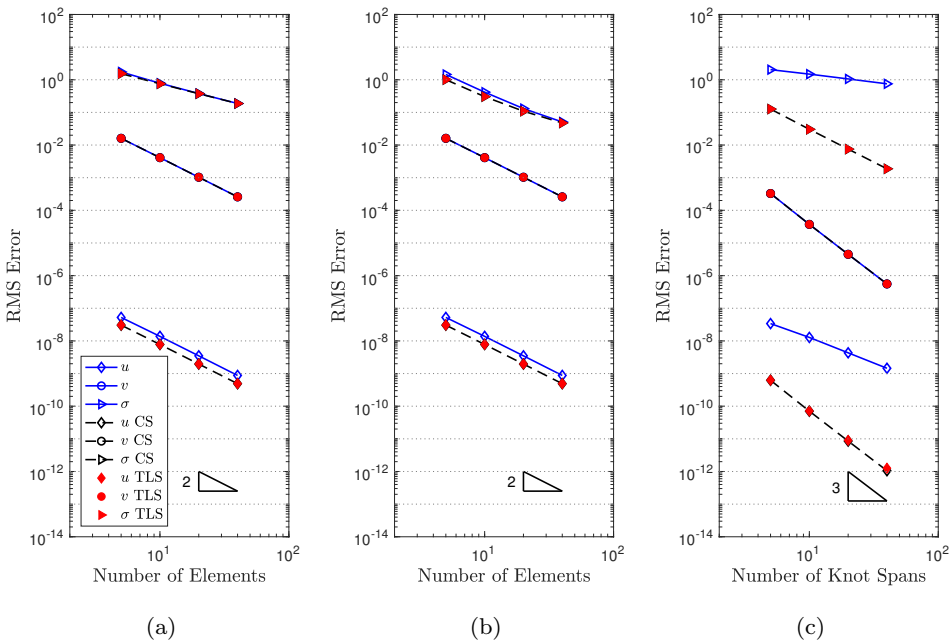

Figure 5.9: Spatial convergence of material-point methods for the vibrating bar problem without grid crossing. The results are shown for the material-point methods without reconstruction techniques, with cubic-spline reconstruction (CS) and TLS reconstruction: (a) MPM, (b) DDMPM, (c) BSMPM.

The use of quadratic B-splines basis functions leads to a significant decrease in the error and a higher convergence order for the velocity, but causes problems at the boundaries of the domain for both stress and displacement. The absolute error in the stress over the domain is shown in Figure 5.10. The large values of the error at the boundaries prevent the reduction of the RMS error and worsen the convergence properties of the method. However, the use of BSMPM with a function reconstruction technique eliminates the boundary issues. An example is provided in Figure 5.11. Consequently, third-
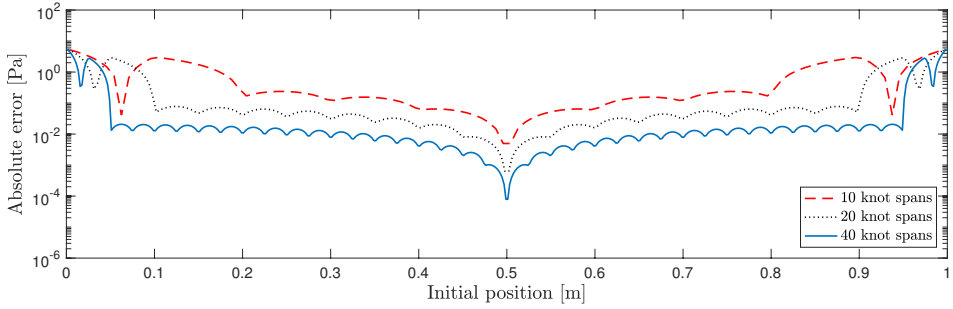
Figure 5.10: Absolute error obtained with BSMPM for stress distribution in the vibrating bar problem without grid crossing.

order convergence is obtained for all considered quantities. It should also be noted that the integration of the TLS or spline reconstruction in BSMPM produces more accurate results than the other considered methods.
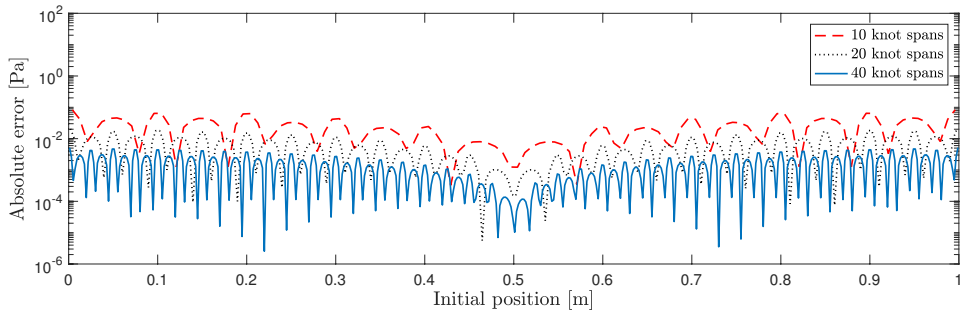

Figure 5.11: Absolute error obtained with BSMPM-TLS for stress distribution in the vibrating bar problem without grid crossing.

Table 5.3 compares the relative error in the mass and momentum made using the TLS and cubic-spline reconstructions. The results are provided for MPM, DDMPM, and BSMPM applied to the vibrating bar problem discretized by 40 elements (knot spans) and 12 PPC. They demonstrate that while the cubic-spline interpolation tends to accurately conserve the mass, it produces errors of order 10^{-6} for the linear momentum. The errors produced by the TLS approach consistently remain close to machine precision and, hence, are orders of magnitude smaller than those generated by the cubic-spline interpolation.

For large-deformation simulations, the parameters from Table 5.4 are used. The time-step size and the simulation time are increased to $1 \cdot 10^{-4}$ s and 0.1 s, respectively. The domain is discretized using 20 elements (knot spans) with initially 8 PPC. The maximal strain that is reached is 0.056 m. Since the analytical solution is not available when the vibrating bar experiences large deformations, the numerical results are compared to the solution obtained with the Updated Lagrangian Finite Element Method (ULFEM)

Table 5.3: Maximum relative errors in the total mass and momentum over the simulation run with cubic-spline and TLS reconstructions when grid crossing does not occur. The vibrating bar is discretized using 40 elements (knot spans) and 12 PPC.

Method	Reconstruction technique	Error mass	Error momentum
MPM	Cubic-spline	$2.3825 \cdot 10^{-12}$	$3.7053 \cdot 10^{-6}$
MPM	TLS	$7.5033 \cdot 10^{-15}$	$2.1007 \cdot 10^{-16}$
DDMPM	Cubic-spline	$3.9211 \cdot 10^{-13}$	$3.7053 \cdot 10^{-6}$
DDMPM	TLS	$7.3896 \cdot 10^{-15}$	$2.5208 \cdot 10^{-16}$
BSMPM	Cubic-spline	$7.3896 \cdot 10^{-15}$	$1.7859 \cdot 10^{-6}$
BSMPM	TLS	$7.5033 \cdot 10^{-15}$	$1.7623 \cdot 10^{-16}$

Table 5.4: Exemplary parameters allowing for large deformations in the vibrating bar problem.

| Parameter	Symbol	Value	Unit
Length	L	1.00	m
Initial density	ρ	$2.00 \cdot 10^3$	kg/m^3
Young's modulus	E	$4.00 \cdot 10^4$	Pa
Max. initial velocity	v_0	0.80	m/s^2

[183].

In the standard MPM simulation, material points cross the element boundaries more than 100 times leading to significant inaccuracies in the results. Although grid crossing influences the computation of the displacement and velocity, its most evident consequences are in the stress distribution. DDMPM and BSMPM reduce the grid-crossing error, but their results still significantly deviate from the solution provided by ULFEM. This is shown in Figures 5.12 and 5.13.

Table 5.5: Maximum relative errors in the total mass and momentum over the simulation run with cubic-spline and TLS reconstructions when grid crossing occurs. The vibrating bar is discretized using 20 elements (knot spans) and 8 PPC.

Method	Reconstruction technique	Error mass	Error momentum
MPM	Cubic-spline	$8.2084 \cdot 10^{-5}$	$8.0098 \cdot 10^{-5}$
MPM	TLS	$2.9104 \cdot 10^{-14}$	$5.8262 \cdot 10^{-15}$
DDMPM	Cubic-spline	$3.5869 \cdot 10^{-5}$	$3.7566 \cdot 10^{-5}$
DDMPM	TLS	$2.8990 \cdot 10^{-14}$	$5.7562 \cdot 10^{-15}$
BSMPM	Cubic-spline	$7.3889 \cdot 10^{-7}$	$1.5794 \cdot 10^{-5}$
BSMPM	TLS	$2.8876 \cdot 10^{-14}$	$5.7186 \cdot 10^{-15}$

Figures 5.12 and 5.13 also illustrate that the application of the TLS approximation or the cubic-spline reconstruction leads to good agreement of the MPM, DDMPM, and BSMPM solutions with that of ULFEM. The maximal reduction of the relative error in the L_2-norm made by standard MPM is achieved when the reconstruction techniques are combined with BSMPM. More precisely, the integration of the spline interpolation or the TLS reconstruction in BSMPM decreases the MPM error by a factor of 13.5 and
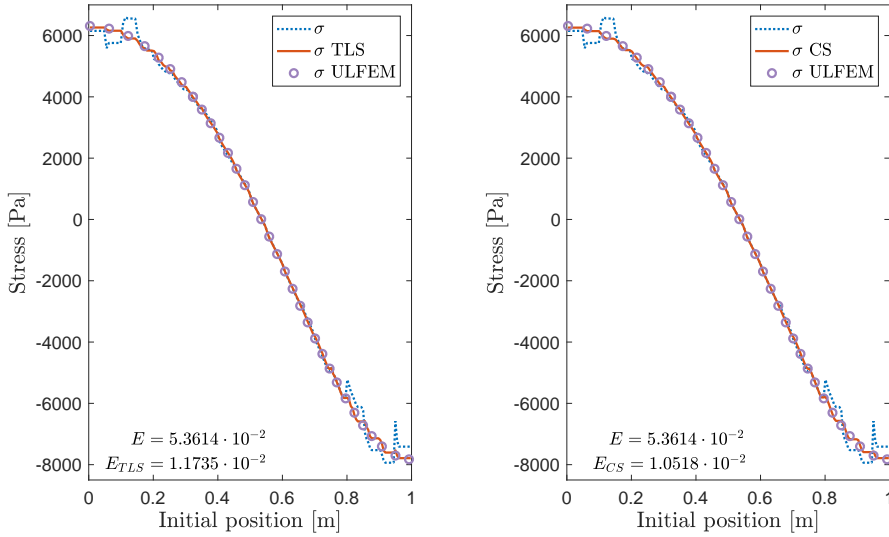
(a) MPM
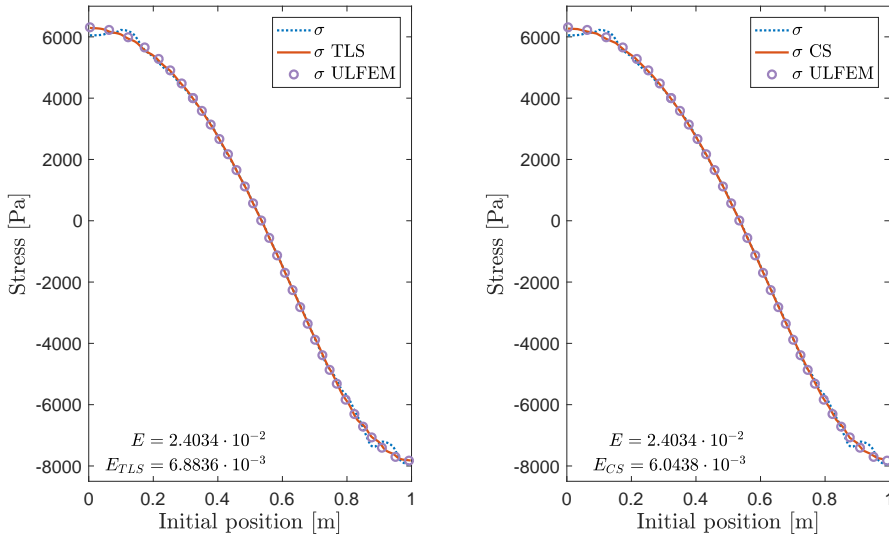
(b) DDMPM

Figure 5.12: Stress distribution and corresponding relative errors in the L_2-norm in the vibrating bar problem with grid crossing. The results are obtained for MPM and DDMPM without reconstruction technique, with cubic-spline reconstruction (CS) and TLS reconstruction, and are compared to the ULFEM results.

9.9, respectively. On the other hand, the combination of the reconstruction techniques with DDMPM leads to highly accurate results as well. The spline and TLS reconstruc-
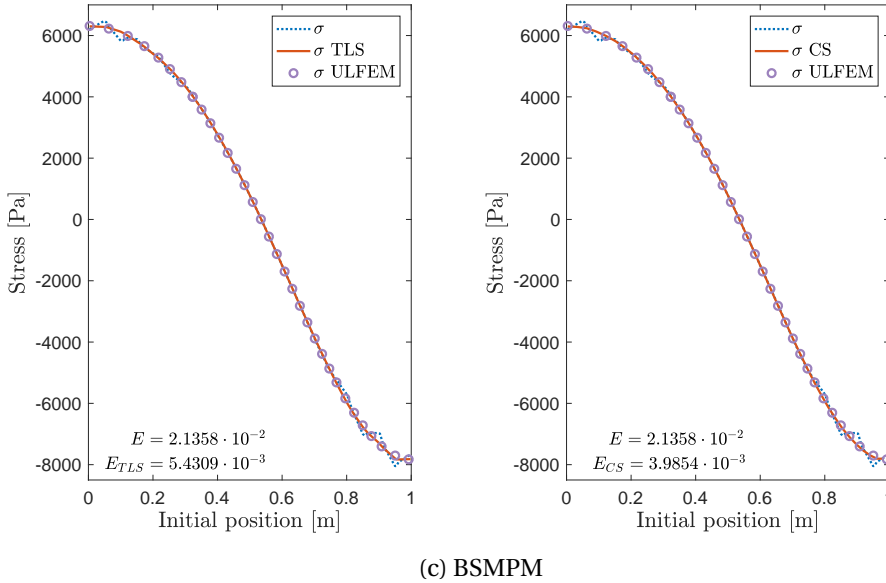
(c) BSMPM

Figure 5.13: Stress distribution and corresponding relative errors in the L_2-norm in the vibrating bar problem with grid crossing. The results are obtained for BSMPM without reconstruction technique, with cubic-spline reconstruction (CS) and TLS reconstruction, and are compared to the ULFEM results.

tion reduces the MPM error by a factor of 8.9 and 7.8, respectively. As expected, the TLS approximation conserves the total mass and linear momentum significantly more accurately than the spline interpolation, regardless of the material-point method. The conservative properties of the reconstruction techniques are provided in Table 5.5.

5.4.2. COLUMN COMPACTION

This example considers the response of a column of material to a compressive body force. The body force increases linearly with time during the simulation. A detailed description of the problem that includes the analytical solution is provided by Bardenhagen and Kober [24].

Table 5.6: Exemplary parameters for the column compaction problem.

Parameter	Symbol	Value	Unit
Initial height	H	50.0	m
Initial density	ρ	1.00	kg/m^3
Young's modulus	E	$1.00 \cdot 10^6$	Pa
Max. body force	b_f	-200	N

Parameter values used for the simulation are listed in Table 5.6. The time-step size and total simulation time are $1 \cdot 10^{-4}$ s and 0.5 s, respectively. Moreover, the number

of elements (knot spans) is equal to 50, while the initial PPC is set to 20. When the standard MPM is used, material points cross element boundaries 180 times, and the maximal strain is equal to 0.27 m.

Table 5.7: Maximum relative errors in the total mass and momentum over the simulation run with cubic-spline and TLS reconstructions for column compaction problem.

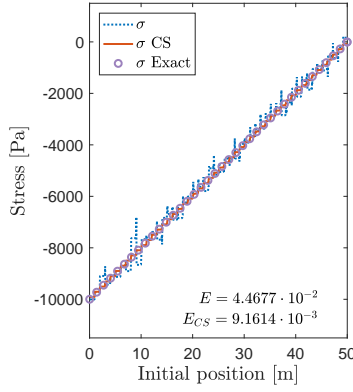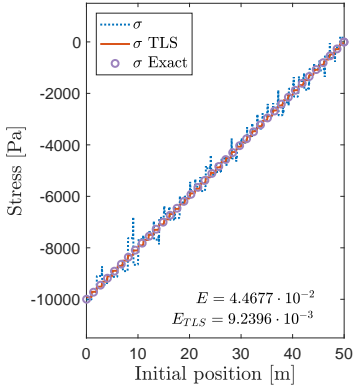| Method	Reconstruction technique	Error mass	Error momentum
MPM	Cubic-spline	$3.0803 \cdot 10^{-6}$	$3.5400 \cdot 10^{-6}$
MPM	TLS	$1.0526 \cdot 10^{-12}$	$5.6413 \cdot 10^{-13}$
DDMPM	Cubic-spline	$4.7571 \cdot 10^{-7}$	$2.0403 \cdot 10^{-6}$
DDMPM	TLS	$1.0524 \cdot 10^{-12}$	$5.6464 \cdot 10^{-13}$
BSMPM	Cubic-spline	$1.5112 \cdot 10^{-9}$	$1.4039 \cdot 10^{-7}$
BSMPM	TLS	$1.0527 \cdot 10^{-12}$	$5.6578 \cdot 10^{-13}$

Table 5.8: Relative error in the L_2-norm for the stress, comparison with the relative error in the L_2-norm of DDMPM-TLS, and computational time normalized with respect to DDMPM-TLS obtained using DDMPM with subpoints for the column compaction problem.

| Number of subpoints	$E_{subpoints}$	$E_{subpoints}/E_{TLS}$	$t_{subpoints}/t_{TLS}$
2	$3.4086 \cdot 10^{-2}$	4.9518	0.3037
4	$2.7207 \cdot 10^{-2}$	3.9524	0.4224
8	$2.1469 \cdot 10^{-2}$	3.1189	0.6793
16	$1.5236 \cdot 10^{-2}$	2.2134	1.2786
32	$8.1219 \cdot 10^{-3}$	1.1799	3.3569
64	$6.2274 \cdot 10^{-3}$	0.9047	6.3747
128	$5.6118 \cdot 10^{-3}$	0.8152	14.422

The obtained results are shown in Figure 5.14. The influence of grid-crossing error on the solution is clearly evident for the standard MPM. While DDMPM only slightly smoothens the oscillations, BSMPM significantly improves the results. In fact, the use of B-spline basis functions reduces the relative error in L_2-norm by a factor of 10.

Both the TLS reconstruction technique and cubic-spline reconstruction interpolation provide considerably more accurate results than MPM and DDMPM. They also further improve the accuracy of BSMPM, but their influence is less distinct in this case. It should be noted that the error produced with the cubic-spline reconstruction is slightly lower than that achieved with the TLS approach. However, the TLS technique outperforms the former method by accurately conserving the mass and linear momentum. As was mentioned previously in this section, the maximum relative errors of MPM, DDMPM, and BSMPM are bounded by $1 \cdot 10^{-13}$. While the TLS approach has little influence on these errors, the spline reconstruction technique increases them to $2.04 \cdot 10^{-6}$ (Table 5.7).

From the table it follows that for a low number of subpoints, the technique of Dhakal and Zhang effectively decreases the relative error of $4.4555 \cdot 10^{-2}$ produced by DDMPM and is less computationally intensive than the TLS technique. However, its performance may be limited by computational time and memory when a higher accuracy has to be reached. In fact, the method approaches the accuracy of the TLS reconstruction tech-

(a) MPM
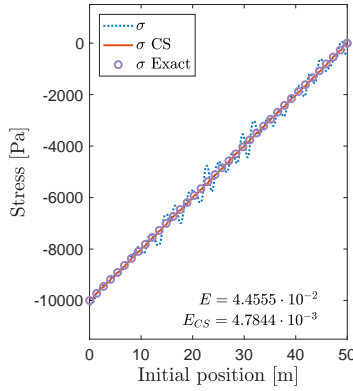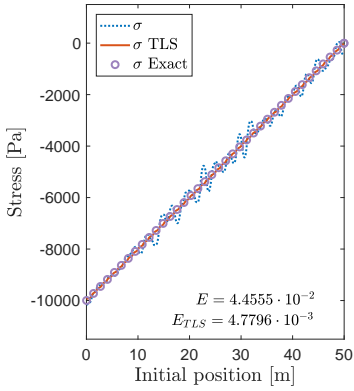
(b) DDMPM
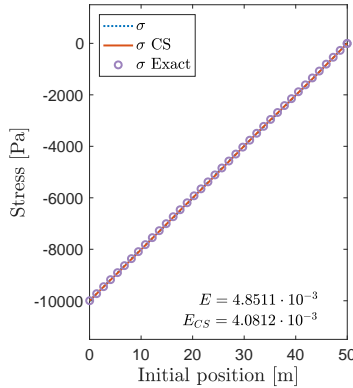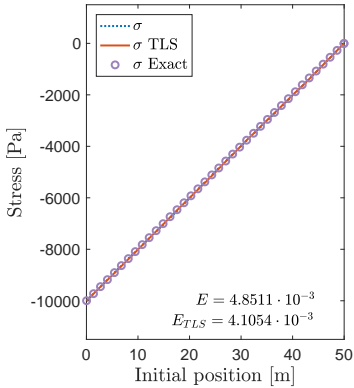
(c) BSMPM

Figure 5.14: Stress distribution and corresponding relative errors in the L_2-norm in the column compaction problem with grid crossing. The results are obtained for material-point methods without reconstruction technique, with cubic-spline reconstruction (CS) and TLS reconstruction, and are compared to the analytical solution.

nique when 128 subpoints per particle are used leading to a significant (factor of 14.4) increase in the computational time.

5.5. SUMMARY AND CONCLUSIONS

In this chapter, we have introduced a novel reconstruction approach, called Taylor least-squares function reconstruction, and applied it to MPM and related methods. The proposed technique combines the least-squares approximation with local Taylor basis functions to accurately reconstruct the quantities of interest (e.g., stress and density fields) from scattered particle data within each element. We have shown that in contrast to standard reconstruction techniques, the TLS approximation conserves the mass and linear momentum of the system after the material-point data is mapped to the integration points. More importantly, in conjunction with a sufficiently accurate numerical quadrature method, the technique preserves the total mass and momentum after the information is projected to the degrees of freedom of the grid. This implies that the TLS reconstruction maintains the physical properties of the standard material-point methods.

The TLS technique was applied to MPM, DDMPM, and BSMPM within the bar-with-fixed-ends vibration and column compaction problems. For the bar-with-fixed-ends problem in the absence of grid crossing, the reconstruction technique had little influence on MPM and DDMPM, but was able to improve the convergence properties of BSMPM. When material points started to cross cell boundaries in both considered examples, the TLS approximation smoothened the solutions of MPM and related methods and brought them closer to the solution computed analytically or by ULFEM.

Furthermore, the novel TLS reconstruction technique was compared to the global spline reconstruction technique used within BSMPM in Chapter 3. In general, the differences in spatial accuracy of the techniques were negligible, but the error in total mass and linear momentum was consistently much lower for the TLS reconstruction. For the column compaction problem, it was also shown that the TLS technique integrated into DDMPM reaches a higher accuracy than the conservative DDMPM method with sub-points, while requiring lower computational costs. Therefore, this study has demonstrated that the combination of MPM and related methods with the TLS technique leads to a higher accuracy of the material-point methods and preserves their fundamental physical properties.

6

DISCUSSION: EXTENSION OF TAYLOR LEAST-SQUARES FUNCTION RECONSTRUCTION TECHNIQUE TO TWO-DIMENSIONAL PROBLEMS AND ITS APPLICATION TO THE MATERIAL-POINT METHOD

After a comprehensive introduction of the Taylor least-squares (TLS) function reconstruction technique and its successful application to one-dimensional material-point method (MPM) simulations in Chapter 5, this chapter describes the extension of the technique to two-dimensional problems. This constitutes the first step towards the application of the TLS reconstruction to more complex and realistic examples. However, further research is required to optimize the use of the TLS reconstruction within two-dimensional MPM computations, in terms of efficiency and robustness. For this reason, this chapter also summarizes the main issues encountered when the technique is adopted within two-dimensional MPM simulations. Section 6.1 outlines the construction of the TLS approximation on both rectangular and triangular grids. Section 6.2 shows a number of examples of the TLS function reconstruction and demonstrates its convergence behavior. Section 6.3 considers the application of the technique to two-dimensional MPM simulations, shows the preliminary results, and explains the main difficulties associated with it. Finally, Section 6.4 draws the conclusions.

6.1. CONSTRUCTION

On two-dimensional domains, the construction of the TLS approximation is similar to the one-dimensional procedure presented in Section 5.2. Since the extension of the least-squares method to two-dimensional data sets $\{\mathbf{x}_p\}_{p=1}^{\mathcal{M}}$ is trivial, this section focuses solely on the local Taylor basis functions. In general, the first six normalized two-dimensional Taylor basis functions are given by [182]

$$\psi_1 = 1, \tag{6.1}$$

$$\psi_2 = \frac{x_1 - x_{1c}}{\Delta x_1}, \tag{6.2}$$

$$\psi_3 = \frac{x_2 - x_{2c}}{\Delta x_2}, \tag{6.3}$$

$$\psi_4 = \frac{(x_1 - x_{1c})^2}{2\Delta x_1^2} - \overline{\frac{(x_1 - x_{1c})^2}{2\Delta x_1^2}}, \tag{6.4}$$

$$\psi_5 = \frac{(x_2 - x_{2c})^2}{2\Delta x_2^2} - \overline{\frac{(x_2 - x_{2c})^2}{2\Delta x_2^2}}, \tag{6.5}$$

$$\psi_6 = \frac{(x_1 - x_{1c})(x_2 - x_{2c})}{\Delta x_1 \Delta x_2} - \overline{\frac{(x_1 - x_{1c})(x_2 - x_{2c})}{\Delta x_1 \Delta x_2}}. \tag{6.6}$$

Here, the notation for the volume average from Chapter 5 is adopted (see Equation 5.6), and x_{kc} with $k = \{1, 2\}$ denote the coordinates of the cell centroid. Further evaluation of the local Taylor basis functions strongly depends on the element type.

6.1.1. RECTANGULAR ELEMENTS

For rectangular elements, $x_{1c} = \frac{x_{1\max} + x_{1\min}}{2}$, $x_{2c} = \frac{x_{2\max} + x_{2\min}}{2}$, $\Delta x_1 = \frac{x_{1\max} - x_{1\min}}{2}$, and $\Delta x_2 = \frac{x_{2\max} - x_{2\min}}{2}$. The volume $|\Omega_e|$ of a rectangular cell e is equal to

$$|\Omega_e| = (x_{1\max} - x_{1\min})(x_{2\max} - x_{2\min}). \tag{6.7}$$

The first three Taylor basis functions are computed by substituting the above expressions into Equations (6.1) to (6.3). Using the definition of the volume average of a function, ψ_4, ψ_5, and ψ_6 can be obtained from

$$\psi_4 = \frac{(x_1 - x_{1c})^2}{2\Delta x_1^2} - \frac{1}{2\Delta x_1^2 (x_{1\max} - x_{1\min})} \left(\frac{1}{3} \left(x_{1\max}^3 - x_{1\min}^3 \right) - x_{1c} \left(x_{1\max}^2 - x_{1\min}^2 \right) \right. \tag{6.8}$$

$$\left. + x_{1c}^2 (x_{1\max} - x_{1\min}) \right), \tag{6.9}$$

$$\psi_5 = \frac{(x_2 - x_{2c})^2}{2\Delta x_2^2} - \frac{1}{2\Delta x_2^2 (x_{2\max} - x_{2\min})} \left(\frac{1}{3} \left(x_{2\max}^3 - x_{2\min}^3 \right) - x_{2c} \left(x_{2\max}^2 - x_{2\min}^2 \right) \right. \tag{6.10}$$

$$\left. + x_{2c}^2 (x_{2\max} - x_{2\min}) \right), \tag{6.11}$$

$$\psi_6 = \frac{(x_1 - x_{1c})(x_2 - x_{2c})}{\Delta x_1 \Delta x_2} - \left(\frac{1}{\Delta x \Delta y |\Omega_e|} \left(\frac{1}{2} \left(x_{1\max}^2 - x_{1\min}^2 \right) - x_{1c} \left(x_{1\max} - x_{1\min} \right) \right) \cdot \quad (6.12)$$

$$\left(\frac{1}{2} \left(x_{2\max}^2 - x_{2\min}^2 \right) - x_{2c} \left(x_{2\max} - x_{2\min} \right) \right) \right). \quad (6.13)$$

The basis functions are illustrated in Figure 6.2. The figure shows that all basis functions have zero integral value, except ψ_1, whose integral value is equal to the element's volume.

6.1.2. Triangular elements

While the computation of the Taylor basis functions within a rectangular element can be performed generically, on a triangular mesh, the calculation is significantly simplified by transferring the elements into a reference triangle. Such transformations are common for FEM and are also required for quadrilateral elements other than the discussed above rectangular elements.

Mapping from a general to the reference triangle

A schematic representation of the mapping from a general triangle to the reference triangle is shown in Figure 6.1.
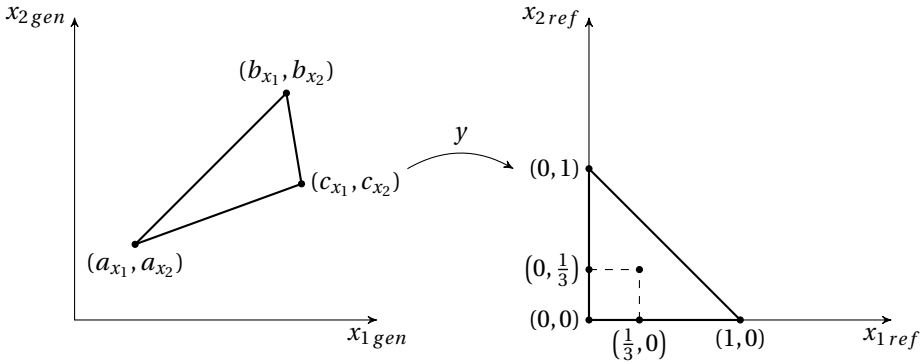
Figure 6.1: Schematic illustration of the mapping to the reference triangle.

The affine map y from a general to the reference triangle can be written as follows [184]:

$$y(\mathbf{x}_{gen}) = A \left(\mathbf{x}_{gen} - \begin{pmatrix} a_{x_1} \\ a_{x_2} \end{pmatrix} \right) = \mathbf{x}_{ref}, \quad (6.14)$$

where $\mathbf{x}_{gen}$ and $\mathbf{x}_{ref}$ are the coordinated within the general and reference triangles, repsectively. The matrix A is defined as follows:

$$A = \frac{1}{\left(c_{x_1} - a_{x_1} \right) \left(b_{x_2} - a_{x_2} \right) - \left(c_{x_2} - a_{x_2} \right) \left(b_{x_1} - a_{x_1} \right)} \begin{bmatrix} b_{x_2} - a_{x_2} & -\left(b_{x_1} - a_{x_1} \right) \\ -\left(c_{x_2} - a_{x_2} \right) & c_{x_1} - a_{x_1} \end{bmatrix}. \quad (6.15)$$

Figure 6.2: The first six Taylor basis functions on a rectangular element.

Figure 6.3: The first six Taylor basis functions on the reference triangle.

LOCAL TAYLOR BASIS FUNCTIONS
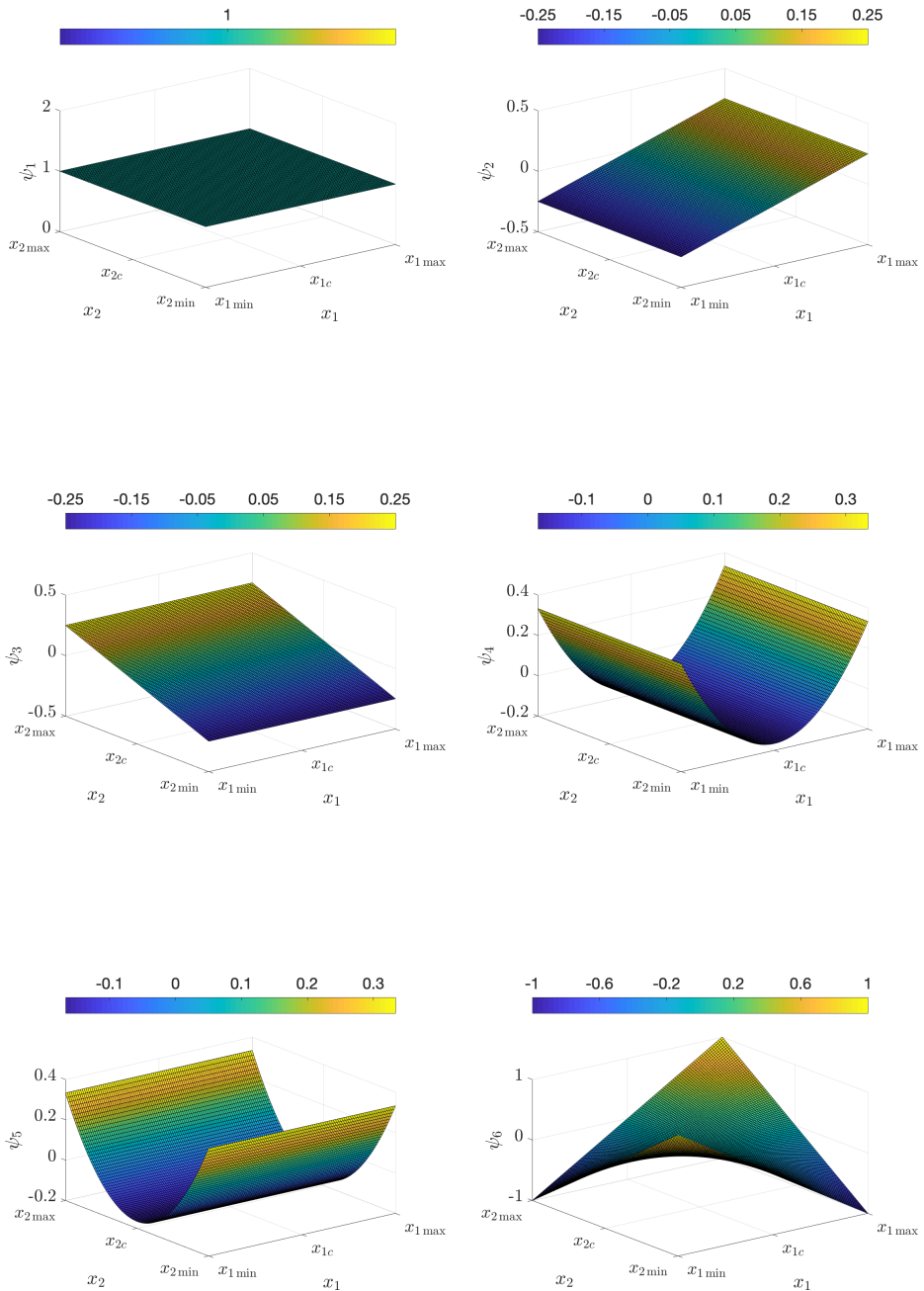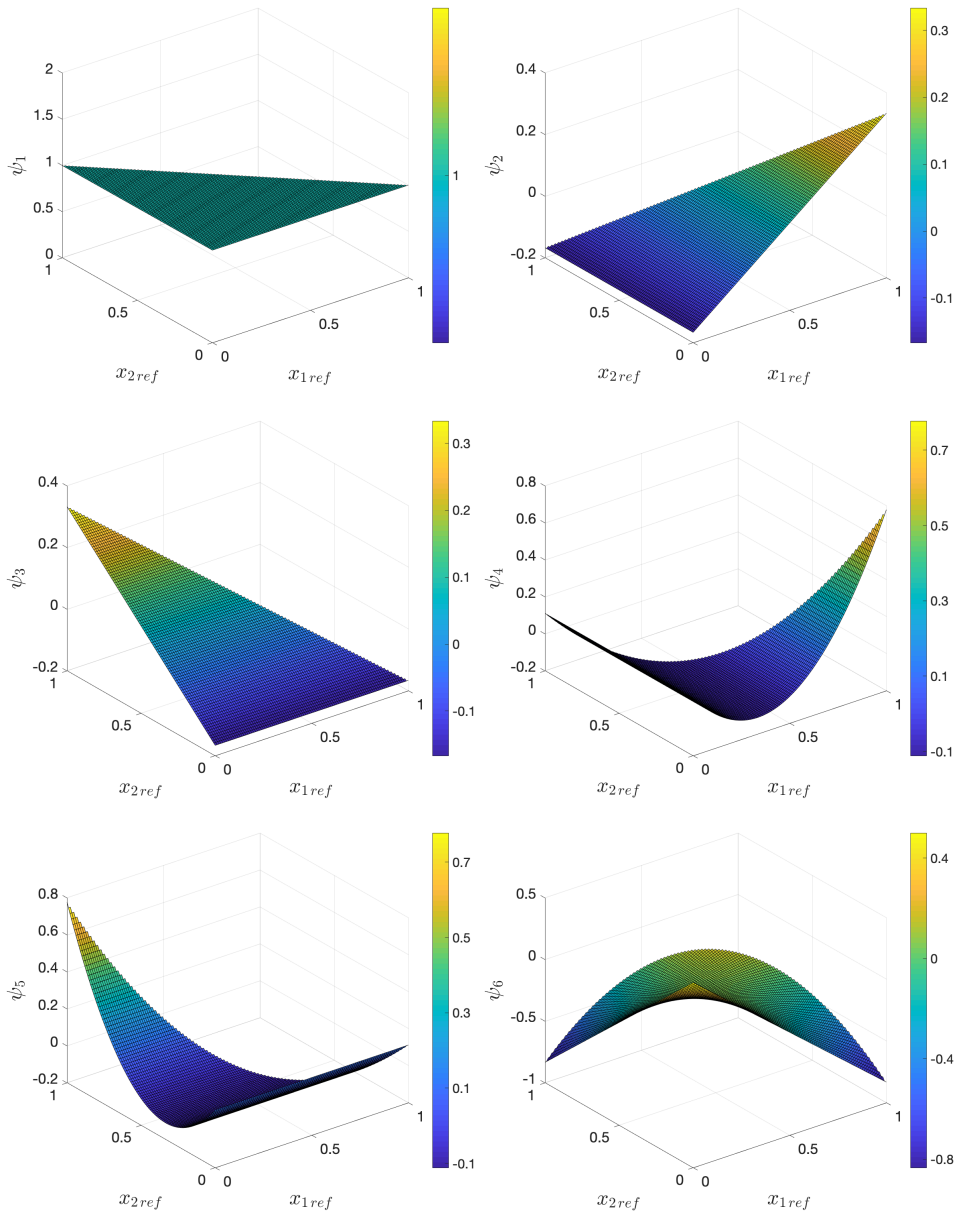
For the standard triangle, $\Delta x_1 = \Delta x_2 = \frac{1}{2}$ and $|\Omega_e| = \frac{1}{2}$. The cell centroid is located at $\left(\frac{1}{3}, \frac{1}{3}\right)$. The first three Taylor basis functions are obtained by simply substituting these values into Equations (6.1) to (6.3). The local Taylor basis functions requiring the computation of the volume average can be written as

$$\psi_4 = 2\left(x_1 - \frac{1}{3}\right)^2 - \frac{1}{9}, \tag{6.16}$$

$$\psi_5 = 2\left(x_2 - \frac{1}{3}\right)^2 - \frac{1}{9}, \tag{6.17}$$

$$\psi_6 = 4\left(x_1 - \frac{1}{3}\right)\left(x_2 - \frac{1}{3}\right) + \frac{1}{18}. \tag{6.18}$$

The local Taylor basis functions on the reference triangle are shown in Figure 6.3. As previously, the figure demonstrates that all basis functions have zero integral value, except ψ_1, whose integral value is equal to the element's volume.

6.2. EXAMPLES OF TAYLOR LEAST-SQUARES RECONSTRUCTION

6.2.1. RECTANGULAR GRID

To illustrate the TLS approximation on a rectangular grid, the function $f(x_1, x_2)$ defined as $= \sin(x_1)\cos(x_2)$ is reconstructed on the $[0,5] \times [0,5]$ domain. The corresponding integral value is approximately equal to $-6.8691 \cdot 10^{-1}$. The domain is discretized by 1×1 rectangular elements with 6 data points are placed within each element. Their x_1-coordinates are set equal to $x_{1\min} + \frac{1}{3}$ and $x_{1\max} - \frac{1}{3}$, and x_2-coordinates set to $x_{2\min} + \frac{1}{4}$, $\frac{x_{2\min} + x_{2\max}}{2}$, and $x_{2\max} - \frac{1}{4}$. Six Taylor basis functions are used for the approximation and four Gauss points per element are defined to compute the RMS and relative errors. The RMS error for the function is equal to $6.17 \cdot 10^{-3}$, while the relative error for the integral is equal to $6.6613 \cdot 10^{-16}$. The reconstruction is visualized in Figure 6.4.

After that, the TLS approximation is produced using 150 randomly distributed data points, while keeping all other settings unchanged. In this case, the RMS error for f reaches $5,48 \cdot 10^{-2}$, but the relative error for the integral of the reconstructed function remains accurate to machine precision (i.e., $1.1102 \cdot 10^{-16}$). The results are illustrated in Figure 6.5.

The convergence properties of the two-dimensional Taylor basis functions are depicted in Figure 6.6. The reconstruction is obtained on the $[0,10] \times [0,10]$ domain for $f(x_1, x_2) = \sin\left(\frac{x_1}{10}\right)\cos\left(\frac{x_2}{10}\right)$, which excludes high frequency oscillations from the coarsest grid and hence allowing for an accurate reconstruction on all considered meshes. Data points are distributed uniformly within the domain and each element contains 9 data points. As expected, when only the first Taylor basis function ψ_1 is used the convergence order is equal to one, whereas with the first three basis functions $\{\psi_1, \psi_2, \psi_3\}$ the second-order convergence is obtained and with the first six basis functions the convergence order is equal to three.

Figure 6.4: TLS reconstructions of $f(x_1, x_2) = \sin(x_1)\cos(x_2)$ on $[0,5] \times [0,5]$ using 25 rectangular elements with 6 data points within each element: side and top view. The TLS reconstruction is evaluated at Gauss-point positions within each element.

6
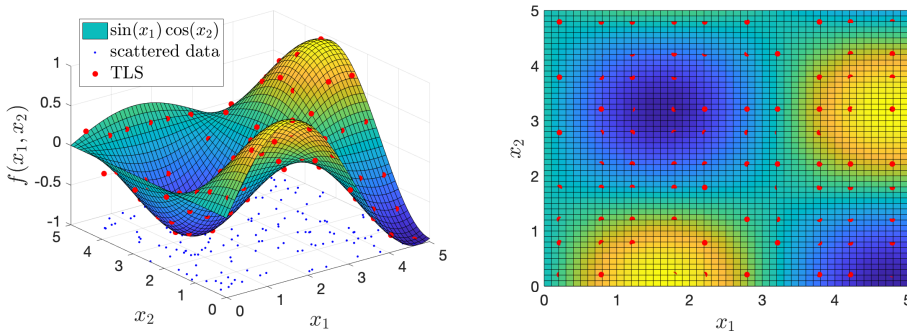
Figure 6.5: TLS reconstructions of $f(x_1, x_2) = \sin(x_1)\cos(x_2)$ on $[0,5] \times [0,5]$ using 25 rectangular elements with randomly distributed data points: side and top view. The TLS reconstruction is evaluated at Gauss-point positions within each element.
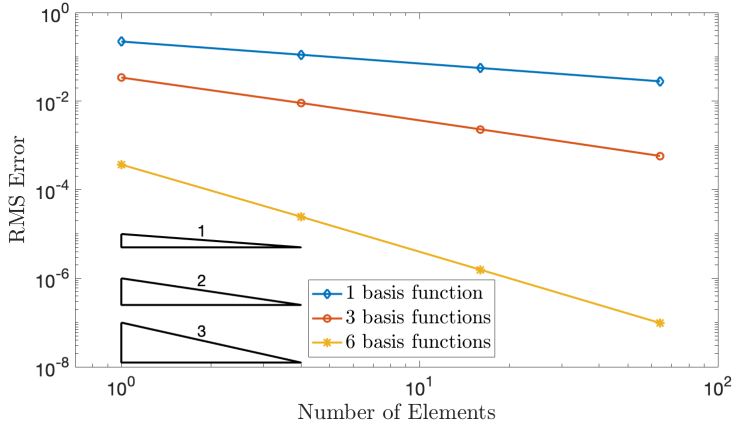
Figure 6.6: Convergence behavior of the two-dimensional TLS approximation. The basis functions are constructed on the $[0, 10] \times [0, 10]$ domain using nine equally distributed data points within each rectangular element.

6.2.2. Triangular grid

Here, it is briefly illustrated that similar results can be obtained for the TLS reconstruction on a triangular grid. Using 300 data points uniformly distributed over the $[0, 5] \times [0, 5]$ domain, the RMS error for $f(x_1, x_2) = \sin(x_1)\cos(x_2)$ is equal to $1.80 \cdot 10^{-3}$, while the integral is preserved to machine precision. The reconstructed function is shown in Figure 6.7.
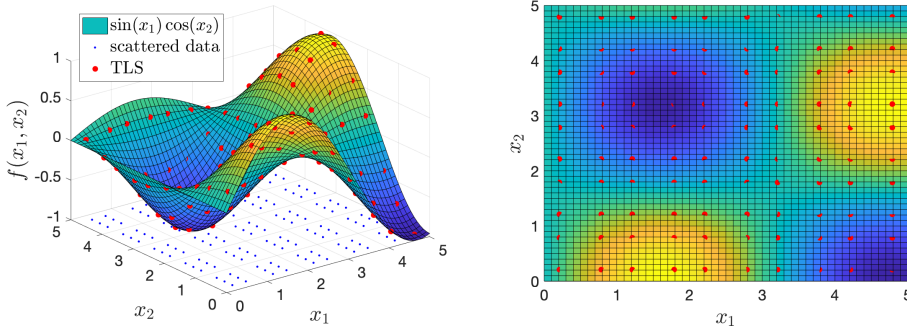

Figure 6.7: TLS reconstructions of $f(x_1, x_2) = \sin(x_1)\cos(x_2)$ on $[0, 5] \times [0, 5]$ using 30 triangular elements with 6 data points within each element: side and top view. The TLS reconstruction is evaluated at Gauss-point positions within each element.
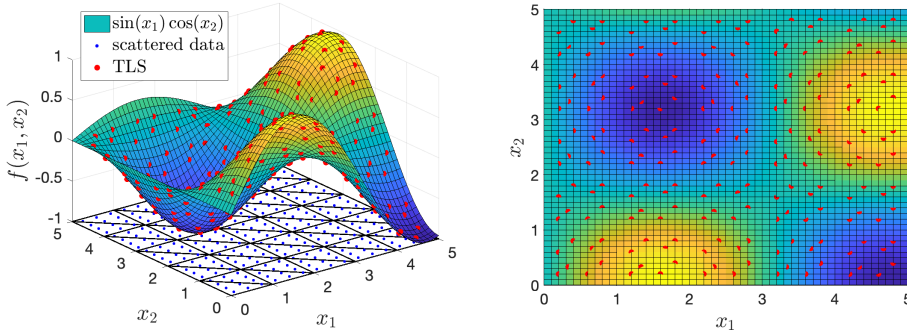
6.3. APPLICATION OF TAYLOR LEAST-SQUARES TECHNIQUE TO THE MATERIAL-POINT METHOD

The application of the TLS function reconstruction technique within two-dimensional MPM simulations is strongly related to the procedure described in Section 5.3. Therefore, this section focuses only on the multi-dimensional aspects of the approach. In order to compute the two-dimensional internal forces, the TLS reconstruction is applied to all the components of the stress tensor separately (see Equation 5.12 and 5.13). The obtained approximations are used for Gauss integration in the following way:

$$f_{int_{x_1},i} \approx \sum_{g=1}^{\mathcal{N}_g} \left(\frac{\partial \phi_i^0}{\partial x_1} \left(\mathbf{x}_g \right) \sigma_{x_1 x_1} \left(\mathbf{x}_g \right) + \frac{\partial \phi_i^0}{\partial x_2} \left(\mathbf{x}_g \right) \sigma_{x_1 x_2} \left(\mathbf{x}_g \right) \right) \omega_g, \qquad (6.19)$$

$$f_{int_{x_2},i} \approx \sum_{g=1}^{\mathcal{N}_g} \left(\frac{\partial \phi_i^0}{\partial x_2} \left(\mathbf{x}_g \right) \sigma_{x_2 x_2} \left(\mathbf{x}_g \right) + \frac{\partial \phi_i^0}{\partial x_1} \left(\mathbf{x}_g \right) \sigma_{x_1 x_2} \left(\mathbf{x}_g \right) \right) \omega_g. \qquad (6.20)$$

Here, $\sigma_{x_k x_l}$ with $k, l = \{1, 2\}$ represents a component of the stress tensor.

Similarly, the TLS technique is applied for the computation of the linear momentum field $\left(\widehat{\rho \mathbf{v}} \right)$, which is used to compute the components of the linear momentum. More precisely, Equations (5.14), (5.15), and (5.16) are extended to both the x_1- and x_2-directions. This allows for the components of the nodal velocity to be obtained in analogy with Equation (5.18). Moreover, the extension of the TLS method to two dimensions does not affect its properties in terms of the conservation of mass and momentum within MPM and related methods. While this is directly evident for the mass conservation, the momentum conservation can be shown by repeating the proof in Equation (5.21) for both directions.

6.3.1. PRELIMINARY NUMERICAL RESULTS

The benchmark describing the plate undergoing axis-aligned displacement, which has been described in Chapter 2, is considered in order to investigate the performance of the TLS function reconstruction within two-dimensional MPM simulations. As previously, the plate is assumed to be a square with length and width of 1 m. Young's modulus and Poisson's ratio are set to 10^4 kPa and 0.3, respectively. The initial density is equal to 1000 kg/m^3 and the maximum amplitude of displacement is limited to 0.05 m. The time step size and total simulation time are set to $1.25 \cdot 10^{-4}$ s and 0.015 s, respectively.

Table 6.1: Maximum relative errors in the total mass and and absolute errors in the momentum over the simulation run with MPM and MPM with the TLS reconstruction. The plate is discretized using rectangular mesh with 1024 elements and a triangular mesh with 1058 elements with 24 material points per element.

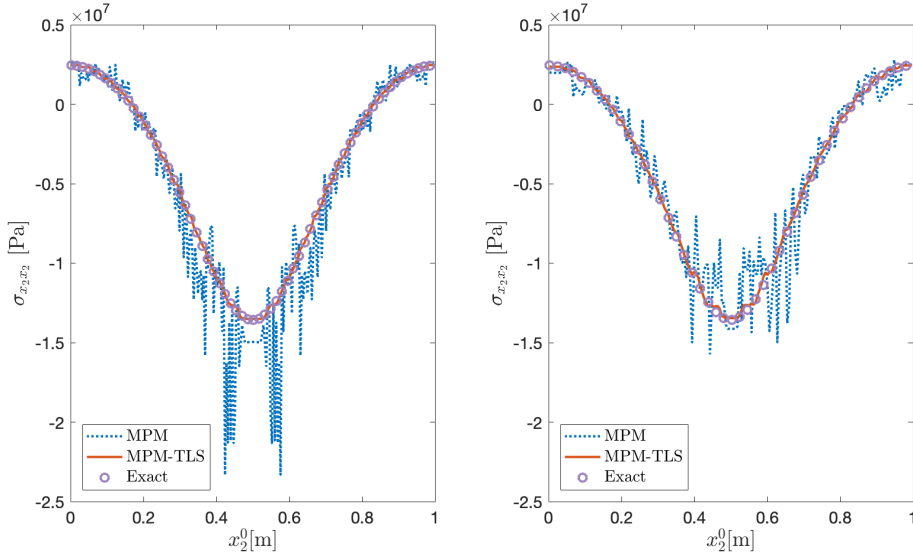| Method	Mesh	Error $\mathbb{M}$	Error $\mathbb{Q}_x$ [Pa]	Error $\mathbb{Q}_y$ [Pa]
MPM	Rectangular	$7.2100 \cdot 10^{-13}$	$8.7421 \cdot 10^{-13}$	$1.9854 \cdot 10^{-13}$
MPM-TLS	Rectangular	$7.3351 \cdot 10^{-13}$	$8.1654 \cdot 10^{-11}$	$4.7041 \cdot 10^{-11}$
MPM	Triangular	$3.3253 \cdot 10^{-13}$	$1.0270 \cdot 10^{-12}$	$1.7689 \cdot 10^{-11}$
MPM-TLS	Triangular	$3.3265 \cdot 10^{-13}$	$6.9142 \cdot 10^{-13}$	$7.7780 \cdot 10^{-12}$

Figure 6.8: Numerical results in terms of the normal stress in x_2-direction for material points with $x_1^0 = 0.5$ m. The results are obtained for a rectangular mesh with 1024 elements (left) and a triangular mesh with 1058 elements (right) with 24 material points per element. The stresses distributions are obtained for standard MPM and MPM with the TLS reconstruction.

Figure 6.8 illustrates that the TLS technique significantly improves the MPM results for both triangular and rectangular elements. In addition, Table 6.1 shows that the use of the TLS approximation within MPM preserves the physical properties of standard MPM with respect to mass and linear momentum conservation during the mapping of the material-point data to the background grid. This is similar to the results in Chapter 5, where it was additionally shown that the cubic-spline reconstruction does not posses such conservation properties. Nevertheless, the MPM-TLS results are strongly dependent on the number of material points and their location within each element. For instance, simulations performed on a triangular grid with 1058 elements and 12 or 16 uniformly distributed material points per element terminate prematurely due to the linear dependence of the columns of the matrix $D = \sum_{p=1}^{\mathcal{M}} \boldsymbol{\psi}\left(\mathbf{x}_p\right) \boldsymbol{\psi}^\top\left(\mathbf{x}_p\right)$ in the least-squares computation (see Section 5.2.1). This imposes a major limitation of the approach, because large deformations do not allow to control the number and positions of the particles within elements.

6.3.2. FUTURE RESEARCH

Section 6.3.1 indicates that data-point distribution becomes a critical issue for the application of the TLS function reconstruction technique within two-dimensional MPM simulations. Although limited information from the data points posed a problem for one-dimensional simulations as well, in general, the availability of only three data points with distinct locations was sufficient to enable linear independence of the columns of

the matrix D. A number of options that assure a successful TLS computation regardless of the information available from the data points were presented in Section 5.2.3. A similar study is required for two-dimensional simulations. The following topics should be considered:

- *Data distribution.* First of all, the properties of data-point (or material-point, in case of MPM-TLS computations) distributions, which support linear independence of the columns of the matrix D, should be identified. According to Section 6.3.1, the accuracy of the TLS reconstruction depends on the data distribution within the domain. In addition, Section 6.3.1 indicates stability issues within MPM-TLS simulations that are related to the number and position of material points.

- *Completion strategy.* After the specifications for the data distribution are identified, a suitable method to achieve these specifications should be found. For one-dimensional examples, Section 5.2.3 proposes either the direct use of neighboring data points or application of the TLS technique to obtain the data for the virtual points. Theoretically, both techniques are suitable for two-dimensional computations. However, their implementation requires access to neighboring elements and a search algorithm, which may affect the efficiency of the computation. Nevertheless, obtaining extra information directly from the neighboring data points is cheaper than its preliminary mapping to the virtual data points. Thus, given a robust and efficient search algorithm, the extension of the neighboring-data-point technique can offer an effective solution for two-dimensional problems.

- *Mesh-refinement techniques.* The accuracy of the TLS approximation can be further increased by mesh-refinement techniques. These techniques should be used to avoid prescribing non-zero function values to the parts of an element where no material, and hence, no material points are located (see Figure 6.9).
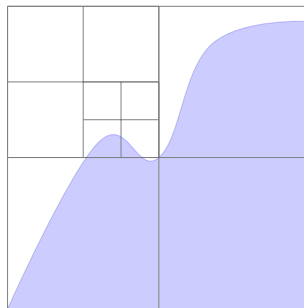
Figure 6.9: An example of mesh refinement to prevent non-zero function values in the top left corner of the domain, which does not contain the continuum.

- *Computational costs.* It was illustrated in Chapter 5 that the TLS approximation requires considerable computational effort for one-dimensional problems. For two-dimensional problems, the computational costs increase further, even without a completion strategy or a mesh-refinement technique. A reduction of computational time can be achieved by exploiting the local nature of the method for its

parallel implementation: the standard TLS computation only requires the information specific to the considered element. However, the costs of the parallel implementation can be significantly increased if a completion technique is adopted. In fact, certain designs of the strategies based on neighboring or virtual data points can lead to dense communication patterns.

6.4. CONCLUSIONS

In this chapter, the extension of the TLS function reconstruction technique to rectangular and triangular elements as well as its application within two-dimensional MPM simulations have been discussed. It has been shown that the technique can provide an accurate reconstruction with a relatively low number of data points within each element. However, for its application to two-dimensional MPM problems, further research is required.

6

IV

CLOSURE

7
CONCLUSIONS

This chapter describes the key findings of the literature study, provides the main conclusions of this thesis, and identifies possible directions for future research. Section 7.1 gives a short summary of the literature study. Section 7.2 presents an overview of the conclusions drawn from the original research of this thesis. Section 7.3 provides a number of suggestions for future research.

7.1. SUMMARY OF THE LITERATURE STUDY

This thesis provides an overview of the development trends within MPM based on the existing literature. It identifies the following main directions in the development of MPM: formulations, time-integration schemes, algorithms, and related methods. Each of these directions is shortly described below.

To apply MPM to different types of material, the research on its formulations is essential. In the existing literature, the distinction is made between one-, two-, and three-phase problems. Furthermore, in previous studies, two-phase materials are modeled by formulations that use either one or two sets of material points. When only one set is adopted, each particle contains both phases, whereas double-point formulations represent each phase by a separate set of material points.

At the same time, MPM has been combined with both explicit and implicit time-integration schemes. Previous research points out that the main advantage of implicit schemes over the explicit ones is their high numerical accuracy.

Furthermore, four versions of MPM algorithm have been proposed: USL, MUSL, USF, and USAVG. The MUSL algorithm combined with a lumped mass matrix reduces the stability issues observed with the USL algorithm. Both the USF and USAVG algorithms attempt to improve the energy conservation within the MUSL algorithm. However, it has been shown that the MUSL algorithm damps out unresolved modes more accurately than the USF and USAVG algorithms.

A number of methods have been published that are based on standard MPM, but either reduce or completely eliminate grid-crossing errors. The GIMP and CPDI methods mitigate grid-crossing inaccuracies by introducing an alternative particle representation. The GIMP method is a generalization of MPM that represents material points by particle-characteristic functions. The CPDI method extends GIMP in order to accurately capture shear distortion. On the other hand, the DDMPM preserves the original point-mass representation of the material points, but adjusts the gradients of the basis functions to avoid grid-crossing errors. Finally, BSMPM eliminates the oscillations related to cell crossing by replacing piecewise-linear basis functions by higher-order B-spline basis functions.

Of course, the classification of the research on the development of MPM that is provided in this thesis is not comprehensive. Many fundamental studies of the method are not classified as part of a certain research course due to their unique nature.

7.2. GENERAL CONCLUSIONS

The main conclusions of this dissertation are presented according to the specified research objectives.

- *Connection to meshfree methods.*
  This thesis highlights the close relation between MPM and the broader family of meshfree methods by establishing a connection between MPM and the optimal transportation meshfree method.
  MPM and the OTM method are derived in a different fashion: MPM is based on the weak form of the momentum equation, while the OTM method uses concepts from optimal transportation theory. In addition, MPM solves the governing equa-

tions on the background grid and typically employs piecewise-linear basis functions, while the OTM method does not require a mesh and adopts maximum entropy basis functions.

This thesis shows that despite these differences, MPM and the OTM method can be related. In fact, assuming a constant time step, the validity of the backward Euler scheme for material-point displacement in MPM, as well as the linear completeness of the MPM basis functions and their ability to translate nodal velocities into material-point velocities, the only distinction between the algorithms emerges from the update of the basis functions and their gradients. However, this is a fundamental difference that results in a combination of Eulerian and Lagrangian descriptions for MPM and a fully Lagrangian description for the OTM method.

Based on this comparison between MPM and the OTM method, a unified approach is proposed. Similarly to BSMPM, this hybrid method uses B-spline basis functions, but updates them following the OTM approach. The unified approach shows high levels of performance in terms of numerical stability, computational effort, and simplicity of implementation and use.

- *Further development of BSMPM.*
  In this thesis, BSMPM is combined with a mapping technique that allows for the projection of material-point data to the background grid using cubic-spline interpolation. Furthermore, a version of BSMPM is proposed that extends the method to unstructured triangular grids which allows for the application of the method to more realistic and complex problems.

  The mapping strategy works as follows. First the field variables of interest are reconstructed from scattered material-point data by means of cubic-spline interpolation. After that, the reconstructed field variables are evaluated at Gauss-point positions, which are subsequently used for Gauss quadrature. In this dissertation, each knot span is divided into two equal parts, and within each part two Gauss points are placed. Therefore, polynomials up to third order can be integrated exactly within each half of a knot span. The numerical results show that the modified mapping strategy improves the performance of BSMPM on the boundaries of the domain, yields higher-order convergence of the method, and reduces the absolute error.

  The extension of BSMPM to triangular grids is achieved by employing quadratic Powell-Sabin splines. For relatively simple problems, this technique demonstrates promising results.

- *Development of an advanced technique to reconstruct functions from scattered data points and its application to MPM and related methods.*
  This thesis introduces Taylor least-squares function reconstruction technique. The TLS reconstruction is a novel technique that combines the least-squares approximation with local Taylor basis functions. It is designed to accurately reconstruct functions from scattered data points, while preserving a priori known integral values. The numerical examples demonstrate that the TLS function reconstruction reproduces given integrals up to machine precision regardless of the data-point

distribution.

The technique can be applied within MPM and related methods for the projection of material-point data to the background grid, essentially forming an alternative to the function reconstruction with cubic splines. In conjunction with MPM, the TLS technique reconstructs field variables, such as stress and density fields, locally within each element. After that, it adopts Gauss quadrature rules to obtain high-accuracy approximations of the internal force and momentum vectors as well as the mass matrix. The main advantage of the TLS technique over other alternative mapping strategies (e.g., the mapping with cubic spline interpolation) is that it conserves the total mass and linear momentum of the system.

In this dissertation, the TLS mapping has been applied to MPM, DDMPM, and BSMPM and shown to significantly improve their performance. It is also demonstrates that the technique conserves the total mass and momentum with precision similar to that of the original direct mapping. This implies that for the considered examples, the TLS function reconstruction technique resembles the approximation properties of highly-accurate spline reconstruction, while preserving the physical properties of the standard algorithm.

7.3. Suggestions for future research

The results provided in this dissertation point to several opportunities for future research that would further contribute to the goals of this thesis or broaden its research scope.

- As suggested in Chapter 6, further investigation of the TLS function reconstruction for two-dimensional problems is required. The study should identify data distributions, from which an accurate approximation can be constructed without causing stability issues in the least-squares approximation, and find a strategy that allows for such distributions from available data. The accuracy of the approximation can be further increased by mesh-refinement techniques. Aside from the robustness and accuracy of the method, further research should consider its computational costs. After that, the technique can be applied for the particle-data projection within MPM and related methods using triangular or quadrilateral elements.

- The combination of MPM with Powell-Sabin splines for unstructured triangular grids has to be adopted to problems that contain partially filled elements. A preliminary study has indicated that partial lumping of the mass matrix can be used to provide a stable and relatively accurate solution. However, the robustness and efficiency of this approach need further testing. In addition, future studies can investigate if other basis functions suited for unstructured C^1-discretizations can overcome the issues associated with Powell-Sabin splines within MPM.

- Powell-Sabin splines can be employed within the unified approach that combines BSMPM and the OTM method. On the one hand, the update of the degrees of freedom throughout the simulation may offer an alternative to partial lumping of the mass matrix. On the other hand, the applicability to unstructured grids brings the unified approach closer to the original OTM method, while avoiding the use of the local maximum entropy basis functions.

- The research presented in this thesis offers new mathematical techniques that can be applied to MPM. However, the considered benchmarks are limited to one-phase problems. The use of the proposed techniques with multi-phase formulations should lead to more accurate and stable simulations of complex phenomena.

- In addition, computational aspects of the presented techniques should be further investigated. For example, a study can be performed to identify whether higher-order accuracy of BSMPM combined with the cubic-spline or TLS reconstruction technique is sufficient to outperform massively parallel implementations of the existing variants of MPM (e.g., DDMPM).

- This dissertation gives an overview of existing research on the development of MPM, but does not provide a numerical comparison of the techniques. Such a study may be valuable for research groups whose main focus is on the application of MPM.

- The use of the TLS function reconstruction technique as a general technique to construct physically correct surrogate methods, outside of MPM, forms another possible research direction. For example, the reconstruction technique can be used to approximate anisotropic diffusion fields from discrete measurement data.

7

REFERENCES

[1] D. Sulsky, Z. Chen, and H. Schreyer, *A particle method for history-dependent materials,* Computer Methods in Applied Mechanics and Engineering **118**, 179 (1994).

[2] D. Sulsky, S. Zhou, and H. Schreyer, *Application of a particle-in-cell method to solid mechanics,* Computer Physics Communications **87**, 236 (1995).

[3] S. Brenner and L. Scott, *The Mathematical Theory of Finite Element Methods*, 3rd ed. (Springer, New York, 2008).

[4] S. Andersen and L. Andersen, *Modelling of landslides with the material-point method,* Computational Geosciences **14**, 137 (2010).

[5] D. Sulsky and L. Schreyer, *MPM simulation of dynamic material failure with a decohesion constitutive model,* European Journal of Mechanics-A/Solids **23**, 423 (2004).

[6] F. Zabala and E. Alonso, *Progressive failure of Aznalcóllar dam using the material point method,* Géotechnique **61**, 795 (2011).

[7] C. M. Mast, P. Arduino, P. Mackenzie-Helnwein, and G. R. Miller, *Simulating granular column collapse using the material point method,* Acta Geotechnica **10**, 101 (2015).

[8] S. Bandara, A. Ferrari, and L. Laloui, *Modelling landslides in unsaturated slopes subjected to rainfall infiltration using material point method,* International Journal for Numerical and Analytical Methods in Geomechanics **40**, 1358 (2016).

[9] B. Wang, M. Hicks, and P. Vardon, *Slope failure analysis using the random material point method,* Géotechnique letters **6**, 113 (2016).

[10] N. Phuong, A. Van Tol, A. Elkadi, and A. Rohe, *Numerical investigation of pile installation effects in sand using material point method,* Computers and Geotechnics **73**, 58 (2016).

[11] J. A. Nairn, *Material point method calculations with explicit cracks,* Computer Modeling in Engineering and Sciences **4**, 649 (2003).

[12] Y. Guo and J. Nairn, *Simulation of dynamic 3D crack propagation within the material point method,* Computer Modeling in Engineering & Sciences **113**, 389 (2017).

[13] A. Gilmanov and S. Acharya, *A hybrid immersed boundary and material point method for simulating 3D fluid–structure interaction problems,* International Journal for Numerical Methods in Fluids **56**, 2151 (2008).

[14] A. R. York, D. Sulsky, and H. L. Schreyer, *Fluid–membrane interaction based on the material point method,* International Journal for Numerical Methods in Engineering **48**, 901 (2000).

[15] Y.-P. Lian, Y. Liu, and X. Zhang, *Coupling of membrane element with material point method for fluid–membrane interaction problems,* International Journal of Mechanics and Materials in Design **10**, 199 (2014).

[16] F. Hamad, Z. Więckowski, and C. Moormann, *Interaction of fluid–solid–geomembrane by the material point method,* Computers and Geotechnics **81**, 112 (2017).

[17] A. Stomakhin, C. Schroeder, L. Chai, J. Teran, and A. Selle, *A material point method for snow simulation,* ACM Transactions on Graphics (TOG) **32**, 102 (2013).

[18] D. Sulsky, H. Schreyer, K. Peterson, R. Kwok, and M. Coon, *Using the material-point method to model sea ice dynamics,* Journal of Geophysical Research: Oceans **112** (2007).

[19] J. Gaume, T. Gast, J. Teran, A. van Herwijnen, and C. Jiang, *Dynamic anticrack propagation in snow,* Nature Communications **9**, 3047 (2018).

[20] S. Ma, X. Zhang, Y. Lian, and X. Zhou, *Simulation of high explosive explosion using adaptive material point method,* Computer Modeling in Engineering and Sciences (CMES) **39**, 101 (2009).

[21] T. R. Dhakal and D. Z. Zhang, *Material point methods applied to one-dimensional shock waves and dual domain material point method with sub-points,* Journal of Computational Physics **325**, 301 (2016).

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

[23] I. Ionescu, J. E. Guilkey, M. Berzins, R. M. Kirby, and J. A. Weiss, *Simulation of soft tissue failure using the material point method,* Journal of Biomechanical Engineering **128**, 917 (2006).

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

[25] D. Sulsky and M. Gong, *Improving the Material-Point Method,* Lecture Notes in Applied and Computational Mechanics, Vol. 81 (Springer, Cham, 2016).

[26] M. Gong, *Improving the material point method,* Ph.D. thesis, Albuquerque, NM, USA (2015).

[27] E. Love and D. Sulsky, *An unconditionally stable, energy–momentum consistent implementation of the material-point method,* Computer Methods in Applied Mechanics and Engineering **195**, 3903 (2006).

[28] M. Steffen, R. M. Kirby, and M. Berzins, *Decoupling and balancing of space and time errors in the material point method (MPM),* International Journal for Numerical Methods in Engineering **82**, 1207 (2010).

[29] A. Sadeghirad, R. M. Brannon, and J. Burghardt, *A convected particle domain interpolation technique to extend applicability of the material point method for problems involving massive deformations,* International Journal for Numerical Methods in Engineering **86**, 1435 (2011).

[30] D. Z. Zhang, X. Ma, and P. T. Giguere, *Material point method enhanced by modified gradient of shape function,* Journal of Computational Physics **230**, 6379 (2011).

[31] M. Steffen, R. M. Kirby, and M. Berzins, *Analysis and reduction of quadrature errors in the material point method (MPM),* International Journal for Numerical Methods in Engineering **76**, 922 (2008).

[32] B. Li, F. Habbbal, and M. Ortiz, *Optimal transportation meshfree approximation schemes for fluid and plastic flows,* International Journal for Numerical Methods in Engineering **83**, 1541 (2010).

[33] B. J. Alder and T. E. Wainwright, *Studies in molecular dynamics. I. General method,* The Journal of Chemical Physics **31**, 459 (1959).

[34] A. Rahman, *Correlations in the motion of atoms in liquid argon,* Physical Review **136**, A405 (1964).

[35] P. A. Cundall and O. D. Strack, *A discrete numerical model for granular assemblies,* Géotechnique **29**, 47 (1979).

[36] L. E. Malvern, *Introduction to the Mechanics of a Continuous Medium*, Monograph (Prentice-Hall, Inc., Englewood Cliffs, NJ, USA, 1969).

[37] I. Kafaji, *Formulation of a dynamic material point method (MPM) for geomechanical problems*, Ph.D. thesis, Stuttgart, Germany (2013).

[38] J. Donea, A. Huerta, J. Ponthot, and A. Rodrıguez-Ferran, *Arbitrary Lagrangian–Eulerian methods,* in *Encyclopedia of Computational Mechanics*, edited by R. Stein, R. de Borst, and T. Hughes (Wiley, Chichester, UK, 2004).

[39] C. W. Hirt, A. A. Amsden, and J. Cook, *An arbitrary Lagrangian-Eulerian computing method for all flow speeds,* Journal of Computational Physics **14**, 227 (1974).

[40] W. F. Noh, *CEL: A time-dependent, two-space-dimensional, coupled Eulerian-Lagrange code*, Tech. Rep. (Lawrence Radiation Lab., Univ. of California, Livermore, 1963).

[41] J. Trepanier, M. Reggio, H. Zhang, and R. Camarero, *A finite-volume method for the Euler equations on arbitrary Lagrangian-Eulerian grids,* Computers & Fluids **20**, 399 (1991).

[42] T. J. Hughes, W. K. Liu, and T. K. Zimmermann, *Lagrangian-Eulerian finite element formulation for incompressible viscous flows,* Computer Methods in Applied Mechanics and Engineering **29**, 329 (1981).

[43] S. Li and W. K. Liu, *Meshfree and particle methods and their applications,* Applied Mechanics Reviews **55**, 1 (2002).

[44] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl, *Meshless methods: an overview and recent developments,* Computer Methods in Applied Mechanics and Engineering **139**, 3 (1996).

[45] J.-S. Chen, M. Hillman, and S.-W. Chi, *Meshfree methods: progress made after 20 years,* Journal of Engineering Mechanics **143**, 04017001 (2017).

[46] L. B. Lucy, *A numerical approach to the testing of the fission hypothesis,* The Astronomical Journal **82**, 1013 (1977).

[47] R. A. Gingold and J. J. Monaghan, *Smoothed particle hydrodynamics: theory and application to non-spherical stars,* Monthly Notices of the Royal Astronomical Society **181**, 375 (1977).

[48] J. J. Monaghan, *Why particle methods work,* SIAM Journal on Scientific and Statistical Computing **3**, 422 (1982).

[49] J. J. Monaghan, *An introduction to SPH,* Computer Physics Communications **48**, 89 (1988).

[50] T. Belytschko, Y. Y. Lu, and L. Gu, *Element-free Galerkin methods,* International Journal for Numerical Methods in Engineering **37**, 229 (1994).

[51] T. Belytschko, Y. Lu, and L. Gu, *Crack propagation by element-free Galerkin methods,* Engineering Fracture Mechanics **51**, 295 (1995).

[52] Y. Lu, T. Belytschko, and L. Gu, *A new implementation of the element free Galerkin method,* Computer Methods in Applied Mechanics and Engineering **113**, 397 (1994).

[53] J. M. Melenk, *On generalized finite element methods,* Ph.D. thesis, research directed by Dept. of Mathematics.University of Maryland at College Park (1995).

[54] T. Strouboulis, K. Copps, and I. Babuška, *The generalized finite element method,* Computer Methods in Applied Mechanics and Engineering **190**, 4081 (2001).

[55] 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**, 314 (1986).

[56] J. Brackbill, *The ringing instability in particle-in-cell calculations of low-speed flow,* Journal of Computational Physics **75**, 469 (1988).

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

[58] Y. N. Grigoryev, V. A. Vshivkov, and M. P. Fedoruk, *Numerical "Particle-In-Cell" Methods: Theory and Applications* (Walter de Gruyter, Zeist, Netherlands, 2012).

[59] J. Brackbill, *FLIP MHD: A particle-in-cell method for magnetohydrodynamics,* Journal of Computational Physics **96**, 163 (1991).

[60] Y. Zhu and R. Bridson, *Animating sand as a fluid,* in *Proceeding SIGGRAPH '05 ACM SIGGRAPH 2005 Papers,* edited by M. Gross (ACM Press, 2005) pp. 965–972.

[61] C. Jiang, C. Schroeder, A. Selle, J. Teran, and A. Stomakhin, *The affine particle-in-cell method,* ACM Transactions on Graphics (TOG) **34**, 51 (2015).

[62] J. Brackbill, *Particle methods,* International Journal for Numerical Methods in Fluids **47**, 693 (2005).

[63] T. Haugbølle, J. T. Frederiksen, and Å. Nordlund, *Photon-plasma: A modern high-order particle-in-cell code,* Physics of Plasmas **20**, 062904 (2013).

[64] A. J. M. Spencer, *Continuum Mechanics* (Courier Corporation, 2004).

[65] K. Abe, K. Soga, and S. Bandara, *Material point method for coupled hydromechanical problems,* Journal of Geotechnical and Geoenvironmental Engineering **140**, 04013033 (2013).

[66] S. Bandara and K. Soga, *Coupling of soil deformation and pore fluid flow using material point method,* Computers and Geotechnics **63**, 199 (2015).

[67] A. Yerro, E. Alonso, and N. Pinyol, *Run-out of landslides in brittle soils,* Computers and Geotechnics **80**, 427 (2016).

[68] W. Sołowski and S. Sloan, *Evaluation of material point method for use in geotechnics,* International Journal for Numerical and Analytical Methods in Geomechanics **39**, 685 (2015).

[69] A. Cromer, *Stable solutions using the Euler approximation,* American Journal of Physics **49**, 455 (1981).

[70] B. Ydri, A. Bouchareb, and R. Chemam, *Lectures on Computational Physics* (Physics Department, Badji Mokhtar University, Annaba, Algeria, 2013).

[71] M. Mieremet, D. Stolle, F. Ceccato, and C. Vuik, *Numerical stability for modelling of dynamic two-phase interaction,* International Journal for Numerical and Analytical Methods in Geomechanics **40**, 1284 (2016).

[72] S. Bardenhagen, *Energy conservation error in the material point method for solid mechanics,* Journal of Computational Physics **180**, 383 (2002).

[73] E. Love and D. Sulsky, *An energy-consistent material-point method for dynamic finite deformation plasticity,* International Journal for Numerical Methods in Engineering **65**, 1608 (2006).

[74] H. Zhang, K. Wang, and Z. Chen, *Material point method for dynamic analysis of saturated porous media under external contact/impact of solid bodies,* Computer Methods in Applied Mechanics and Engineering **198**, 1456 (2009).

[75] I. Jassim, D. Stolle, and P. Vermeer, *Two-phase dynamic analysis by material point method,* International Journal for Numerical and Analytical Methods in Geomechanics **37**, 2502 (2013).

[76] A. Yerro Colom, E. Alonso Pérez de Agreda, and N. M. Pinyol Puigmartí, *The material point method for unsaturated soils,* Géotechnique **65**, 201 (2015).

[77] K. Soga, E. Alonso, A. Yerro, K. Kumar, and S. Bandara, *Trends in large-deformation analysis of landslide mass movements with particular emphasis on the material point method,* Géotechnique **66**, 248 (2015).

[78] H. Zhang, K. Wang, and Z. Zhang, *Material point method for numerical simulation of failure phenomena in multiphase porous media,* in *Proceedings of the 2007 International Symposium on Computational Mechanics in Beijing,* edited by Z. Yao and M. Yuan (Springer Science & Business Media, 2007) pp. 36–47.

[79] W. K. Shin, *Numerical simulation of landslides and debris flows using an enhanced material point method,* Ph.D. thesis, Seattle, WA, USA (2009).

[80] P. Mackenzie-Helnwein, P. Arduino, W. Shin, J. Moore, and G. Miller, *Modeling strategies for multiphase drag interactions using the material point method,* International Journal for Numerical Methods in Engineering **83**, 295 (2010).

[81] H. Schreyer, D. Sulsky, and S.-J. Zhou, *Modeling delamination as a strong discontinuity with the material point method,* Computer Methods in Applied Mechanics and Engineering **191**, 2483 (2002).

[82] Z. Więckowski, *The material point method in large strain engineering problems,* Computer Methods in Applied Mechanics and Engineering **193**, 4417 (2004).

[83] X. Zhang, K. Sze, and S. Ma, *An explicit material point finite element method for hyper-velocity impact,* International Journal for Numerical Methods in Engineering **66**, 689 (2006).

[84] P. C. Wallstedt and J. Guilkey, *An evaluation of explicit time integration schemes for use with the generalized interpolation material point method,* Journal of Computational Physics **227**, 9628 (2008).

[85] J. E. Guilkey and J. A. Weiss, *Implicit time integration for the material point method: Quantitative and algorithmic comparisons with the finite element method,* International Journal for Numerical Methods in Engineering **57**, 1323 (2003).

[86] D. Sulsky and A. Kaul, *Implicit dynamics in the material-point method,* Computer Methods in Applied Mechanics and Engineering **193**, 1137 (2004).

[87] L. Beuth, Z. Więckowski, and P. Vermeer, *Solution of quasi-static large-strain problems by the material point method,* International Journal for Numerical and Analytical Methods in Geomechanics **35**, 1451 (2011).

[88] J. Sanchez, H. Schreyer, D. Sulsky, and P. Wallstedt, *Solving quasi-static equations with the material-point method,* International Journal for Numerical Methods in Engineering **103**, 60 (2015).

[89] B. Wang, P. J. Vardon, M. A. Hicks, and Z. Chen, *Development of an implicit material point method for geotechnical applications,* Computers and Geotechnics **71**, 159 (2016).

[90] J. Ma, H. Lu, and R. Komanduri, *Structured mesh refinement in generalized interpolation material point (GIMP) method for simulation of dynamic problems,* Computer Modeling in Engineering and Sciences **12**, 213 (2006).

[91] A. Brydon, S. Bardenhagen, E. Miller, and G. Seidler, *Simulation of the densification of real open-celled foam microstructures,* Journal of the Mechanics and Physics of Solids **53**, 2638 (2005).

[92] N. P. Daphalapurkar, H. Lu, D. Coker, and R. Komanduri, *Simulation of dynamic crack growth using the generalized interpolation material point (GIMP) method,* International Journal of Fracture **143**, 79 (2007).

[93] C. Liu, Q. Sun, F. Jin, and G. G. Zhou, *A fully coupled hydro-mechanical material point method for saturated dense granular materials,* Powder Technology **314**, 110 (2017).

[94] A. Sadeghirad, R. M. Brannon, and J. Guilkey, *Second-order convected particle domain interpolation (CPDI2) with enrichment for weak discontinuities at material interfaces,* International Journal for Numerical Methods in Engineering **95**, 928 (2013).

[95] V. P. Nguyen, C. T. Nguyen, T. Rabczuk, and S. Natarajan, *On a family of convected particle domain interpolations in the material point method,* Finite Elements in Analysis and Design **126**, 50 (2017).

[96] R. Leavy, J. Guilkey, B. Phung, A. Spear, and R. Brannon, *A convected-particle tetrahedron interpolation technique in the material-point method for the mesoscale modeling of ceramics,* Computational Mechanics (2019).

[97] F. Hamad, *Formulation of the axisymmetric CPDI with application to pile driving in sand,* Computers and Geotechnics **74**, 141 (2016).

[98] Y. Zheng, F. Gao, H. Zhang, and M. Lu, *Improved convected particle domain interpolation method for coupled dynamic analysis of fully saturated porous media involving large deformation,* Computer Methods in Applied Mechanics and Engineering **257**, 150 (2013).

[99] Y. Gan, Z. Sun, Z. Chen, X. Zhang, and Y. Liu, *Enhancement of the material point method using B-spline basis functions,* International Journal for Numerical Methods in Engineering **113**, 411 (2018).

[100] C. Long, D. Zhang, C. Bronkhorst, and G. Gray III, *Representing ductile damage with the dual domain material point method,* Computer Methods in Applied Mechanics and Engineering **300**, 611 (2016).

[101] A. P. Tampubolon, T. Gast, G. Klár, C. Fu, J. Teran, C. Jiang, and K. Museth, *Multi-species simulation of porous sand and water mixtures,* ACM Transactions on Graphics (TOG) **36**, 105 (2017).

[102] D. Z. Zhang, Q. Zou, W. B. VanderHeyden, and X. Ma, *Material point method applied to multiphase flows,* Journal of Computational Physics **227**, 3159 (2008).

[103] C. Gritton and M. Berzins, *Improving accuracy in the MPM method using a null space filter,* Computational Particle Mechanics **4**, 131 (2017).

[104] M. Cortis, W. Coombs, C. Augarde, M. Brown, A. Brennan, and S. Robinson, *Imposition of essential boundary conditions in the material point method,* International Journal for Numerical Methods in Engineering **113**, 130 (2018).

[105] Y. Bing, M. Cortis, T. Charlton, W. Coombs, and C. Augarde, *B-spline based boundary conditions in the material point method,* Computers & Structures **212**, 257 (2019).

[106] W. M. Coombs, T. J. Charlton, M. Cortis, and C. E. Augarde, *Overcoming volumetric locking in material point methods,* Computer Methods in Applied Mechanics and Engineering **333**, 1 (2018).

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

[108] T. Charlton, W. Coombs, and C. Augarde, *iGIMP: An implicit generalised interpolation material point method for large deformations,* Computers & Structures **190**, 108 (2017).

[109] Y. Lian, X. Zhang, and Y. Liu, *Coupling of finite element method with material point method by local multi-mesh contact method,* Computer Methods in Applied Mechanics and Engineering **200**, 3482 (2011).

[110] Y. Lian, X. Zhang, and Y. Liu, *An adaptive finite element material point method and its application in extreme deformation problems,* Computer Methods in Applied Mechanics and Engineering **241**, 275 (2012).

[111] Z. Chen, X. Qiu, X. Zhang, and Y. Lian, *Improved coupling of finite element method with material point method based on a particle-to-surface contact algorithm,* Computer Methods in Applied Mechanics and Engineering **293**, 1 (2015).

[112] S. J. Raymond, B. Jones, and J. R. Williams, *A strategy to couple the material point method (MPM) and smoothed particle hydrodynamics (SPH) computational techniques,* Computational Particle Mechanics **5**, 49 (2018).

[113] L. He, Y. Gan, and Z. Chen, *Preliminary effort in developing the smoothed material point method for impact,* Computational Particle Mechanics **6**, 45 (2019).

[114] H. Lu, N. Daphalapurkar, B. Wang, S. Roy, and R. Komanduri, *Multiscale simulation from atomistic to continuum–coupling molecular dynamics (MD) with the material point method (MPM),* Philosophical Magazine **86**, 2971 (2006).

[115] T. R. Dhakal and D. Z. Zhang, *Combining dual domain material point method with molecular dynamics for thermodynamic nonequilibriums,* Journal of Computational Physics **374**, 984 (2018).

[116] M. Steffen, P. Wallstedt, J. Guilkey, R. Kirby, and M. Berzins, *Examination and analysis of implementation choices within the material point method (MPM),* Computer Modeling in Engineering and Sciences **32**, 107 (2008).

[117] E. Wobbes, R. Tielen, M. Möller, C. Vuik, and V. Galavi, *Comparison between Material Point Method and meshfree schemes derived from optimal transportation theory,* in *Proceedings of the 2nd International Conference on the Material Point Method for Modelling Soil-Water-Structure Interaction,* edited by D. Liang, K. Kumar, and R. Alexander (China Ocean Press, 2019) pp. 173–178.

[118] P. De Koster, R. Tielen, E. Wobbes, and M. Möller, *Comparison and unification of material point and optimal transportation meshfree methods (submitted),* (submitted) .

[119] T. J. Hughes, J. A. Cottrell, and Y. Bazilevs, *Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement,* Computer Methods in Applied Mechanics and Engineering **194**, 4135 (2005).

[120] J. A. Samareh, *Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization,* AIAA Journal **39**, 877 (2001).

[121] I. Ainsworth, M. Ristic, and D. Brujic, *CAD-based measurement path planning for free-form shapes using contact probes,* The International Journal of Advanced Manufacturing Technology **16**, 23 (2000).

[122] Y. Bazilevs, V. M. Calo, Y. Zhang, and T. J. Hughes, *Isogeometric fluid–structure interaction analysis with applications to arterial blood flow,* Computational Mechanics **38**, 310 (2006).

[123] L. De Lorenzis, I. Temizer, P. Wriggers, and G. Zavarise, *A large deformation frictional contact formulation using nurbs-based isogeometric analysis,* International Journal for Numerical Methods in Engineering **87**, 1278 (2011).

[124] C. De Boor, *A practical guide to splines. Applied Mathematical Sciences* (Springer, New York, NY, USA, 2001).

[125] D. Shepard, *A two-dimensional interpolation function for irregularly-spaced data,* in *Proceedings of the 1968 23rd ACM National Conference,* Proceedings of the 1968 23rd ACM National Conference, edited by R. B. Blue (ACM Digital Library, 1968) pp. 517–524.

[126] C. Vuik, P. Van Beek, F. Vermolen, and J. Van Kan, *Numerical Methods for Ordinary differential equations* (VSSD, Delft, Netherlands, 2007).

[127] A. Nealen, *An as-short-as-possible introduction to the least squares, weighted least squares and moving least squares methods for scattered data approximation and interpolation,* URL: http://www.nealen.com/projects **130**, 25 (2004).

[128] Y. Bazilevs, G. Moutsanidis, J. Bueno, K. Kamran, D. Kamensky, M. C. Hillman, H. Gomez, and J. S. Chen, *A new formulation for air-blast fluid–structure interaction using an immersed approach: Part II—coupling of IgA and meshfree discretizations,* Computational Mechanics **60**, 101 (2017).

[129] Y. G. Motlagh and W. M. Coombs, *An implicit high-order material point method,* Procedia Engineering **175**, 8 (2017).

[130] E. Wobbes, M. Möller, V. Galavi, and C. Vuik, *Conservative Taylor least squares reconstruction with application to material point methods,* International Journal for Numerical Methods in Engineering **117**, 271 (2019).

[131] M. J. Powell and M. A. Sabin, *Piecewise quadratic approximations on triangles,* ACM Transactions on Mathematical Software (TOMS) **3**, 316 (1977).

[132] H. Speleers, C. Manni, F. Pelosi, and M. L. Sampoli, *Isogeometric analysis with Powell–Sabin splines for advection–diffusion–reaction problems,* Computer Methods in Applied Mechanics and Engineering **221**, 132 (2012).

[133] P. Dierckx, S. Van Leemput, and T. Vermeire, *Algorithms for surface fitting using Powell-Sabin splines,* IMA Journal of Numerical Analysis **12**, 271 (1992).

[134] C. Manni and P. Sablonniere, *Quadratic spline quasi-interpolants on Powell-Sabin partitions,* Advances in Computational Mathematics **26**, 283 (2007).

[135] P. Sablonnière, *Error bounds for Hermite interpolation by quadratic splines on an α-triangulation,* IMA Journal of Numerical Analysis **7**, 495 (1987).

[136] T. Nguyen and J. Peters, *Refinable C^1 spline elements for irregular quad layout,* Computer Aided Geometric Design **43**, 123 (2016).

[137] P. Dierckx, *On calculating normalized Powell-Sabin B-splines,* Computer Aided Geometric Design **15**, 61 (1997).

[138] E. Wobbes, R. Tielen, M. Möller, and C. Vuik, *Comparison and unification of material point and optimal transportation meshfree methods (submitted),* (submitted) .

[139] B. Li, A. Kidane, G. Ravichandran, and M. Ortiz, *Verification and validation of the Optimal Transportation Meshfree (OTM) simulation of terminal ballistics,* International Journal of Impact Engineering **42**, 25 (2012).

[140] B. Li, A. Pandolfi, and M. Ortiz, *Material-point erosion simulation of dynamic fragmentation of metals,* Mechanics of Materials **80**, 288 (2015).

[141] L. Fedeli, A. Pandolfi, and M. Ortiz, *Geometrically exact time-integration meshfree schemes for advection-diffusion problems derived from optimal transportation theory and their connection with particle methods,* International Journal for Numerical Methods in Engineering **112**, 1175 (2017).

[142] P. Navas, S. López Querol, R. Yu, and M. Pastor, *Optimal Transportation Meshfree method in geotechnical engineering problems under large deformation regime,* International Journal for Numerical Methods in Engineering **115**, 1217 (2018).

[143] C. Villani, *Topics in Optimal Transportation Theory*, Vol. 58 (American Mathematical Society, Providence, Rhode Iland, USA, 2003).

[144] N. Sukumar, *Construction of polygonal interpolants: a maximum entropy approach,* International Journal for Numerical Methods in Engineering **61**, 2159 (2004).

[145] M. Arroyo and M. Ortiz, *Local maximum-entropy approximation schemes: a seamless bridge between finite elements and meshfree methods,* International Journal for Numerical Methods in Engineering **65**, 2167 (2006).

[146] C. Weißenfels and P. Wriggers, *Stabilization algorithm for the optimal transportation meshfree approximation scheme,* Computer Methods in Applied Mechanics and Engineering **329**, 421 (2018).

[147] C. Cyron, M. Arroyo, and M. Ortiz, *Smooth, second order, non-negative meshfree approximants selected by maximum entropy,* International Journal for Numerical Methods in Engineering **79**, 1605 (2009).

[148] J.-D. Benamou and Y. Brenier, *A numerical method for the optimal time-continuous mass transport problem and related problems,* Contemporary Mathematics **226**, 1 (1999).

[149] R. McCann, *A convexity principle for interacting gases,* Advances in Mathematics **128**, 153 (1997).

[150] W. R. Hamilton, *XV. On a general method in dynamics,* Philosophical Transactions of the Royal Society of London **124**, 247 (1834).

[151] F. Habbal, *The optimal transportation meshfree method for general fluid flows and strongly coupled fluid-structure interaction problems*, Ph.D. thesis, Pasadena, CA, USA (2013).

[152] L. Fan, W. Coombs, and C. Augarde, *The point collocation method with a local maximum entropy approach,* Computers & Structures **201**, 1 (2018).

[153] J. Hale and P. Baiz, *A locking-free meshfree method for the simulation of shear-deformable plates based on a mixed variational formulation,* Computer Methods in Applied Mechanics and Engineering **241**, 311 (2012).

[154] D. Millán, A. Rosolen, and M. Arroyo, *Nonlinear manifold learning for meshfree finite deformation thin-shell analysis,* International Journal for Numerical Methods in Engineering **93**, 685 (2013).

[155] A. Ortiz, M. Puso, and N. Sukumar, *Maximum-entropy meshfree method for compressible and near-incompressible elasticity,* Computer Methods in Applied Mechanics and Engineering **199**, 1859 (2010).

[156] T. Belytschko, Y. Krongauz, J. Dolbow, and C. Gerlach, *On the completeness of meshfree particle methods,* International Journal for Numerical Methods in Engineering **43**, 785 (1998).

[157] N. Agmon, Y. Alhassid, and R. D. Levine, *An algorithm for finding the distribution of maximal entropy,* Journal of Computational Physics **30**, 250 (1979).

[158] S. Boyd and L. Vandenberghe, *Convex Optimization* (Cambridge University Press, 2004).

[159] B. T. Polyak, *Newton's method and its use in optimization,* European Journal of Operational Research **181**, 1086 (2007).

[160] L. Yaw, N. Sukumar, and S. Kunnath, *Meshfree co-rotational formulation for two-dimensional continua,* International Journal for Numerical Methods in Engineering **79**, 979 (2009).

[161] N. Sukumar and R. Wright, *Overview and construction of meshfree basis functions: from moving least squares to entropy approximants,* International Journal for Numerical Methods in Engineering **70**, 181 (2007).

[162] *Maxent basis functions for Julia, MATLAB and C++,* http://camlab.cl/software/maxent/, accessed: 16-05-2019.

[163] J. Guilkey, J. Hoying, and J. Weiss, *Computational modeling of multicellular constructs with the material point method,* Journal of Biomechanics **39**, 2074 (2006).

[164] L. Zhang, *Dynamic description of texture evolution in polycrystalline nickel under mechanical loading with elastic and plastic deformation via Monte Carlo and Material Point Method simulation,* Ph.D. thesis, Colorado School of Mines (2008).

[165] B. Li, M. Stalzer, and M. Ortiz, *A massively parallel implementation of the Optimal Transportation Meshfree method for explicit solid dynamics,* International Journal for Numerical Methods in Engineering **100**, 40 (2014).

[166] T. Erhart, W. A. Wall, and E. Ramm, *Robust adaptive remeshing strategy for large deformation, transient impact simulations,* International Journal for Numerical Methods in Engineering **65**, 2139 (2006).

[167] P. Thoutireddy and M. Ortiz, *A variational r-adaption and shape-optimization method for finite-deformation elasticity,* International Journal for Numerical Methods in Engineering **61**, 1 (2004).

[168] J. Mosler and M. Ortiz, *On the numerical implementation of variational arbitrary Lagrangian–Eulerian (VALE) formulations,* International Journal for Numerical Methods in Engineering **67**, 1272 (2006).

[169] M. Zielonka, M. Ortiz, and J. Marsden, *Variational r-adaption in elastodynamics,* International Journal for Numerical Methods in Engineering **74**, 1162 (2008).

[170] E. D. Wobbes, M. Möller, V. Galavi, and C. Vuik, *Taylor least squares reconstruction technique for material point methods,* in *Proceedings of the 6th. European Conference on Computational Mechanics (Solids, Structures and Coupled Problems), 7th. European Conference on Computational Fluid Dynamics*, edited by R. Owen, R. De Borst, J. Reese, and C. Pearce (International Center for Numerical Methods in Engineering, 2018) pp. 806–817.

[171] A. Legendre, *Nouvelles méthodes pour la détermination des orbites des comètes* (F. Didot, 1805).

[172] P. Lancaster and K. Salkauskas, *Surfaces generated by moving least squares methods,* Mathematics of Computation **37**, 141 (1981).

[173] T. H. Wonnacott and R. J. Wonnacott, *Introductory Statistics*, Vol. 5 (Wiley New York, 1990).

[174] G. E. Box, J. S. Hunter, and W. G. Hunter, *Statistics for experimenters,* in *Wiley Series in Probability and Statistics* (Wiley Hoboken, NJ, 2005).

[175] C. Bajaj, I. Ihm, and J. Warren, *Higher-order interpolation and least-squares approximation using implicit algebraic surfaces,* ACM Transactions on Graphics (TOG) **12**, 327 (1993).

[176] V. Pratt, *Direct least-squares fitting of algebraic surfaces,* in *SIGGRAPH '87 Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, edited by M. C. Stone (ACM Press, 1987) pp. 145–152.

[177] M. Unser, A. Aldroubi, and M. Eden, *Enlargement or reduction of digital images with minimum loss of information,* IEEE Transactions on Image Processing **4**, 247 (1995).

[178] N. Petrovskaya, *The accuracy of least-squares approximation on highly stretched meshes,* International Journal of Computational Methods **5**, 449 (2008).

[179] S. N. Atluri and S. Shen, *The basis of meshless domain discretization: the meshless local Petrov–Galerkin (MLPG) method,* Advances in Computational Mathematics **23**, 73 (2005).

[180] G.-R. Liu and Y.-T. Gu, *An introduction to meshfree methods and their programming* (Springer Science & Business Media, Dordrecht, Netherlands, 2005).

[181] T. Sonar, *Difference operators from interpolating moving least squares and their deviation from optimality,* ESAIM: Mathematical Modelling and Numerical Analysis **39**, 883 (2005).

[182] H. Luo, J. Baum, and R. Löhner, *A discontinuous Galerkin method based on a Taylor basis for the compressible flows on arbitrary grids,* Journal of Computational Physics **227**, 8875 (2008).

[183] K. Bathe, E. Ramm, and E. Wilson, *Finite element formulations for large deformation dynamics analysis,* International Journal for Numerical Methods in Engineering **9**, 353 (1975).

[184] J. Burkardt, *Computational Geometry Lab: Mapping triangles* (Virginia Polytechnic Institute and State University, Blacksburg, VA, USA, 2010).

ACKNOWLEDGEMENTS

To my copromotor, Matthias Möller, I owe the greatest debt for making this dissertation possible. What started for both of us as a voluntary collaboration to support a Master's project has entirely changed my view on MPM and resulted in fruitful and joyful research. I am very grateful to Matthias for supervising my scientific studies during the last years of my PhD, sharing his elegant ideas, and critically reviewing my papers and thesis. Matthias is not only an exceptionally talented scientist with a broad range of interests, but also a great mentor and friend. I sincerely appreciate his patience, encouragement, and support throughout my PhD.

I am also very grateful to my promotor, Kees Vuik. I would like to thank him for giving me the opportunity to work on the MPM-Flow project. Despite the fact that my research did not completely match his area of expertise, Kees was always interested in my progress and kept me conscious of the mathematical implications of different engineering approaches. His wise suggestions have helped me with many aspects of my PhD project.

Furthermore, I would like to thank the members of my doctoral committee, Prof. Charles Augarde, Prof. Anna Marina Pandolfi, Prof. Martin Berzins, Prof. Michael Hicks, and Prof. Arnold Heemink, for reading this dissertation and taking part in the defense. I would also like to acknowledge my paranymphs, Zaza van der Have and Joana Silva.

I want to express my gratitude to the participants of the MPM-Flow project and the members of the MPM group at Deltares for sharing their expertise. In addition, I am grateful to my first daily supervisor, Lars Beuth. He provided me with the hands-on experience at Deltares but left the company after the first two years of my PhD.

Moreover, I am very grateful to DUO-IP and especially Erik Fleur and Anton Bal for making the transition from the last stages of my PhD project to my new job easy and smooth.

During my PhD research, I had the pleasure to collaborate with a number of scientists from different parts of the world. One of them is Dieter Stolle, who I would like to thank for counseling me on geotechnics and in particular continuum mechanics. Furthermore, I am grateful to Jeremiah Brackbill for sharing his research on B-spline MPM. I would also like to express my gratitude to Quoc Anh Tran and Wojciech Solowski for many interesting discussions and providing me with an opportunity to visit Aalto University.

A special thank you goes to my colleagues Roel Tielen and Pascal de Koster. Both of them are very talented and hard working students. It was a pleasure to develop new ideas and have in-depth (and sometimes very lengthy) discussions together with Roel and Pascal. I also truly appreciate the effort they had put in writing our joint papers.

Domenico Lahaye supervised my Bacherlor's project and several other projects. To him I owe my interest in scientific research. In addition, I am grateful to Domenico for

our wide-ranging conversations during my PhD and assuring that I could attend my all-time favorite conference, the 13th Copper Mountain Conference. Taking care of administrative tasks at the university would be much harder without the support of Deborah. Her help is truly appreciated. To the SIAM student chapter board 2015-2016 as well as all my colleagues from the Numerical Analysis group, I am grateful for the nice moments we shared. In particular, I would like to express a warm thanks to my office mates throughout the years: Fei, Yue, Jochen, Virginia, Roel, Mousa, Jiao, Shuaiqiang, Luis, Anne, and Thomas.

My PhD would be a much duller experience without Zaza. She has been an amazing friend, who I could always count on, and has provided me with another beautiful friend, Vivienne. I immensely enjoy spending time with both of them. The regular meetings with Miriam have also brighten up my PhD years. I enjoy contributing to Miriam's knowledge of Russian language and extremely value the honesty and depth of our conversations. Yoeri, Yaroslava, and their son, Ilya, are a wonderful trio, I have had the pleasure to spend my time with as well. The lunch and tea breaks with Yoeri had become an essential part of my routine towards the end of my project and I hope we can find a way to continue this tradition despite our new busy schedules. My colleague and friend, Joana, entered the MPM-Flow project approximately a year after me and, despite different research focuses, it was very easy to find common ground with her. Moreover, I am grateful to Reinaldo, Chiara, Dave, Tugce, and Niels for making my PhD experience more cheerful and interesting.

Finally, I would like to thank my family. To my family-in-law, I am indebted for their open-mindedness, encouragement, and love. I really appreciate the fact that they decided to travel more than 6000 km from the United States to the Netherlands to support me during the defense. Of course, special thanks goes to Kate for designing the cover of the thesis. I am also happy that my aunt, uncle, and cousin had found time to visit me in the Netherlands during the time of my PhD. Moreover, I would like to thank Gosha for staying with me in Valencia after the Congress on Numerical Methods in Engineering. There are not enough words to describe how grateful I am to my mother. Without her hard work, strength, and resilience not just my PhD project, but any of my studies in the Netherlands would not be possible. Her endless and unconditional love and trust have been a major source of optimism and motivation for me.

Last but definitely not least, I would like to express my profound gratitude to my husband, Scott, for all his support, patience, and positivity during my PhD project. Meeting him at Copper Mountain, where I had a horrible mountain sickness, was the absolute best part of my studies. Scott had to make many hard and selfless decisions so I could continue working on my research. I have learned so much from him and am very proud of everything he has achieved in his short time in the Netherlands. I could not wish for a better partner in life, and I am looking froward to all the adventures ahead of us.

CURRICULUM VITÆ

Elizaveta Dmitrievna (Lisa) Wobbes was born in Ryazan, the USSR, on 20 November 1989. From 1996 until 2005 she attended Lyceum na Sobornoi in Russia. In 2005 Lisa moved to Assen, the Netherlands. There she attended the ISK (Dutch: internationale schakelklas) at Dr. Nassau College Penta and was an intern at Dr. Nassau College Quintus until 2006. Between 2006 and 2008 she received secondary education at Quintus at VWO (Dutch: voorbereidend wetenschappelijk onderwijs) level.

She began her studies in Applied Mathematics at Delft University of Technology in 2008. In 2011 Lisa obtained her BSc diploma, and in 2014 she received her MSc degree. For her Master's project, she did an internship at Schlumberger, in Abingdon, the UK. From October 2014 until April 2015, she worked as a researcher at Delft University of Technology.

Lisa started her PhD at Delft University of Technology and Deltares in April 2015. For her scientific research, she focused on algorithmic improvements of the material-point method and development of the Taylor least-squares function reconstruction technique. Besides that, Lisa assisted with teaching several Bachelor courses and supervising a Master student. Furthermore, she was the vice president of the SIAM Student Chapter from 2015 until 2016. From April until June 2018 she was visiting Aalto University, Espoo, Finland.

Since September 2019, Lisa is working as a data scientist at DUO (Dutch: Dienst Uitvoering Onderwijs), Ministry of Education, Culture, and Science, in the Hague, the Netherlands. For more information visit elizavetawobbes.com.

LIST OF PUBLICATIONS

JOURNAL ARTICLES

3. **E. D. Wobbes**, R. Tielen, M. Möller, C. Vuik, *Comparison and unification of material-point and optimal transportation meshfree methods*, submitted.

2. P. de Koster, R. Tielen, **E. D. Wobbes**, M. Möller, *Extension of B-spline Material Point Method for unstructured triangular grids using Powell-Sabin splines*, submitted.

1. **E. D. Wobbes**, M. Möller, V. Galavi, C. Vuik, *Conservative Taylor least squares reconstruction with application to material point methods*, International Journal for Numerical Methods in Engineering **117(3)**, 271–290 (2019).

BOOK CHAPTERS

1. **E. D. Wobbes**, R. Tielen, M. Möller, V. Galavi, C. Vuik, *Recent developments to improve the numerical accuracy*, The Material Point Method for Geotechnical Engineering: A Practical Guide, CRC Press (2019).

PEER REVIEWED CONFERENE PROCEEDINGS

7. Q.-A. Tran, **E. D. Wobbes**, W. Solowski, M. Möller, C. Vuik, *Moving least squares reconstruction for B-spline Material Point Method*, Proceedings of the 2nd International Conference on the Material Point Method for Modeling Soil-Water-Structure Interaction - MPM 2019, Cambridge, United Kingdom, 8–10 January, 35–41 (2019).

6. **E. D. Wobbes**, R. Tielen, M. Möller, C. Vuik, V. Galavi, *Comparison between Material Point Method and meshfree schemes derived from optimal transportation theory*, Proceedings of the 2nd International Conference on the Material Point Method for Modeling Soil-Water-Structure Interaction - MPM 2019, Cambridge, United Kingdom, 8–10 January, 173–178 (2019).

5. **E. D. Wobbes**, M. Möller, V. Galavi, C. Vuik, *Taylor least squares reconstruction technique for material point methods*, Proceedings of the 6th European Conference on Computational Mechanics - ECCM 6, 7th European Conference on Computational Fluid Dynamics - ECFD 7, Glasgow, United Kingdom, 11–15 June, 806–817 (2018).

4. **E. D. Wobbes**, C. Vuik, L. Beuth, V. Galavi, D. Stolle, *Modeling of seismic liquefaction using dynamic two-phase FEM with modified UBC3D-PLM model*, Proceedings of the Congress on Numerical Methods in Engineering - CMN 2017, Valencia, Spain, 3–5 July, 570–581(2017).

3. R. Tielen, **E. D. Wobbes**, M. Möller, L. Beuth, C. Vuik, *A high order material point method*, Procedia Engineering, 175: 265–272 (2017).

2. **E. D. Wobbes**, L. Beuth, C. Vuik, D. Stolle, *Modeling of liquefaction using two-phase FEM with UBC3D-PLM model*, Procedia Engineering, 175: 349–356 (2017).

1. T. B. Jönsthövel, A. A. Lukyanov, **E. D. Wobbes**, C. Vuik, *Monotone Non-Galerkin Algebraic Multigrid Method Applied to Reservoir Simulations*, Proceedings of the 15th European Conference on the Mathematics of Oil Recovery - ECMOR XV, Amsterdam, Netherlands, 29 August–1 September (2016).