The Material Point Method for dynamic Metal Processing

Von der Fakultät für Ingenieurwissenschaften,
Abteilung Bauwissenschaften
der Universität Duisburg-Essen
zur Erlangung des akademischen Grades

Doktor-Ingenieur

genehmigte Dissertation

von

M.Sc. Sascha Maassen

Hauptberichter:	Prof. Dr.-Ing. habil. J. Schröder
Korreferenten:	Prof. Tarek I. Zohdi, Ph.D.

Tag der Einreichung:	24. November 2021
Tag der mündlichen Prüfung:	22. März 2023

Fakultät für Ingenieurwissenschaften,
Abteilung Bauwissenschaften
der Universität Duisburg-Essen
Institut für Mechanik
Prof. Dr.-Ing. habil. J. Schröder

Berichte des Instituts für Mechanik, Universität Duisburg-Essen

Nr. 27

Herausgeber:

Prof. Dr.-Ing. habil. J. Schröder

Organisation und Verwaltung:

Prof. Dr.-Ing. habil. J. Schröder
Institut für Mechanik
Fakultät für Ingenieurwissenschaften
Abteilung Bauwissenschaften
Universität Duisburg-Essen
Universitätsstraße 15
45141 Essen
Tel.: 0201 / 183 - 2682
Fax.: 0201 / 183 - 2680

ISBN-13 978-3-9821811-3-4

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Mechanik (Abt. Bauwissenschaften, Fak. Ingenieurwissenschaften) an der Universität Duisburg-Essen zu einem großen Teil im Rahmen der durch das "Mercator Research Center Ruhr (MERCUR)" geförderte Projekt "Virtual Machining (PE-216-0024)" unter der Leitung von Prof. P. Wiederkehr.

An erster Stelle danke ich meinem geschätzten Doktorvater Prof. Jörg Schröder ohne dessen fachliche Unterstützung diese Arbeit nicht möglich gewesen wäre. Auch für unvergessliche Momente und Erfahrungen während meiner Promotionszeit möchte ich meine Dankbarkeit ausdrücken. Außerdem danke ich meinem Zweitgutachter Prof. Tarek Zohdi für die Übernahme des Korreferats meiner Promotion. Darüber hinaus gilt mein Dank dem MERCUR und allen an diesem Projekt beteiligten Personen.

Einen erweiterten Dank möchte ich an allen Weggefährten und Mitarbeiter des Instituts aussprechen. Ohne die Personen, die mich insbesondere zu Begin meiner wissenschaftlichen Arbeit begleitet haben, wäre meine akademische Laufbahn anders verlaufen. Sie sind somit ebenfalls entscheidend für die vorliegende Arbeit gewesen.

Besonderer Dank gilt meinen Freunden und Kollegen aus der brasilianischen Wissenschaftscommunity rund um den Lehrstuhl von Prof. P.M. Pimenta. Auch hier geht die Wertschätzung über die fachlichen Beiträge zu meiner wissenschaftlichen Arbeit hinaus.

Natürlich darf am Ende der Dank meiner Familie nicht fehlen - insbesondere meiner Frau Kathrin für eine Unterstützung die ich nicht mit Gold aufwiegen kann, und meiner Tochter Lea, dass sie sich Zeit gelassen hat.

Duisburg, Juni 2023 Sascha F. Maassen

Abstract

Particle methods represent a fundamental column of simulation technology. In the last decades, these methods have been used more and more frequently also in the field of engineering disciplines. Their main advantage over established simulation techniques such as the Finite Element Method (FEM) is the possibility to represent large deformations and material discontinuities, such as chip formation, particularly well.

The "Material Point Method" (MPM) is a relatively new technique, which allows solving a continuum mechanical representation of the differential equations using a background computational grid. Mechanical bodies are represented by particles. These not only represent the current deformation state, but also carry material history and material laws. Effectively, the MPM combines the advantages of an Eulerian and Lagrangian approach for greater performance in extreme deformations.

In this thesis, the MPM is presented and discussed in detail. One focus is on the implementation of the method in the `ELSE` code for the explicit solution of engineering problems. Established benchmark problems are performed and complemented to validate the presented implementation. Further developments of MPM such as "Convected-Particle-Domain-Interpolation" (CPDI), contact mechanisms to ideally stiff bodies, and a grid-shift technique are also considered.

The second focus of the work is on the analysis of highly dynamic metalworking processes. The method is used to mimic the Split-Hopkinson-Pressure-Bar (SHPB) experiment to demonstrate the applicability of the current implementation in this domain. Subtractive metalworking processes are then simulated to analyze the performance of the method, and its sensitivity to simulation parameters.

Zusammenfassung

Partikelmethoden stellen eine fundamentale Säule der Simulationstechnik dar. In den letzten Jahrzehnten wurden diese Methoden immer häufiger auch im Bereich von Ingenieurdisziplinen eingesetzt. Ihr Hauptvorteil gegenüber etablierten Simulationstechniken wie der Finite Elemente Methode (FEM) ist die Möglichkeit große Deformationen und Materialdiskontinuitäten, wie zum Beispiel bei Spanbildung, besonders gut abzubilden. Die "Material Point Method" (MPM) stellt hierbei eine relative junge Technik dar, welche mithilfe eines Hintergrund-Berechnungsgitters eine kontinuumsmechanische Darstellung der Differenzialgleichungen löst. Dabei werden mechanische Körper durch Partikel abgebildet. Diese bilden nicht nur den aktuellen Verformungszustand ab, sondern tragen auch Materialhistorie und Stoffgesetze. Effektiv kombiniert die MPM hierdurch Vorteile einer Eulerschen- sowie Lagrangschen-Betrachtungsweise und ermöglicht somit die Darstellung extremer Deformationszustände.

In dieser Arbeit wird die MPM detailliert dargelegt und diskutiert. Ein Fokus liegt in der Implementierung der Methode im `ELSE` code, zur expliziten Lösung von Ingenieursproblemen. Etablierte Benchmarkprobleme werden durchgeführt und ergänzt um die dargelegte Implementierung zu validieren. Dabei werden auch Weiterentwicklungen der MPM wie "Convected-Particle-Domain-Interpolation" (CPDI), Kontaktmechanismen zu ideal steifen Körpern und eine "grid-shift" Technik betrachtet.

Der zweite Schwerpunkt der Arbeit liegt in der Analyse von hochdynamischen Metallbearbeitungsprozessen. Die Methode wird verwendet, um das "Split-Hopkinson-Pressure-Bar" (SHPB) Experiment nachzubilden und so die Anwendbarkeit der vorliegenden Implementierung in diesem Bereich nachzuweisen. Anschließend werden subtraktive Metallbearbeitungen simuliert und die Leistungsfähigkeit der Methode, sowie ihre Sensitivität gegenüber Simulationsparametern zu analysieren.

Contents

1 Introduction, Motivation and Outline

Numerical simulations are an established tool in engineering fields for the analysis of physical problems. They are used to predict the performance of new designs on mechanical constructions and reduce the need for more expensive real experiments. Also, they allow for in situ analysis of the stresses in structures that experiments often cannot provide. Failure analysis of structures, as done, e.g., in car-crash simulations, are very much used to improve mechanical designs or layout performance boundaries for finished products. With raising computational power at low prices and performant simulation techniques, the application of numerical analysis is extended to various fields. Simulations of isolated processing steps are one of these fields, which leads to improved techniques to optimize the final product in geometry and finish while minimizing the tool wear. Especially, subtractive and additive manufacturing processes are quite challenging for established numerical tools such as the Finite Element Method (FEM) because of the topological changes and material discontinuities. With a mesh representing the physical bodies, mesh distortion at large deformations occur. New connections between individual bodies as in welding or the opposite in milling, cannot be represented by the initial meshes at all or with the cost of realistic patterns.

As an alternative simulation technique for these cases, particle methods are used. In particle methods, physical bodies are represented by a set of particles instead of meshes. The degree to which the body is approximated may vary. Some models aim to simulate the material with particles representing individual atoms or molecules, while others focus on scales where particles represent grains or are used rather as a discrete mathematical representation of an underlying continuum.

The Material Point Method (MPM) is a relatively young particle method developed by Sulsky et al. [142] and categorizes to the latter group. As such, it is well suited for the simulation of macroscopic problems with reasonable computational efforts. In MPM, particles do not interact directly with each other. Instead, a Computational Background Grid (CBG) is introduced to handle these interactions. For each time step in an MPM simulation, the algorithm demands a mapping operation from the particles that carry the material information onto the CBG, where the actual differential equations are solved. Using the CBG allows for a variety of non-local mapping operators and the usage of established techniques such as a Galerkin weak form to represent and solve the numerical problem. At the end of each time step, results from the CBG are projected to the particles and used to update their state. The CBG does not preserve the solution further.

The uniqueness to the MPM is the Lagrangian representation of matter and the computation of the solution on a temporary grid in an Eulerian sense. The advantage of material points is that convection errors are avoided. Mass is preserved perfectly. Also local variables such as material history are kept at the material points. Evaluation of constitutive equations on these data at the material points is very stable, allowing for practical application of complex material laws with minimum computational hustle.

The Eulerian perspective on the simulation space, represented by the CBG, features independence from the current topology. Hence, mesh distortion is completely avoided, and effective mesh topologies can be chosen. Usually, regular cartesian grids are chosen without compromising the results.

Due to these advantages, the MPM was chosen for the simulation of dynamic metal processing applications in the scope of the ”Virtual Machining” project, founded by the ”Stiftung Mercator” and the ”Mercator Research Center Ruhr” (PE-216-0024).

The goal of the research that leads to the present thesis was to develop a computational tool for using the MPM in the scope of highly dynamic metal processing. It was used to model physical experiments conducted in the ”Virtual Machining” project with an extended focus on orthogonal cutting. In this process, a strip is cut from a metal workpiece. The process introduces various challenges to numerical models such as large deformations, material separation, a wide band of deformation rates, and temperature dependence. In order to investigate the performance of the MPM on metal cutting, this thesis presents benchmark problems to discuss general mechanisms and problems with the MPM and presents a numerical analysis of the Split-Hopkinson-Pressure-Bar experiment as a pre step.

This thesis, shows that the MPM is in no way inferior to other particle methods and holds to its promises. The present `ELSE` code is capable of performing orthogonal cutting simulations with good results. The usage of equally grid spaced cartesian grids using linear shape functions offers unmatched computational performance regarding the mapping operation. For the first time, the influence of the position of such a grid with respect to physical space is analyzed. It is shown that in highly dynamic problems, a grid-shift technique can be employed to resolve stress waves better than a grid fixed in space. The technique improves the stability of the analysis and produces a smooth, realistic solution without introducing any more amount of computational expense. Further, localization behavior of grid-shift results is compared with MPM simulations on a CBG, which is fixed in space.

For an overview, the structure of this thesis is outlined in the following.

Chapter 2 provides a synopsis of particle methods in general to allow a better classification of the MPM. An overview of the most important and recent works is given. **Chapter 3** provides fundamentals of classical Continuum Mechanics as used for the model in this thesis. **Chapter 4** introduces the fundamentals of constitutive modeling and presents the hyperelastic and elasto-plastic constitutive relations used in the later examples. In **Chapter 5**, standard time integration techniques are shortly revisited. To allow for a better understanding of the present MPM implementation, **Chapter 6** gives a broad review on implicit and explicit solution strategies to nonlinear partial differential equations. The MPM fundamentals are discussed in **Chapter 7**. The weak form of the equation of motion is derived on the CBG and the specific implementation of the MPM as used for this thesis is given in detail. A more algorithmic view is given in **Chapter 8** alongside a presentation of the `ELSE` code design. Basic benchmark examples are used to verify the correct implementation. **Chapter 9** has a contact focus. Standard MPM contact is shown, and the frictional MPM to rigid body contact is presented. A view on MPM variants is given in **Chapter 10**. Also, problems that may occur in MPM simulations are investigated and shown in exemplary cases. The SHPB model is presented in **Chapter 11**, featuring a Johnson-Cook material law to give an example of material fitting using MPM. In **Chapter 12**, metal cutting examples are analyzed, and the grid-shift strategy is presented. **Chapter 13** concludes this thesis and gives an outline for future research on MPM in machining processes.

2 Overview to particle methods and the MPM

The main concept of numerical simulations is to discretize mathematical equations for computer systems. For physical analysis, two main approaches were followed. In an Eulerian simulation, the physical space itself is discretized, while in a Lagrangian concept, the material is discretized. For the latter one, the material is discretized using nodes that compose a mesh or by particles that concentrate this matter. These two concepts were developed not completely independent from each other but rather have several aspects in common. Also mixed approaches exist. In time, a zoo of various techniques has been created where some are so similar that it is even hard to differentiate them. In the following, a selection of particle methods is discussed. The selection by no means claims to be exhaustive. For this review, three main categories for particle methods are introduced. The first category summarizes those that do not use a mesh in their algorithms. Rather, these methods rely on direct particle-particle interactions. A second category extends the first by assigning each particle in the simulation a larger region of influence than its immediate neighboring particles. Finally, the third category incorporates meshes into their computational algorithms.

A very prominent representative of the first category is **Molecular Dynamics** (MD) which was one of the first simulation techniques used. Early works date back to the fifties, such as Alder and Wainwright [5]. As indicated by the name, the idea of MD is that particles represent the atoms or molecules of a material. As such, MD can be used to estimate macroscopical material behavior using averaging techniques. MD is usually based on the very basic equation of motion and potential-based interactions between particles and their direct neighbors. For a more comprehensive entry into MD, the interested reader is referred to Rapaport [113].

In recent years, the **Discrete Element Method** (DEM) found more attention, especially in the engineering community. Since its introduction in Cundall and Strack [34], particles in this method interact via direct contact laws. In this sense, long-range forces are not intended. An overview of possible contact interactions can be found, e.g., in Popp and Wriggers [112] or Neto et al. [104]. Contrary to MD, a DEM particle has a rotation state. Different rotational discretizations are discussed in Campello [18]. The inclusion of rotation triggered by frictional contact between particles makes this method very popular in the geomechanics engineering community, with a focus on modeling granular materials. Rock cutting simulations in two and three dimensions were done in Rojek et al. [116] and in combination with finite elements in Oñate and Rojek [107]. In Fernandes et al. [40], the authors also coupled FEM and DEM for modeling particles in a fluid flow. In DEM, it is possible to investigate the crushing of granular materials by directly splitting particles into smaller ones, as shown in Chaudry and Wriggers [24]. With the growing importance of additive manufacturing processes, DEM is often applied to model powders, e.g., in Marchais et al. [98].

The **Smoothed Particle Hydrodynamics** (SPH) scheme is a common representative of the second category. Originally, the scheme was developed for astrophysical simulations, as in Gingold and Monaghan [47]. During the last decades, it has been developed to be used in many applications as reviewed in Vacondio et al. [149]. In SPH, the particles interact with each other through an implicit influence region with compact support. The resulting field constructions are more smooth than particle methods in the first category, making this method very successful in modeling fluids. Multi fluid simulations on SPH were shown in Fürstenau et al. [44]. For the improvement of numerical stability, the kernel functions for the mathematical particle support can be altered. For example, Lahiri et al. [81] used an adaptive B-Spline kernel to stabilize SHP simulations under large tensile deformation. Efforts like this elaborate the SPH towards reliable solid mechanic modeling. Combined with its ability to model material discontinuities, its application to metal cutting in Islam et al. [67] is a natural fit. Also, applications in the opposite direction are considered. Welding, as considered in Bagheri et al. [9], and additive manufacturing models require material to change its topology to a great extend. Laser-driven liquefaction of powders at their free surfaces in additive manufacturing was studied in Meier et al. [99] and Russell et al. [117]. SPH is often used in computer graphics for modeling fluids, which introduces a high demand for performant algorithms and implementations. A GPU accelerated powder melting simulation was presented in Fürstenau et al. [45]; cutting simulations on GPUs are performed in Afrasiabi et al. [3] as examples for improvements in this area.

The relatively young **Peridynamics** (PD), proposed by Silling [128], is a particle method using an initial horizon to introduce neighborhoods. Depending on the horizon size, the neighborhoods size is scaled. Particles interact with each other through so called bonds, which are defined between each particle pair in a neighborhood. Figure 2.1 shows three horizons exemplarily, with the established bonds. In bond-based PD, the bonds act as springs, which can be released by means of damage criterion. This way, PD was first applied to crack simulations in Silling [129] on a Kalthoff-Winkler experiment and extended to more complex applications, e.g., Hu et al. [61]. Hu and Madenci [62] extended the bond-based approach to model arbitrary Poisson's ratios which was a major drawback of this variant. The bond-based PD is a proven method for realistic fracture modeling, compare Butt and Meschke [16]. However, the bond-based approach requires more efforts in developing thermodynamically consistent elaborated materials as in Javili et al. [69].

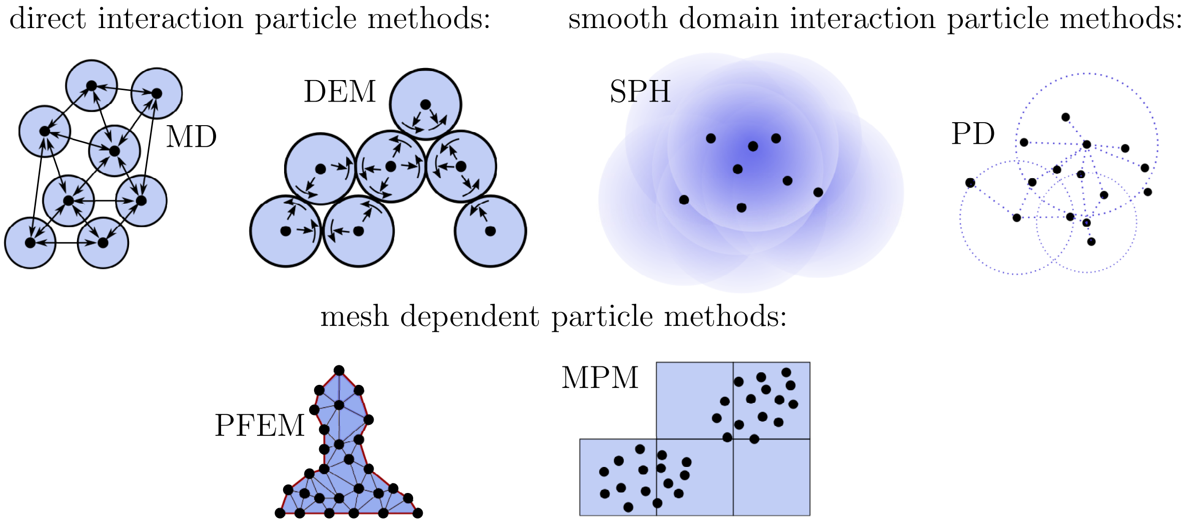

direct interaction particle methods:
smooth domain interaction particle methods:
MD
DEM
SPH
PD
mesh dependent particle methods:
PFEM
MPM

Figure 2.1: Idealized graphical illustration of the interaction mechanism of a selection of particle methods.

As an alternative to bond-based PD, the concept of state-based PD and the correspondence model was first introduced by Silling et al. [130]. The correspondence enables the reconstruction of a classical deformation gradient at each particle, which in return offers usage of constitutive formulations derived in a continuum mechanical setting. Correspondence visco-plasticity in PD was first shown in Foster et al. [43], damage models incorporated in Tupek et al. [148], and thermoplastic fractures analyzed by Amani et al. [6]. A crucial issue in correspondence formulations is the occurrence of zero-energy modes. Considerable research efforts focused on this subject and stabilized formulations as in Chowdhury et al. [29]. Based on this improvement, Bode et al. [14] showed a mixed PD formulation for effective handling of incompressibility. Also advanced applications as additive manufacturing processes are possible to model, such as in Hartmann et al. [54]. As with many other particle methods, there is an interest in coupling PD with the FEM to achieve more effective numerical models. This approach was followed, e.g. in Liu and Hong [88], and Ongaro et al. [110], to name just a few.

The third category contains particle methods that necessarily use computational meshes or grids. The **Particle Finite Element Method** (PFEM) is the youngest technique in this overview. Proposed by Idelsohn et al. [66], the idea is to use the FEM to compute the solution within a time step. The FEM mesh, however, is not fixed in time, nor is its boundary. Instead, the boundary of the considered particle cloud is updated frequently, and the FEM mesh is renewed. Consequently, the method can be used for fluid simulations where free surfaces frequently occur, e.g., in Aubry et al. [8] and for fluid-structure interaction problems, Oñate et al. [108] or Galano et al. [46]. As a result, PFEM became very popular in geomechanics for the analysis of landslides (Salazar et al. [121]) or in geotechnical engineering, see, e.g., Monforte et al. [101]. With its foundation in FEM, the method is versatile, enabling the simulation of solid mechanics with considerable topology changes, such as in cutting (Oñate et al. [109], Carbonell et al. [19]) as well. For a comprehensive review on PFEM, the interested reader is referred to, e.g., Cremonesi et al. [33].

The **Material Point Method** (MPM) also relies on a computational mesh. Originally, the successor of this method, the **Particle In Cell** (PIC) method, was proposed very early in Harlow [52] with the intention to be applied in fluid simulations. The main concept, very similar to the present MPM, was to consider a fixed Eulerian mesh to solve the differential equations and have Lagrangian particles represent the bodies that move through this grid. In PIC, the particles carry only their mass with their current position. Solution-relevant fields, e.g., velocities, are associated with the mesh.

While this was sufficient for fluid mechanical problems, Sulsky et al. [142] extended the PIC particles to carry all properties of the material and treat the Computational Background Grid (CBG) to be completely temporary within a time step. Especially the consistent preservation of material history variables, which are by default local variables, lead to the rebranding to today's MPM. From the beginning, MPM was used to simulate dynamics solid mechanical problems as in Sulsky et al. [143] or with considering an axisymmetric setting in Sulsky and Schreyer [141]. The concept of local history leads to numerically very efficient and stable conditions, even for complex constitutive formulations in MPM. To date, finite strain plasticity formulations exist that even take advantage of anti-locking techniques developed in FEM, see, e.g., Coombs et al. [31].

The MPM algorithm starts with the first mapping operation of the particle's properties onto the grid. The CBG then provides the solution for the current time step. At last, this solution is mapped back to the particles and used there to advance in time, i.e., move the particles in space. Finally, the CBG is wiped so that it does not carry permanent information. Consequently, the CBG does not accumulate deformation. Hence, even at large deformations of Lagrangian bodies, mesh distortion is avoided completely.

Usually, CBG is of cartesian geometry. The reason is the performance benefit, as it is easy to compute the enclosing grid cell for each particle. However, unstructured meshes are used frequently as well, demanding elaborated search algorithms as presented e.g., Wang et al. [152].

Since the beginning of the method, research was done on how and when to perform the mapping operations. Besides linear ansatz functions defined on cartesian or arbitrary CBGs, the Generalized Interpolation Material Point (GIMP), see Bardenhagen and Kober [11], was used to provide a higher-order interpolation scheme. Also, B-Splines defined on the CBG can be used, as shown, for example, in Tielen et al. [146], and Koster et al. [80]. Alternative mapping operators are nowadays established. A detailed review and analysis can be found in chapter 10.

Modifications on the explicit algorithm for performing the mapping operations and when to update the mapping operators where studied as well. The main considered algorithms are the Update Stress First (USF), the Update Stress Last (USL), and the Modified Update Stress Last (MUSL) schemes. In an MPM simulation, the constitutive equations are evaluated at the particles. The deformation state, however, requires information from the grid. In USF, the deformation state used for updating the stresses is computed at the beginning of the time step. Therefore, the mapping operators involved, were only updated with the current position of the particle. Contrary, the USL scheme demands the stress update at the end of a time step. Here, the utilized mapping operators still refer to the beginning of the time step, even though the position of the particle has changed. The MUSL scheme intends to provide better accuracy in time by introducing an additional mapping and solution circle for the velocity before updating the stress at the end of the time step, compare Zhang et al. [163].

A comparison of the USF and USL algorithms with respect to energy conservation was made in Bardenhagen [10]. The investigations denoted that both algorithms provide fair energy conservation in ranges where the computational grid spacing is able to resolve the structural sound waves. However, it is interesting that while the USF algorithm conserves the energy well, the USL behaves dissipative. Nevertheless, the author considered this behavior preferable in some cases to improve the stability of the simulation. In USL, especially the high frequencies are truncated, which usually cause instabilities in time discretization. To avoid spurious damping effects, Hammerquist and Nairn [51] proposed an alternative update scheme.

Earlier works on the MPM make use of explicit solution schemes. In this framework, the nodal accelerations are directly computed by dividing nodal forces with their associated masses. If, however, a nodal mass is very small, which frequently happens when a material point is near the borders of a cell, this causes instabilities or even crashes the simulation. To address this issue, commonly, a mass-cuttoff value is used to check for small nodal masses for the cost of numerical dissipation. Another approach was chosen by Ma et al. [92], where the authors applied correction factors to the computation of nodal forces without losing linear momentum. An alternative to explicit solution schemes, implicit schemes are introduced in Sulsky and Kaul [140]. However, dynamics or pseudo-dynamic load steps are still used to take advantage of the convective phase, where material points move. The main advantage of these strategies is to utilize larger time steps and achieve actual equilibrium solutions. In Charlton et al. [23], the authors presented an implicit MPM algorithm for GIMP and in Wang et al. [151] in combination with Convected Particle Domain Interpolation (CPDI). B-Splines in an implicit MPM solution scheme were used by Yamaguchi et al. [160]. A discussion on the frame of reference in implicit MPM and their consequences with respect to the used mapping functions is given in Coombs et al. [32].

The MPM is a very versatile numerical tool and has been applied to various problems. Similar to other particle methods, it can be used in fracture mechanics, see Raymond et al. [114] and in combination with an XFEM approach on the CBG in Liang et al. [86]. Also, the geoengineering community takes advantage of MPM for modeling granular soils, as in Cecchi and Rizzi [21], Ceccato and Simonini [20], or Li et al. [83], to name just a few. Investigations of water-saturated soils were done in Mroginski et al. [102] and Lei et al. [82]. A whole Fluid-Structure Interaction (FSI) problem using MPM for fluid and the solid material was shown in Su et al. [139]. An attempt to simulate sea ice behavior was taken in Sulsky et al. [144]. Stomakhin et al. [137; 138] used the MPM to model snow and melting processes. In mechanical engineering, the MPM was applied in an additive manufacturing context with laser sintering in Maeshima et al. [95] or a subtractive manufacturing context of orthogonal cutting in Nairn [103]. Various impact problems can be found in Love and Sulsky [90], Zhang et al. [162], Huang et al. [64], and Lian et al. [84; 85].

For all mentioned particle methods, it is complicated to deal with boundary conditions. With the particles representing the deformable bodies, there is no explicit boundary given. In Sandim et al. [122], a boundary detection algorithm is presented to detect those particles that lie on the virtual boundary of a body. A different approach especially focusing on boundary conditions on the implied surface of an MPM discretized body, was chosen in Liu and Sun [87] in terms of the shifted domain theory. In PFEM, the boundary must be detected for the mesh generation anyway, usually following an alpha-shape concept, see Edelsbrunner and Mücke [39]. Nevertheless, many physical problems can be simulated without boundary conditions, or using contact mechanics, e.g., with rigid surfaces. Details about contact in MPM are discussed in chapter 9.

Other physical problems involve thermal loads, e.g., for laser melting where Zohdi [166] and Wessels et al. [153] show modeling examples with a focus on particles. For the initial particle discretization of a body, multiple possibilities exist. For example, Liu and Sun [87] report a technique to use image data as the basis for particle discretizations, Fernández et al. [41] used topological contour line data for numerical reconstruction of landscapes, to highlight some more advanced concepts.

3 The basics of classical Continuum Mechanics

3.1 Describing quantities in a continuum

The kinematical framework for describing material motion in a continuum relies on the concept of configurations. A configuration describes the position of each material point in space, for a specific time. The reference configuration describes the body $\mathcal{B}_0 \in \mathbb{R}^3$ at time $t = t_0$. The material points positions is given by vectors $\boldsymbol{X}$. Further, if not stated otherwise, we assume the reference configuration to be stress-free. In the scope of this thesis, the term material point is a technical specification that differs from its meaning in the theoretical concept of describing a continuum. If not stated otherwise, the term material point describes an infinitesimal volume occupied by a body.

For any other given point in time, the material points might be displaced and the body may deform. Such a placement is called current configuration, where the body is denoted by $\mathcal{B} \in \mathbb{R}^3$ and the material point positions are given by $\boldsymbol{x}$. The difference between the configurations is given by the invertible tensor function $\varphi(\boldsymbol{X}, t) : \mathcal{B}_0 \rightarrow \mathcal{B}$ and the deformation gradient $\boldsymbol{F}$, as illustrated in figure 3.1.

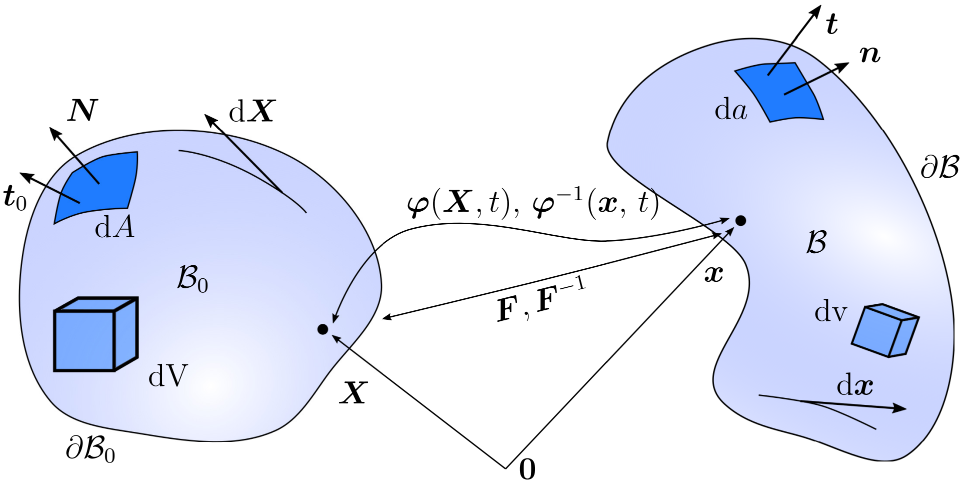


Figure 3.1: Illustration of a body and its evolution in time.

The deformation gradient represents the description of deformation in a continuum. It relates the position vectors of the current configuration to those of the reference configuration as in

$$\boldsymbol{F} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} = \operatorname{Grad} \boldsymbol{u} + \boldsymbol{I}, \tag{3.1}$$

with the displacement vector $\boldsymbol{u} = \boldsymbol{x} - \boldsymbol{X}$ and the second-order identity tensor $\boldsymbol{I}$.

Its determinant, also known as the Jacobian, can be used to describe the volume with

$$J = \det \boldsymbol{F} > 0. \tag{3.2}$$

The deformation gradient and its determinant are used for mapping differential line elements d$\boldsymbol{x}$, surface areas da and volume elements dv between configurations by

$$\mathrm{d}\boldsymbol{x} = \boldsymbol{F} \cdot \mathrm{d}\boldsymbol{X}, \qquad \boldsymbol{n}\,\mathrm{da} = \mathrm{cof}\boldsymbol{F} \cdot \boldsymbol{N}\,\mathrm{dA}, \qquad \mathrm{dv} = J\,\mathrm{dV}, \tag{3.3}$$

where $\boldsymbol{n}$, $\boldsymbol{N}$ denote the normal vectors of differential surface areas in the current and reference placement.

The deformation gradient describes the deformation and the rotation which can be shown in its polar decomposition into a left and right stretch tensor $\boldsymbol{V}$, $\boldsymbol{U}$ and a rotation tensor $\boldsymbol{R}$ as

$$\boldsymbol{F} = \boldsymbol{V} \cdot \boldsymbol{R} = \boldsymbol{R} \cdot \boldsymbol{U}. \tag{3.4}$$

In material modeling it is often desired to have deformation measures which are indifferent to rotation. As such we introduce the left and right Cauchy-Green deformation tensors

$$\begin{aligned} \boldsymbol{b} &= \boldsymbol{F} \cdot \boldsymbol{F}^T = \boldsymbol{V} \cdot \boldsymbol{R} \cdot \boldsymbol{R}^T \cdot \boldsymbol{V}^T = \boldsymbol{V} \cdot \boldsymbol{V}^T, \\ \boldsymbol{C} &= \boldsymbol{F}^T \cdot \boldsymbol{F} = \boldsymbol{U}^T \cdot \boldsymbol{R}^T \cdot \boldsymbol{R} \cdot \boldsymbol{U} = \boldsymbol{U}^T \cdot \boldsymbol{U}, \end{aligned} \tag{3.5}$$

where it can easily be shown that the rotation part is vanishing due to the relation $\boldsymbol{R}^T = \boldsymbol{R}^{-1}$ of orthogonal rotation tensor $\boldsymbol{R}$. Additionally, we can identify the symmetry for both deformation tensors $\boldsymbol{C} = \boldsymbol{C}^T$ and $\boldsymbol{b} = \boldsymbol{b}^T$.

Further, elaborated strain tensors are introduced. Of importance for this work are the Almansi strain tensor $\boldsymbol{a}$ and the Cauchy-Green strain tensor $\boldsymbol{E}$,

$$\boldsymbol{a} = \frac{1}{2}\left(\boldsymbol{I} - \boldsymbol{b}^{-1}\right), \qquad \boldsymbol{E} = \frac{1}{2}\left(\boldsymbol{C} - \boldsymbol{I}\right) \tag{3.6}$$

which are defined with respect to the current and reference configuration, respectively. By definition, any strain measure renders a dimensionless quantity.

3.2 Stresses

It can be observed that a body resists deformation due to external loading. This resistance is modeled in terms of a constitutive behaviors that leads to stresses. Stresses are the internal counterpart to external loading and are balanced to describe equilibrium. Mechanical loads are characterized by force vectors $\boldsymbol{f}$ in N, which can be applied onto a differential surface area. From this, the so-called traction vector is defined as

$$\boldsymbol{t} = \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{da}}. \tag{3.7}$$

The traction vector preserves the direction of the force but describes it as a differential with respect to the corresponding differential surface area. In other words its a directed force per area. In equilibrium, the body which is loaded by traction on its surface builds an internal stress state, which can be described by the Cauchy stress tensor $\boldsymbol{\sigma}$ in $\mathrm{N\,m^{-2}}$ through Cauchy's theorem

$$\boldsymbol{t} = \boldsymbol{\sigma} \cdot \boldsymbol{n}. \tag{3.8}$$

The surface normal $\boldsymbol{n}$ is not necessarily defined only on the boundary surface of the body, but can also be defined on virtual cuts within the solid domain. This concept of virtual cuts to follow the flow of stresses through the body represents a basis in continuum mechanics. Further, in knowing the deformation of a body, the aforementioned relation can be described in the reference configuration of the body as well. Using the transport theorems as depicted in equation (3.3), the definition of the first Piola-Kirchhoff stress tensor $\boldsymbol{\mathcal{P}}$ in $\mathrm{N\,m^{-2}}$ can be formulated by

$$\boldsymbol{t}_0 = \boldsymbol{\mathcal{P}} \cdot \boldsymbol{N}. \tag{3.9}$$

The vector $\boldsymbol{N}$ is defined on the surface in the reference configuration, for which the traction vector $\boldsymbol{t}_0$ is computed. It is the projection of the traction vector $\boldsymbol{t}$ defined on a current surface area into the reference configuration, compare figure 3.1. It is also possible to transform the stress tensors, using the relation

$$\boldsymbol{\sigma} = \frac{1}{J}\boldsymbol{\tau} = \frac{1}{J}\boldsymbol{\mathcal{P}} \cdot \boldsymbol{F}^T, \tag{3.10}$$

with the Kirchhoff stress tensor $\boldsymbol{\tau}$. The constitutive response of a body can be described with respect to both configurations with these stress tensors. All stress tensors and traction vectors alike are defined in $\mathrm{N\,m^{-2}}$. Although the scope of this thesis allows focusing on the here mentioned set of stresses, it might be noted that there exist more definitions of stress tensors for various applications.

3.3 Important rates and other time derivatives

The rates, i.e. change in time of any property are used to describe the evolution of properties in continuum mechanics. For example, the rate of the deformation gradient describes the velocity gradient with respect to the reference configuration

$$\dot{\boldsymbol{F}} = \frac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{X}} = \operatorname{Grad} \boldsymbol{v}. \tag{3.11}$$

The velocity gradient, in return, renders the spatial velocity gradient with respect to the current configuration. The velocity gradient can be computed from the rate of the deformation gradient, sing the chain rule

$$\boldsymbol{L} = \frac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{x}} = \frac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{X}} \cdot \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{x}} = \dot{\boldsymbol{F}} \cdot \boldsymbol{F}^{-1} = \operatorname{grad} \boldsymbol{v}. \tag{3.12}$$

Besides the definition of $\boldsymbol{L}$ itself, we remark the property $\dot{\boldsymbol{F}} = \boldsymbol{L} \cdot \boldsymbol{F}$ which is used later. The velocity gradient is defined in the current configuration, which renders the importance for describing a continuum in the same frame. It is important with respect to the balance equations used in this thesis, and for constitutive equations that work with the true stresses, such as plasticity. As in formerly discussed strain tensors, the velocity gradient can also be additively decomposed into a symmetric $\boldsymbol{d} = \boldsymbol{d}^T$ and skew-symmetric part $\boldsymbol{W} = -\boldsymbol{W}^T$. In a descriptive sense, these tensors represent the rate of deformation

$$\boldsymbol{d} = \frac{1}{2}\left(\boldsymbol{L} + \boldsymbol{L}^T\right) \tag{3.13}$$

and the rate of rotation, i.e., the spin tensor

$$\boldsymbol{w} = \frac{1}{2}\left(\boldsymbol{L} - \boldsymbol{L}^T\right). \tag{3.14}$$

As said, rates describe the change of a property in time. This definition might not necessarily equal the partial differentiation with respect to time, due to hidden dependencies with respect to the properties configuration. For the rate of the Green-Lagrange strain tensor,

$$\dot{\boldsymbol{E}} = \frac{1}{2}\left(\dot{\boldsymbol{F}}^T \cdot \boldsymbol{F} + \boldsymbol{F}^T \cdot \dot{\boldsymbol{F}}\right) = \boldsymbol{F}^T \cdot \frac{1}{2}\left(\boldsymbol{L} + \boldsymbol{L}^T\right) \cdot \boldsymbol{F} = \boldsymbol{F}^T \cdot \boldsymbol{d} \cdot \boldsymbol{F} \tag{3.15}$$

it can be shown that its rate is closely related to the rate of deformation. The difference of both properties is set by their reference frame. The operation used in equation (3.15) is also known as pull back, as it ”pulls” the deformation rate ”back” from the current, to the reference configuration.

As another example, we consider the rate of the left deformation tensor. Using the relation derived above, we get

$$\dot{\boldsymbol{b}} = \left(\boldsymbol{F} \cdot \boldsymbol{F}^T\right)^{\cdot} = \dot{\boldsymbol{F}} \cdot \boldsymbol{F}^T + \boldsymbol{F} \cdot \dot{\boldsymbol{F}^T} = \boldsymbol{L} \cdot \boldsymbol{F} \cdot \boldsymbol{F}^T + \boldsymbol{F} \cdot \boldsymbol{F}^T \cdot \boldsymbol{L}^T, \tag{3.16}$$

and with further applying the symmetry property $\boldsymbol{b} = \boldsymbol{b}^T$ we arrive at

$$\dot{\boldsymbol{b}} = 2\,\boldsymbol{b} \cdot \frac{1}{2}\left(\boldsymbol{L} + \boldsymbol{L}^T\right) = 2\,\boldsymbol{b} \cdot \boldsymbol{d}. \tag{3.17}$$

All rates presented here describe the change of unitless quantities in time, and hence themselves are defined in s^{-1}.

3.4 Differential equations

In the following the most fundamental differential equations to describe a solid continuum are discussed. A so-called local statement can be extracted for each, which describes demands for each infinitesimal volume element in a continua, to satisfy the aforementioned governing equations.

3.4.1 Balance of mass

The balance of mass represents a fundamental balance equation that describes the conservation of mass m. In this context mass is defined as material density ρ in $\mathrm{kg\,m^{-3}}$ integrated over the current volume of the body for which it is defined. We use the volume mapping (3.3) to compute the mass rate by integrating over the reference domain, to exploit all temporal dependencies,

$$\dot{m} = \left[\int_{\mathcal{B}} \rho \,\mathrm{dv}\right]^{\cdot} = \left[\int_{\mathcal{B}_0} J\,\rho \,\mathrm{dV}\right]^{\cdot} = \int_{\mathcal{B}_0} \dot{J}\,\rho + J\,\dot{\rho}\,\mathrm{dV} = 0. \tag{3.18}$$

Using $\dot{J} = J\ \mathrm{div}\,\dot{\boldsymbol{x}}$ we arrive at the local statement of the balance of momentum

$$\int_{\mathcal{B}_0} J\ \mathrm{div}\,\dot{\boldsymbol{x}}\,\rho + J\,\dot{\rho}\,\mathrm{dV} = \int_{\mathcal{B}} \mathrm{div}\,\dot{\boldsymbol{x}}\,\rho + \dot{\rho}\,\mathrm{dv} = 0 \quad \Rightarrow \quad \dot{\rho} + \rho\ \mathrm{div}\,\dot{\boldsymbol{x}} = 0. \tag{3.19}$$

Likewise, the mass of an infinitesimal volume element of a body could be computed using the reference density ρ_0 in $\mathrm{kg\,m^{-3}}$ to expose a consequence of mass conservation for the Jacobian,

$$\mathrm{d}m = \rho\,\mathrm{dv} = \rho_0\,\mathrm{dV} \quad \Rightarrow \quad J = \frac{\rho_0}{\rho}. \tag{3.20}$$

In the framework of a Lagrangian description, where the body is represented by discrete points, rather than a continuous field, equation 3.20 is of special importance.

3.4.2 Balance of linear momentum

Preservation of linear momentum $\boldsymbol{p}$ in $\mathrm{kg\,m\,s^{-1}}$ is the governing equation of motion in a material continuum. Its rate is equal to acting forces in the continuum which can be decomposed. The quantity $\mathfrak{b}$ in $\mathrm{m\,s^{-2}}$ represents a volumetric, load-like acceleration, such as gravity.

$$\dot{\boldsymbol{p}} = \left[\int_{\mathcal{B}} \rho\, \dot{\boldsymbol{x}} \,\mathrm{dv} \right]^{\cdot} = \int_{\mathcal{B}} \rho\, \ddot{\boldsymbol{x}} \,\mathrm{dv} = \int_{\mathcal{B}} \rho\, \mathfrak{b} \,\mathrm{dv} + \int_{\partial\mathcal{B}} \boldsymbol{t} \,\mathrm{da} = \boldsymbol{f}. \tag{3.21}$$

Using Cauchy's theorem (3.8), the traction surface integral can be reformulated as

$$\int_{\partial\mathcal{B}} \boldsymbol{t} \,\mathrm{da} = \int_{\partial\mathcal{B}} \boldsymbol{\sigma} \cdot \boldsymbol{n} \,\mathrm{da} = \int_{\mathcal{B}} \operatorname{div} \boldsymbol{\sigma} \,\mathrm{dv}, \tag{3.22}$$

which leads to the local statement of the balance of linear momentum, defined in the current configuration

$$\operatorname{div} \boldsymbol{\sigma} + \rho\, (\mathfrak{b} - \ddot{\boldsymbol{x}}) = \boldsymbol{0}. \tag{3.23}$$

It is noted here, that the given local statement in equation (3.23) can be equally defined in an integral form for the current, and reference configuration. In the latter case, application of the transport theorems is required; see equation (3.3).

3.4.3 Balance of angular momentum

Just as linear momentum, physics demand equilibrium also for angular momentum $\tilde{\boldsymbol{p}}$ and its force-driven counterpart $\tilde{\boldsymbol{f}}$. Hence we observe its rate

$$\dot{\tilde{\boldsymbol{p}}} = \left[\int_{\mathcal{B}} \boldsymbol{x} \times \rho\, \dot{\boldsymbol{x}} \,\mathrm{dv} \right]^{\cdot} = \int_{\mathcal{B}} \boldsymbol{x} \times \rho\, \ddot{\boldsymbol{x}} \,\mathrm{dv} = \int_{\mathcal{B}} \boldsymbol{x} \times \rho\, \mathfrak{b} \,\mathrm{dv} + \int_{\partial\mathcal{B}} \boldsymbol{x} \times \boldsymbol{t} \,\mathrm{da} = \tilde{\boldsymbol{f}}. \tag{3.24}$$

After some algebraic manipulations, including

$$\int_{\partial\mathcal{B}} \boldsymbol{x} \times \boldsymbol{t} \,\mathrm{da} = \int_{\mathcal{B}} \boldsymbol{x} \times \operatorname{div} \boldsymbol{\sigma} + \boldsymbol{I} \times \boldsymbol{\sigma} \,\mathrm{dv} \tag{3.25}$$

and also usage of the equation (3.23), we arrive at the local statement $\boldsymbol{I} \times \boldsymbol{\sigma} = \boldsymbol{0}$. This local statement demands symmetry of the Cauchy stress tensor,

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^T. \tag{3.26}$$

3.4.4 Balance of energy

The conservation of energy represents another concept postulated for physical processes. While many different contributions of energy exist, we introduce here only the most common definitions for solid mechanics. Energy is defined in $\mathrm{J} = \mathrm{kg\,m^2\,s^{-2}}$. We introduce the capacitive quantities of kinetic energy $\mathcal{K}$ and the internal energy of a material $\mathcal{E}$

$$\mathcal{K} = \frac{1}{2} \int_{\mathcal{B}} \rho\, \dot{\boldsymbol{x}} \cdot \dot{\boldsymbol{x}} \,\mathrm{dv}, \quad \mathcal{E} = \int_{\mathcal{B}} \rho\, \varepsilon \,\mathrm{dv}, \tag{3.27}$$

which relies on the internal specific energy ε in $\mathrm{J\,kg^{-1}}$. The balance of energy, also known as the first law of thermodynamics, states that the rate of stored energy is balanced with the rate of mechanical work $\dot{\mathcal{W}}$ or the rate of thermal work $\dot{\mathcal{Q}}$, consequently both given in $\mathrm{J\,s^{-1}}$ and computed via

$$\dot{\mathcal{W}} = \int_{\mathcal{B}} \dot{\boldsymbol{x}} \cdot \rho\, \mathfrak{b} \,+ \boldsymbol{\sigma} : \boldsymbol{d}\, \mathrm{dv}, \quad \text{and} \quad \dot{\mathcal{Q}} = \int_{\mathcal{B}} \rho\, r - \operatorname{div} \boldsymbol{q}\, \mathrm{dv}. \tag{3.28}$$

In the above equation (3.28), we have introduced a specific heat source r in $\mathrm{W\,kg^{-1}} = \mathrm{m^2\,s^{-3}}$ and the heat flux vector $\boldsymbol{q}$, describing heat flow through an infinitesimal area and thus defined in $\mathrm{W\,m^{-2}} = \mathrm{kg\,s^{-3}}$. With these definitions we arrive at the balance of energy, after some algebraic manipulations and using equation (3.23)

$$[\mathcal{K} + \mathcal{E}]^{\cdot} - \dot{\mathcal{W}} - \dot{\mathcal{Q}} = \int_{\mathcal{B}} \rho\, \dot{\varepsilon} - \boldsymbol{\sigma} : \boldsymbol{d} - \rho\, r + \operatorname{div} \boldsymbol{q}\, \mathrm{dv} = 0. \tag{3.29}$$

Further, we apply the Legendre transformation $\psi = \varepsilon - \theta\, \eta$, with a specific Helmholtz free energy ψ in $\mathrm{J\,kg^{-1}}$, the absolute temperature θ in K and the specific entropy η in $\mathrm{J\,K^{-1}\,kg^{-1}}$. Finally, we present the local statement of the balance of energy as

$$\rho\, \left(\dot{\psi} + \dot{\theta}\, \eta + \theta\, \dot{\eta}\right) - \boldsymbol{\sigma} : \boldsymbol{d} - \rho\, r + \operatorname{div} \boldsymbol{q} = 0. \tag{3.30}$$

3.4.5 Entropy inequality

With entropy inequality, a restriction to the transformation of energy is made. It states that the total entropy in a continuum must not reduce. Any material model that fulfills this inequality constraint is thermodynamically consistent. A thermodynamically inconsistent model, may produce energy in a vacuum. The inequality reads

$$\int_{\mathcal{B}} \rho\, \dot{\eta}\, \mathrm{dv} \geq \int_{\mathcal{B}} \rho\, \frac{r}{\theta} - \frac{1}{\theta} \operatorname{div} \boldsymbol{q}\, \mathrm{dv}. \tag{3.31}$$

It is further possible to use the already derived balance equations (3.23) and (3.30) to include the specific free Helmholtz energy into the local statement of (3.31) to arrive at

$$-\,\rho\, \left(\dot{\psi} + \dot{\theta}\, \eta\right) + \boldsymbol{\sigma} : \boldsymbol{d} - \frac{1}{\theta}\, \boldsymbol{q} \cdot \operatorname{grad} \theta \geq 0. \tag{3.32}$$

4 Constitutive theory

Constitutive models are used to describe a mathematical relationship between deformation measures and stresses. These material models may be arbitrarily, complex and may also depend on deformation independent parameters such as spatial coordinates e.g. fibers with prescribed orientations inside a material. In this thesis we focus on isotropic elastic and elasto-plastic materials. For these materials, it is possible to postulate a specific free Helmholtz energy function ψ as introduced in equation (3.30), which is later shown to represent a basis for stress formulations. In the following, we take a survey on frame indifference as an important restriction to quantities involved in material modeling. With these in hand, the material models used in this thesis are discussed.

4.1 Frame indifference for constitutive quantities

A constitutive equation should satisfy a material frame indifference condition, such that a rigid body rotation does not introduce stresses to a body. A rigid body rotation can be described by an orthogonal rotation tensor $\boldsymbol{Q} \in SO(3)$. Hence the transformation that $\boldsymbol{Q}$ represents, only rotates and does not deform the body. Accordingly, we can introduce rotated coordinate vectors and consistently the deformation gradient with superimposed rotations

$$\boldsymbol{x}^{+} = \boldsymbol{Q} \cdot \boldsymbol{x}, \quad \boldsymbol{F}^{+} = \frac{\partial \boldsymbol{x}^{+}}{\partial \boldsymbol{x}} \cdot \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} = \boldsymbol{Q} \cdot \boldsymbol{F}. \tag{4.1}$$

For the right Cauchy-Green deformation tensor, introduced in equation (3.5), we use the properties of orthogonal tensors $\det \boldsymbol{Q} = 1$ and $\boldsymbol{Q}^T = \boldsymbol{Q}^{-1}$ to show that these deformation measure is indifferent to such a transformation as

$$\boldsymbol{C}^{+} = \boldsymbol{F}^{+^T} \cdot \boldsymbol{F}^{+} = \boldsymbol{F}^T \cdot \boldsymbol{Q}^T \cdot \boldsymbol{Q} \cdot \boldsymbol{F} = \boldsymbol{C}. \tag{4.2}$$

However, a spatial tensor such as the left Cauchy-Green deformation tensor shows a different behavior

$$\boldsymbol{b}^{+} = \boldsymbol{F}^{+} \cdot \boldsymbol{F}^{+^T} = \boldsymbol{Q} \cdot \boldsymbol{F} \cdot \boldsymbol{F}^T \cdot \boldsymbol{Q}^T = \boldsymbol{Q} \cdot \boldsymbol{b} \cdot \boldsymbol{Q}^T. \tag{4.3}$$

The transformation behavior of quantities according to, e.g., Holzapfel [56], is summarized by

$$\begin{array}{lll} \boldsymbol{b}^{+} & = \boldsymbol{Q} \cdot \boldsymbol{b} \cdot \boldsymbol{Q}^T & \text{spatial second order tensors,} \\ \boldsymbol{F}^{+} & = \boldsymbol{Q} \cdot \boldsymbol{F} & \text{two field second order tensors,} \\ \boldsymbol{C}^{+} & = \boldsymbol{C} & \text{material second order tensors,} \\ \boldsymbol{v}^{+} & = \boldsymbol{Q} \cdot \boldsymbol{v} & \text{vectors,} \\ a^{+} & = a & \text{scalar quantities.} \end{array} \tag{4.4}$$

In this thesis, stress functions depend on the specific free Helmholtz energy, according to equation. They obey frame indifference and objectivity requirements by definition, if their arguments are invariants of the deformation tensor $\boldsymbol{b}$,

$$I_1(\boldsymbol{b}) := \operatorname{tr} \boldsymbol{b}, \quad I_2(\boldsymbol{b}) := \operatorname{tr}(\operatorname{cof} \boldsymbol{b}), \quad I_3(\boldsymbol{b}) := \det \boldsymbol{b} \tag{4.5}$$

and scalar quantities such as temperature and a set of internal variables ξ_i,

$$\psi(I_1(\boldsymbol{b}),\, I_2(\boldsymbol{b}),\, I_3(\boldsymbol{b}),\, \theta,\, \xi_i). \tag{4.6}$$

The frame indifference for isotropic materials requires

$$\psi(I_k(\boldsymbol{b}),\, \theta,\, \xi_i \,|\, k = 1,2,3) = \psi(I_k(\boldsymbol{b}^+),\, \theta^+,\, \xi_i^+ \,|\, k = 1,2,3). \tag{4.7}$$

4.2 Hyperelasticity - a NeoHookean material

For the materials in this thesis we focus on isotropic elastic behavior as a general assumption in constitutive modeling. In the following, we briefly review the thermodynamically consistent isothermal, elastic material model and further discuss the actual constitutive relationship between deformation and stresses.

It was mentioned before in chapter 3.4.5 that a thermodynamically consistent material must fulfill the entropy inequality. Further it generally must not conflict with the balance equations for a solid. For this material, we assume temperature to be constant in time and space, which leads to

$$\theta = \text{const.}\ \forall \boldsymbol{x},\, t \Rightarrow \dot{\theta} = 0, \qquad \boldsymbol{q} = \boldsymbol{0}. \tag{4.8}$$

Using these for evaluation of the local statement of the entropy inequality, see equation (3.32)

$$-\,\rho\,\dot{\psi} + \boldsymbol{\sigma} : \boldsymbol{d} \geq 0. \tag{4.9}$$

For the specific free Helmholtz energy, we assume

$$\psi = \hat{\psi}\,(\operatorname{tr}\boldsymbol{b}, \det\boldsymbol{b})\,,\ \dot{\psi} = \frac{\partial\,\hat{\psi}}{\partial\,\boldsymbol{b}} : \dot{\boldsymbol{b}} = 2\,\boldsymbol{b}\cdot\frac{\partial\,\hat{\psi}}{\partial\,\boldsymbol{b}} : \boldsymbol{d}. \tag{4.10}$$

By using the derived energy rate, a thermodynamically consistent stress expression can be found,

$$-\,2\,\rho\,\boldsymbol{b}\cdot\frac{\partial\,\hat{\psi}}{\partial\,\boldsymbol{b}} : \boldsymbol{d} + \boldsymbol{\sigma} : \boldsymbol{d} \geq 0 \quad \Rightarrow \boldsymbol{\sigma} = 2\,\rho\,\boldsymbol{b}\cdot\frac{\partial\,\hat{\psi}}{\partial\,\boldsymbol{b}}, \tag{4.11}$$

compare, e.g., Miehe [100], or Holzapfel [56], based on a suitable specific free Helmholtz energy function.

A common choice for hyperelastic materials is a Neo-Hookean type energy function. While this scheme offers some variations for the pressure part of the stresses, the specific model of this thesis renders

$$\hat{\psi} = \psi^{\mathtt{NH}}\,(\operatorname{tr}\boldsymbol{b}, \det\boldsymbol{b}) = \frac{1}{\rho}\left[\frac{\Lambda}{2}\,(\ln J)^2 + \frac{G}{2}\,(\operatorname{tr}\boldsymbol{b} - 3) - G\,\ln J\right], \tag{4.12}$$

with the Jacobian expressed via $J = \sqrt{\det\boldsymbol{b}}$. Further, we use the Lamé constants Λ and G in $\mathrm{N\,m^{-2}}$. Carrying out the derivatives yields a convenient expression for the Cauchy stresses

$$\boldsymbol{\sigma} = \frac{\Lambda}{J}\ln(J)\,\boldsymbol{I} + \frac{G}{J}\,(\boldsymbol{b} - \boldsymbol{I})\,. \tag{4.13}$$

4.3 Thermo-Plasticity

The main application of the MPM in this thesis is modeling metallic materials. For metallic materials, the assumption of elasticity is only suitable for very small deformations. With respect to large deformations specifically, an elasto-plastic material response can be observed. After a certain yield criterion, metallic materials deform inelastically, which means that the deformation can no longer be recovered. Further, the energy for such deformations cannot be retained as in purely elastic deformation.

The reason for this behavior in metallic materials is given by their nanoscale structure. The atoms and molecules in common metals are usually arranged in crystalline lattices. Since most crystalline structures have orientation features, an anisotropic deformation pattern is introduced. Further, on a mesoscale level, the material consists of subregions called grains. Each grain features a lattice orientation, averaging out the anisotropy of the individual grains on a macro level. Notice that some metallic materials still feature anisotropic features for this reason. Nevertheless in this thesis we restrict to isotropic material response on the macroscopic level. An in-depth review of the microstructure of metals is given, e.g., in Hornbogen and Warlimont [59].

The crystalline lattice, however, gives reason to the elasto-plastic behavior. With small deformations, the lattice formation is stretched. This process so far is reversible and the energy introducing this stretch is stored as strain energy into the material. With a materialspecific amount of lattice deformation, it is energetically preferable for the overall response to slip within the crystalline lattice rather than stretch further. This reaction does not necessarily lead to a different structure but represents the irreversible component in the constitutive response. The energy which is used for such a slip action is not stored as strain energy. It is rather mainly transferred into heat, as the slip lines exceed friction between each other. Only a small fraction of the energy is kept in the lattice system. This micro-structural phenomenon can be directly resolved, e.g., in crystal plasticity approaches, compare, e.g., Lubliner [91]. However, one can easily imagine that with a certain amount of slip inside a larger lattice or grain, the resistance to further slipping is raised. Especially on the mesoscale with a composition of grains. Thus, the resistance to slip raises with the introduction of more slip actions. This phenomenon is macroscopically described as hardening.

In the following, we discuss a general continuum mechanical framework for plasticity. Afterward, we evaluate the entropy inequality to obtain a thermodynamically consistent stress relation. Using the principle of maximum dissipation lead to a formulation for the evolution of plastic deformation, which is also discussed as an algorithmic treatment. In the scope of this thesis, we present the rate-independent, isothermal J^2 formulation, which represents a limit case of the derived framework. At last, the rate-dependent JohnsonCook model for finite plasticity is derived. It features a rate-dependent yield surface and adiabatic heating driven by plastic work. The algorithmic framework for these model has its foundation in the works of Simo [131], Miehe [100] and Simo and Hughes [133], to name just a few.

4.3.1 Framework for finite deformation plasticity

Continuum mechanical models for finite deformation plasticity have been of interest for a long time. The reason is simply the significant number of physical applications, as many materials deform plastically. This is not only the case for metals but also granular materials in geo-engineering. It is clear that to handle arbitrary deformations, e.g., finite strains and large rotations, a nonlinear framework is required. A well-established concept is the introduction of a multiplicative split of the deformation gradient

$$\boldsymbol{F} = \boldsymbol{F}^e \cdot \boldsymbol{F}^p. \tag{4.14}$$

It proposes an elastic $\boldsymbol{F}^e$ and a plastic part $\boldsymbol{F}^p$, implying a detour from the reference to the current configuration mapping, through an intermediate plastic configuration. Thus a multiplicative decomposition of the Jacobian $J = J^e\, J^p$ is introduced. Based on the assumption of singlecrystal material the plastic configuration indicates the volumepreserving ($J^p = 1$), unrotated crystalline lattice, which is deformed though slip phenomena. Notice that this configuration is assumed stress-free. The mapping towards the current configuration superimposes the material rotations and also the stretch on this crystalline lattice. In total, both mappings result in an elasto-plastic material deformation.

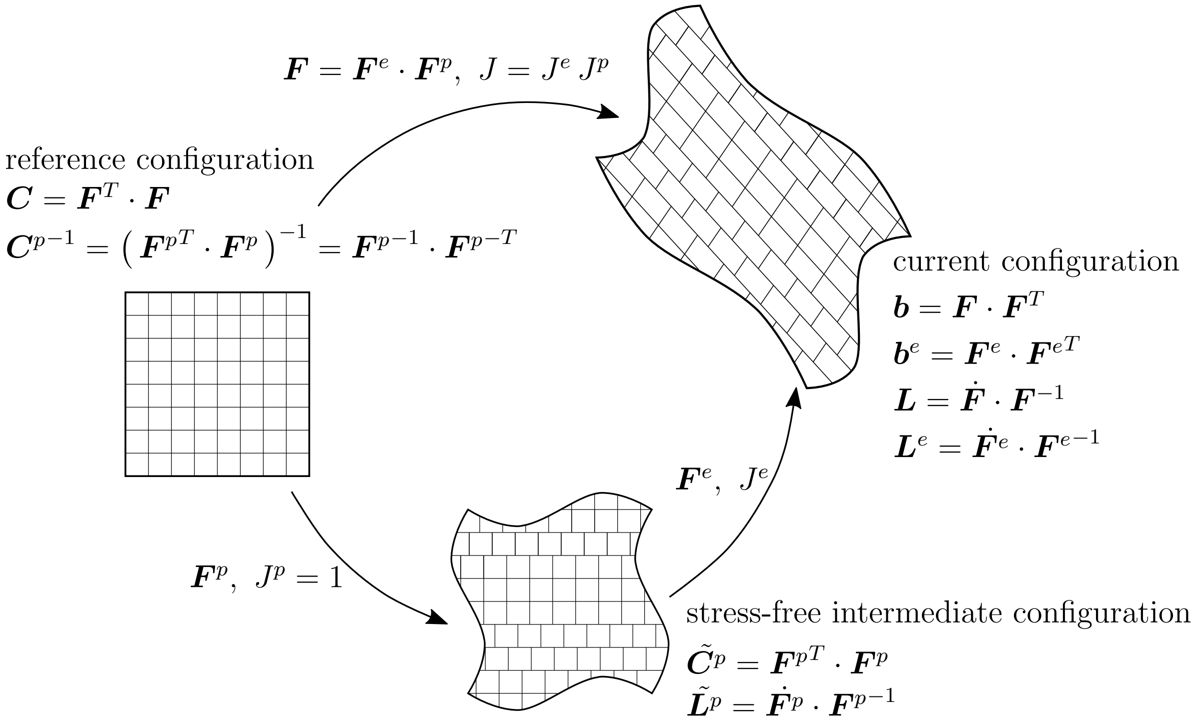

$\boldsymbol{F} = \boldsymbol{F}^e \cdot \boldsymbol{F}^p,\ J = J^e\, J^p$
reference configuration
$\boldsymbol{C} = \boldsymbol{F}^T \cdot \boldsymbol{F}$
$\boldsymbol{C}^{p-1} = \left(\boldsymbol{F}^{pT} \cdot \boldsymbol{F}^p\right)^{-1} = \boldsymbol{F}^{p-1} \cdot \boldsymbol{F}^{p-T}$
current configuration
$\boldsymbol{b} = \boldsymbol{F} \cdot \boldsymbol{F}^T$
$\boldsymbol{b}^e = \boldsymbol{F}^e \cdot \boldsymbol{F}^{eT}$
$\boldsymbol{L} = \dot{\boldsymbol{F}} \cdot \boldsymbol{F}^{-1}$
$\boldsymbol{L}^e = \dot{\boldsymbol{F}}^e \cdot \boldsymbol{F}^{e-1}$
$\boldsymbol{F}^e,\ J^e$
$\boldsymbol{F}^p,\ J^p = 1$
stress-free intermediate configuration
$\tilde{\boldsymbol{C}}^p = \boldsymbol{F}^{pT} \cdot \boldsymbol{F}^p$
$\tilde{\boldsymbol{L}}^p = \dot{\boldsymbol{F}}^p \cdot \boldsymbol{F}^{p-1}$

Figure 4.1: Local mapping for a material point inside a continuum with intermediate plastic configuration following multiplicative split. Additionally mapping properties are given in their associated configuration, respectively.

From this framework, a set of derived quantities arise, which are summarized in figure 4.1. For the derivations in this thesis, we focus on the elastic, left Cauchy deformation tensor $\boldsymbol{b}^e = \boldsymbol{F}^e \cdot \boldsymbol{F}^{eT}$. It describes deformation with respect to the current configuration, just as $\boldsymbol{b}$, but only the elastic part. For differentiation purposes, we rely on the term elastic finger tensor, respectively. In order to describe the evolution of the plastic deformation in this continuum, the rate of elastic finger tensor is required. The finger tensor can be formulated by means of the plastic evolution to expose its full dependencies. Using $\boldsymbol{F}^e = \boldsymbol{F} \cdot \boldsymbol{F}^{p-1}$, we get

$$\boldsymbol{b}^e = \boldsymbol{F} \cdot \boldsymbol{F}^{p-1} \cdot \left(\boldsymbol{F} \cdot \boldsymbol{F}^{p-1}\right)^T = \boldsymbol{F} \cdot \boldsymbol{F}^{p-1} \boldsymbol{F}^{p-T} \cdot \boldsymbol{F}^T = \boldsymbol{F} \cdot \boldsymbol{C}^{p-1} \cdot \boldsymbol{F}^T. \tag{4.15}$$

The consistent material time derivative then reads

$$\dot{\boldsymbol{b}}^e = \dot{\boldsymbol{F}} \cdot \boldsymbol{C}^{p-1} \cdot \boldsymbol{F}^T + \boldsymbol{F} \cdot \dot{\boldsymbol{C}}^{p-1} \cdot \boldsymbol{F}^T + \boldsymbol{F} \cdot \boldsymbol{C}^{p-1} \cdot \dot{\boldsymbol{F}}^T. \tag{4.16}$$

Using the identity $\boldsymbol{F}^{-1} \cdot \boldsymbol{F} = \boldsymbol{I}$, as well as equation (3.12), we can apply the transformation

$$\dot{\boldsymbol{F}} \cdot \boldsymbol{C}^{p-1} \cdot \boldsymbol{F}^T = \dot{\boldsymbol{F}} \cdot \boldsymbol{F}^{-1} \cdot \boldsymbol{F} \cdot \boldsymbol{C}^{p-1} \cdot \boldsymbol{F}^T = \boldsymbol{L} \cdot \boldsymbol{F} \cdot \boldsymbol{C}^{p-1} \cdot \boldsymbol{F}^T = \boldsymbol{L} \cdot \boldsymbol{b}^e, \tag{4.17}$$

to arrive at a very common expression for the rate of the elastic finger tensor

$$\dot{\boldsymbol{b}}^e = \boldsymbol{L} \cdot \boldsymbol{b}^e + \boldsymbol{b}^e \cdot \boldsymbol{L}^T + \boldsymbol{\mathcal{L}}_{\boldsymbol{v}} \boldsymbol{b}^e, \tag{4.18}$$

where $\boldsymbol{\mathcal{L}}_{\boldsymbol{v}} \boldsymbol{b}^e = \boldsymbol{F} \cdot \dfrac{\partial}{\partial t} \boldsymbol{C}^{p-1} \cdot \boldsymbol{F}^T$ is known as the Lie derivative of $\boldsymbol{b}^e$. Lie derivatives are used to obtain objective rates. The interested reader is referred, e.g., to Holzapfel [57] and for a more detailed look on $\boldsymbol{\mathcal{L}}_{\boldsymbol{v}} \boldsymbol{b}^e$ to the appendix D.

The local continuum, as depicted in figure 4.1, is based on a multiplicative decomposition of the deformation gradient. However, it can be shown, as, e.g., in Wriggers [156] that the spatial velocity gradients show additive character

$$\boldsymbol{L}^p = \boldsymbol{L} - \boldsymbol{L}^e \quad \text{with} \quad \boldsymbol{L}^p = \boldsymbol{F}^e \cdot \tilde{\boldsymbol{L}}^p \cdot \boldsymbol{F}^{e-1}. \tag{4.19}$$

Applying the same symmetric, skew-symmetric decomposition as for the classical continuum, compare equations (3.1) and (3.14), the additive split of the spatial deformation rate and spin is exposed

$$\boldsymbol{d} = \boldsymbol{d}^e + \boldsymbol{d}^p \quad \text{and} \quad \boldsymbol{w} = \boldsymbol{w}^e + \boldsymbol{w}^p. \tag{4.20}$$

It is worth noting that we adopt the common assumption, that plastic deformation is free of any rotation for this thesis. Hence also, the plastic spin vanishes $\boldsymbol{w}^p = \boldsymbol{0}$, which reflects on the plastic deformation rate $\boldsymbol{L}^p = \boldsymbol{d}^p$.

4.3.2 Thermo-plastic stress model

In the following, we derive a thermodynamically consistent material model, for thermo-elasto-plasticity in a finite strain framework. Additionally to standard plasticity, which is covered by this model, we derive the terms to account for heat generation from plastic work under adiabatic assumptions. The continuum framework for the in-elastic state is based on the multiplicative decomposition of the deformation gradient $\boldsymbol{F} = \boldsymbol{F}^e \cdot \boldsymbol{F}^p$. Further, we assume the utilization of a specific free energy function $\psi(\operatorname{tr} \boldsymbol{b}^e, \det \boldsymbol{b}^e, \theta, \xi_i)$ in $\mathrm{J\,kg^{-1}}$, which depends on invariants of the elastic finger tensor $\boldsymbol{b}^e$, the absolute temperature θ in K, and a set of dimensionless scalar internal variables ξ_i. Notice that with this choice, the requirements for frame indifference are fulfilled, according to chapter 4.1, respectively. For the here derived J_2 plasticity, we have $\xi_i = [\alpha]$, where α renders the dimensionless accumulated plastic arc length. In the J_2 flow theory, the single surface yield function defined by the invariant J_2, see equation (B.1), is uniformly scaled by this parameter. In the material science context, it is also referred to as equivalent plastic strain ε^{pl} and provides a measure for the amount of plastic deformation. J_2 flow theory is known to be suitable for modeling metallic materials. As we see later, incompressibility on the plastic flow is set. This property is motivated by experimental observations on metallic materials. Further, we use a von Mises yield criterion. This criterion for yielding is based on von Mises stresses and is known to be suitable for ductile materials.

A coupled specific free energy function based on NeoHookean elastic part is given with

$$\psi(\operatorname{tr}\boldsymbol{b}^e, \det\boldsymbol{b}^e, \theta, \alpha) = \psi^{\mathtt{NH}}(\operatorname{tr}\boldsymbol{b}^e, \det\boldsymbol{b}^e) + Q(\theta) + M(\det\boldsymbol{b}^e, \theta) + W(\alpha). \tag{4.21}$$

Notice that although $\psi^{\mathtt{NH}}$ here takes only the elastic part as arguments, it has the same structure as introduced in equation (4.12). Further, we have introduced a coupling term. It represents strain energy, induced by a temperature change relative to the stress-free initial temperature θ_0,

$$M(\det\boldsymbol{b}^e, \theta) = -\frac{3}{\rho}\,\alpha_\sigma\,\kappa\,(\theta - \theta_0)\,\ln J^e \tag{4.22}$$

with the compression modulus κ in $\mathrm{N\,m^{-2}}$, the heat expansion coefficient α_σ in $\mathrm{K^{-1}}$ and also using the alternative representation for the elastic Jacobian $J^e = \sqrt{\det\boldsymbol{b}^e}$.

The stored free energy density in terms of temperature change is included via the term $Q(\theta)$ to account for the non-isothermal nature of the model, with

$$Q(\theta) = -c_\theta\left(\theta\,\ln\frac{\theta}{\theta_0} - \theta + \theta_0\right). \tag{4.23}$$

Hereby we introduce the specific heat capacity c_θ in $\mathrm{J\,kg^{-1}\,K^{-1}}$. At this point, the term $W(\alpha)$ is not given explicitly to preserve flexibility in that regard. It is clear, that it represents free energy, stored from plastic deformation. $W(\alpha)$ is responsible for the hardening behavior of the model. It is later shown that only the first derivative $\partial_\alpha W$ is needed and defined in terms of a yield criterion.

In order to derive the model, partial derivatives of the specific free energy are presented

$$\begin{aligned}
&\frac{\partial\psi}{\partial\boldsymbol{b}^e} = \frac{1}{\rho}\left[\frac{\Lambda}{2}\,\ln J^e\,\boldsymbol{b}^{e-1} + \frac{G}{2}\,\left(\boldsymbol{I} - \boldsymbol{b}^{e-1}\right) - \frac{3}{2}\,\alpha_\sigma\,\kappa\,(\theta-\theta_0)\,\boldsymbol{b}^{e-1}\right],\\
&\frac{\partial\psi}{\partial\theta} = \frac{1}{\rho}\left[-c_\theta\,\rho\,\ln\frac{\theta}{\theta_0} - 3\,\alpha_\sigma\,\kappa\,\ln J^e\right] \quad \text{and} \quad \frac{\partial^2\psi}{\partial\theta^2} = \frac{1}{\rho}\left[-\frac{1}{\theta}\,c_\theta\rho\right].
\end{aligned} \tag{4.24}$$

Using the partial derivatives of equation (4.24), the material time derivative of the specific free Helmholtz energy of equation (4.21) can be derived with

$$\begin{aligned}
\dot{\psi} &= \frac{\partial\psi}{\partial\boldsymbol{b}^e} : \dot{\boldsymbol{b}}^e + \frac{\partial\psi}{\partial\theta}\,\dot{\theta} + \frac{\partial\psi}{\partial\alpha}\,\dot{\alpha}\\
&= \frac{\partial\psi}{\partial\boldsymbol{b}^e} : \left[\boldsymbol{L}\cdot\boldsymbol{b}^e + \boldsymbol{b}^e\cdot\boldsymbol{L}^T + \boldsymbol{\pounds}_{\boldsymbol{v}}\boldsymbol{b}^e\right] + \frac{\partial\psi}{\partial\theta}\,\dot{\theta} + \frac{\partial\psi}{\partial\alpha}\,\dot{\alpha}\\
&= 2\,\frac{\partial\psi}{\partial\boldsymbol{b}^e}\cdot\boldsymbol{b}^e : \left[\boldsymbol{L} + \frac{1}{2}\boldsymbol{\pounds}_{\boldsymbol{v}}\boldsymbol{b}^e\cdot\boldsymbol{b}^{e-1}\right] + \frac{\partial\psi}{\partial\theta}\,\dot{\theta} + \frac{\partial\psi}{\partial\alpha}\,\dot{\alpha}\\
&= 2\,\frac{\partial\psi}{\partial\boldsymbol{b}^e}\cdot\boldsymbol{b}^e : \boldsymbol{L} + \frac{\partial\psi}{\partial\boldsymbol{b}^e}\cdot\boldsymbol{b}^e : \left[\boldsymbol{\pounds}_{\boldsymbol{v}}\boldsymbol{b}^e\cdot\boldsymbol{b}^{e-1}\right] + \frac{\partial\psi}{\partial\theta}\,\dot{\theta} + \frac{\partial\psi}{\partial\alpha}\,\dot{\alpha}\\
&= 2\,\frac{\partial\psi}{\partial\boldsymbol{b}^e}\cdot\boldsymbol{b}^e : \boldsymbol{d} + \frac{\partial\psi}{\partial\boldsymbol{b}^e}\cdot\boldsymbol{b}^e : \left[\boldsymbol{\pounds}_{\boldsymbol{v}}\boldsymbol{b}^e\cdot\boldsymbol{b}^{e-1}\right] + \frac{\partial\psi}{\partial\theta}\,\dot{\theta} + \frac{\partial\psi}{\partial\alpha}\,\dot{\alpha},
\end{aligned} \tag{4.25}$$

where, for the last simplification the symmetry of $\boldsymbol{b}^e$ was used

$$2\,\frac{\partial\,\psi}{\partial\,\boldsymbol{b}^e}\cdot\boldsymbol{b}^e:\boldsymbol{L}=2\,\frac{\partial\,\psi}{\partial\,\boldsymbol{b}^e}\cdot\boldsymbol{b}^e:(\boldsymbol{d}+\boldsymbol{w})=2\,\frac{\partial\,\psi}{\partial\,\boldsymbol{b}^e}\cdot\boldsymbol{b}^e:\boldsymbol{d}. \tag{4.26}$$

The material time derivative can now be used in the evaluation of the entropy inequality to derive a thermodynamically consistent constitutive relation. The entropy inequality reads

$$\begin{aligned}\mathcal{D}\ &=-\rho\left(\dot{\psi}+\dot{\theta}\,\eta\right)+\boldsymbol{\sigma}:\boldsymbol{d} && \geq 0\\ &=-\rho\left(2\,\frac{\partial\,\psi}{\partial\,\boldsymbol{b}^e}\cdot\boldsymbol{b}^e:\boldsymbol{d}+\frac{\partial\,\psi}{\partial\,\boldsymbol{b}^e}\cdot\boldsymbol{b}^e:\left[\boldsymbol{\pounds}_{\boldsymbol{v}}\boldsymbol{b}^e\cdot\boldsymbol{b}^{e-1}\right]+\frac{\partial\,\psi}{\partial\,\theta}\,\dot{\theta}+\frac{\partial\,\psi}{\partial\,\alpha}\,\dot{\alpha}+\dot{\theta}\,\eta\right)+\boldsymbol{\sigma}:\boldsymbol{d} && \geq 0\\ &=\left(\boldsymbol{\sigma}-\rho\,2\,\frac{\partial\,\psi}{\partial\,\boldsymbol{b}^e}\cdot\boldsymbol{b}^e\right):\boldsymbol{d}-\rho\left(\frac{\partial\,\psi}{\partial\,\theta}+\eta\right)\dot{\theta}-\rho\,\frac{\partial\,\psi}{\partial\,\boldsymbol{b}^e}\cdot\boldsymbol{b}^e:\left[\boldsymbol{\pounds}_{\boldsymbol{v}}\boldsymbol{b}^e\cdot\boldsymbol{b}^{e-1}\right]-\rho\frac{\partial\,\psi}{\partial\,\alpha}\,\dot{\alpha} && \geq 0.\end{aligned} \tag{4.27}$$

Using standard argumentation, expressions for the Cauchy stresses and the entropy can be found with

$$\boldsymbol{\sigma}=2\rho\,\frac{\partial\,\psi}{\partial\,\boldsymbol{b}^e}\cdot\boldsymbol{b}^e,\quad \eta=-\frac{\partial\,\psi}{\partial\,\theta}. \tag{4.28}$$

Inserting these relations into the energy inequality, the remaining dissipation reads

$$\begin{aligned}\mathcal{D}\ &=-\rho\,\frac{\partial\,\psi}{\partial\,\boldsymbol{b}^e}\cdot\boldsymbol{b}^e:\left[\boldsymbol{\pounds}_{\boldsymbol{v}}\boldsymbol{b}^e\cdot\boldsymbol{b}^{e-1}\right]-\rho\frac{\partial\,\psi}{\partial\,\alpha}\,\dot{\alpha}\geq 0\\ &=-\boldsymbol{\sigma}:\frac{1}{2}\,\left[\boldsymbol{\pounds}_{\boldsymbol{v}}\boldsymbol{b}^e\cdot\boldsymbol{b}^{e-1}\right]+\beta\,\dot{\alpha}\geq 0,\end{aligned} \tag{4.29}$$

with the stress like quantity $\beta=-\rho\,\dfrac{\partial\,\psi}{\partial\,\alpha}=-\dfrac{\partial\,W}{\partial\,\alpha}$.

A more common representation can be achieved when using the expression of the Lie derivative in terms of the plastic deformation rate in the intermediate configuration as derived in more detail in equation (D.4). With the operation $\boldsymbol{F}^e\cdot(\bullet)\cdot\boldsymbol{F}^{e-1}$ representing the effective push forward of $(\bullet)$ from the intermediate to the current configuration, we obtain

$$\begin{aligned}\boldsymbol{\pounds}_{\boldsymbol{v}}\boldsymbol{b}^e\cdot\boldsymbol{b}^{e-1}\ &=-2\,\boldsymbol{F}^e\cdot\tilde{\boldsymbol{d}}^p\cdot\boldsymbol{F}^{eT}\cdot\boldsymbol{b}^{e-1}\\ &=-2\,\boldsymbol{F}^e\cdot\tilde{\boldsymbol{d}}^p\cdot\boldsymbol{F}^{eT}\cdot\left(\boldsymbol{F}^e\cdot\boldsymbol{F}^{eT}\right)^{-1}\\ &=-2\,\boldsymbol{F}^e\cdot\tilde{\boldsymbol{d}}^p\cdot\boldsymbol{F}^{eT}\cdot\boldsymbol{F}^{e-T}\cdot\boldsymbol{F}^{e-1}\\ &=-2\,\boldsymbol{F}^e\cdot\tilde{\boldsymbol{d}}^p\cdot\boldsymbol{F}^{e-1}\\ &=-2\,\boldsymbol{d}^p.\end{aligned} \tag{4.30}$$

Finally we arrive at the dissipation for the thermoplastic (TP) model.

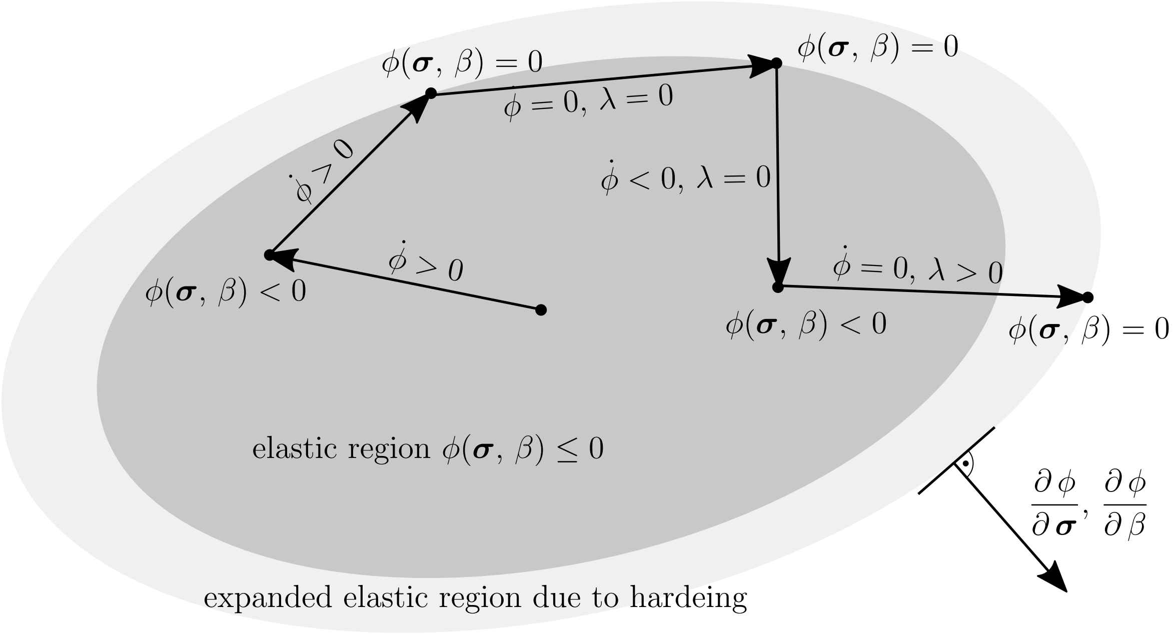
$$\mathcal{D}^{\texttt{TP}}=\boldsymbol{\sigma}:\boldsymbol{d}^p+\beta\,\dot{\alpha}\geq 0. \tag{4.31}$$

The fundamental output at this point is the consistent expressions for stresses, the entropy, and the remainder of dissipation. The latter are used to derive the missing evolution equations for the plastic flow. The stresses and entropy are used to evaluate the balance of energy and derive an expression for the temperature increase due to plastic work, and for solving the balance of linear momentum.

The derivations are equally valid for isothermal elasto-plasticity, i.e. $\psi(\boldsymbol{b}^e,\,\alpha)$. Of course, with the temperature not considered, the balance of energy does not need to be evaluated.

4.3.3 The principle of maximum dissipation

With the dissipation for the thermoplastic material model of equation (4.31), we have the major constraint on the evolution of plastic variables. The material response can be elastic or elastoplastic, where for both cases, the stresses and stresslike quntities are owed to be found in the elastic region. In this thesis, the elastic region is always convex and defined by a yield function $\phi(\boldsymbol{\sigma},\,\beta)$. The yield criterion $\phi(\boldsymbol{\sigma},\,\beta) \leq 0$ must be satisfied for any valid solutions.



$\phi(\boldsymbol{\sigma},\,\beta) = 0$
$\phi(\boldsymbol{\sigma},\,\beta) = 0$
$\phi = 0,\,\lambda = 0$
$\dot{\phi} > 0$
$\dot{\phi} < 0,\,\lambda = 0$
$\dot{\phi} > 0$
$\dot{\phi} = 0,\,\lambda > 0$
$\phi(\boldsymbol{\sigma},\,\beta) < 0$
$\phi(\boldsymbol{\sigma},\,\beta) < 0$
$\phi(\boldsymbol{\sigma},\,\beta) = 0$
elastic region $\phi(\boldsymbol{\sigma},\,\beta) \leq 0$
$\dfrac{\partial\phi}{\partial\boldsymbol{\sigma}},\,\dfrac{\partial\phi}{\partial\beta}$
expanded elastic region due to hardeing

Figure 4.2: Illustration of elastic region defined in the space of stresses and stresslike quantities described by a yield function $\phi(\boldsymbol{\sigma},\,\beta)$. The illustrated loading path covers admissible states for the presented elasto-plastic model. Notice that the Lagrangian fucntional and hence the multiplier λ is only considered at the limit case $\phi = 0$.

Notice that as long as the material response is elastic $\phi(\boldsymbol{\sigma},\,\beta) \leq 0$, i.e., $\boldsymbol{d}^p = \boldsymbol{0}$ and $\dot{\alpha} = 0$, the dissipation is always equal to zero $\mathcal{D}^{\text{TP}} = 0$, and thus, equation (4.31) is satisfied.

In the case that the current stress state violates the yield criterion $\phi(\boldsymbol{\sigma},\,\beta) > 0$, the elasto-plastic response is observed. In order to provide thermodynamically consistent evolution equations for the plastic variables, the principle of maximum dissipation is often applied, see, e.g., Simo [131]. In order to maximize plastic dissipation $\mathcal{D}^{\text{TP}}$ while satisfying the yield criterion for the valid solution, a Lagrangian is formulated,

$$\mathcal{L}^{\text{TP}}\,(\boldsymbol{\sigma},\,\beta,\,\lambda) = -\mathcal{D}^{\text{TP}}(\boldsymbol{\sigma},\,\beta) + \lambda\,\phi(\boldsymbol{\sigma},\,\beta), \tag{4.32}$$

introducing the Lagrangian multiplier λ. Maximizing the Lagrangian by using partial derivatives of the plastic dissipation

$$\frac{\partial\,\mathcal{D}^{\texttt{TP}}}{\partial\,\boldsymbol{\sigma}} = \boldsymbol{d}^p, \quad \frac{\partial\,\mathcal{D}^{\texttt{TP}}}{\partial\,\beta} = \dot{\alpha}, \tag{4.33}$$

yields directly the evolution equations for the plastic variables,

$$\begin{aligned} \frac{\partial\,\mathcal{L}^{\texttt{TP}}}{\partial\,\boldsymbol{\sigma}} &= -\boldsymbol{d}^p + \lambda\frac{\partial\,\phi}{\partial\,\boldsymbol{\sigma}} &&= \mathbf{0} &&\Rightarrow\ \boldsymbol{d}^p = \lambda\frac{\partial\,\phi}{\partial\,\boldsymbol{\sigma}} \\ \frac{\partial\,\mathcal{L}^{\texttt{TP}}}{\partial\,\beta} &= -\dot{\alpha} + \lambda\frac{\partial\,\phi}{\partial\,\beta} &&= 0 &&\Rightarrow\ \dot{\alpha} = \lambda\frac{\partial\,\phi}{\partial\,\beta} \\ \frac{\partial\,\mathcal{L}^{\texttt{TP}}}{\partial\,\lambda} &= \phi(\boldsymbol{\sigma},\,\beta) &&= 0. \end{aligned} \tag{4.34}$$

Hereby the yield criterion acts as a side condition to the optimization problem. The consistency condition $\lambda\,\dot{\phi} = 0$ is further satisfied by the Kuhn-Tucker conditions,

$$\lambda \geq 0, \quad \phi \leq 0, \quad \lambda\,\phi = 0. \tag{4.35}$$

If the current state represent the limit case $\phi = 0$, analysis of the rate of the yield function together with the interpretation of the Lagrangian multiplier as a measure of plastic flow allows for a local interpretation of the material state, i.e.,

$$\begin{aligned} &\dot{\phi} < 0 \quad \text{and} \quad \lambda = 0 \ \rightarrow \text{elastic unloading,} \\ &\dot{\phi} = 0 \quad \text{and} \quad \lambda = 0 \ \rightarrow \text{indifferent loading,} \\ &\dot{\phi} = 0 \quad \text{and} \quad \lambda > 0 \ \rightarrow \text{plastic response.} \end{aligned} \tag{4.36}$$

An illustration of a possible loading path for the described framework is given in figure 4.2. Notice that the evolution equations given in (4.34) demand plastic flow, normal to the surface of the yield function. This property categorizes associative plasticity. However, in the case of plastic flow, the yield function itself changes due to the effects of hardening, or, e.g., thermal softening. Consequently, the tangent directors $\frac{\partial\,\phi}{\partial\,\boldsymbol{\sigma}}, \frac{\partial\,\phi}{\partial\,\beta}$ generally render nonlinear behavior, which has to be taken into account for the numerical treatment.

In this thesis, we use J_2 plasticity based on a von Mises yield criterion. In this scope, the appearance of ϕ renders a specific structure, i.e.

$$\phi(\boldsymbol{\sigma},\,\beta) = \sigma_{\texttt{VM}} - \beta, \tag{4.37}$$

which yields the partial derivatives

$$\frac{\partial\,\phi}{\partial\,\boldsymbol{\sigma}} = \sqrt{\frac{3}{2}}\frac{\partial\,\|\,\mathrm{dev}\,\boldsymbol{\sigma}\|}{\partial\,\boldsymbol{\sigma}} = \sqrt{\frac{3}{2}}\mathbb{N}, \quad \text{and} \quad \frac{\partial\,\phi}{\partial\,\beta} = -1. \tag{4.38}$$

4.3.4 Adiabatic transformation of plastic work into heat

The model discussed in this thesis assumes adiabatic conditions. That implies that heat generated in a material point accumulates in terms of temperature. In order to quantify the temperature increase, the balance of energy (3.30) is evaluated based on the dependencies in (4.21). The adiabatic constraint eliminates heat flux into or out of the considered infinitesimal volume $\operatorname{div}\boldsymbol{q} = 0$, and external heat sources are neglected $r = 0$. With these conditions we have,

$$\begin{aligned}
0 &= \rho\left(\dot{\psi} + \dot{\theta}\,\eta + \theta\,\dot{\eta}\right) - \boldsymbol{\sigma} : \boldsymbol{d} \\
0 &= \rho\left(\frac{\partial\psi}{\partial\boldsymbol{b}^e} : \dot{\boldsymbol{b}}^e + \frac{\partial\psi}{\partial\theta}\,\dot{\theta} + \frac{\partial\psi}{\partial\alpha}\,\dot{\alpha} - \frac{\partial\psi}{\partial\theta}\,\dot{\theta} + \theta\,\dot{\eta}\right) - \boldsymbol{\sigma} : \boldsymbol{d} \\
0 &= \rho\left(\frac{\partial\psi}{\partial\boldsymbol{b}^e} : \dot{\boldsymbol{b}}^e + \frac{\partial\psi}{\partial\alpha}\,\dot{\alpha} + \theta\,\dot{\eta}\right) - \boldsymbol{\sigma} : \boldsymbol{d}.
\end{aligned} \tag{4.39}$$

Using the entropy equation (4.28), and its consistent rate $\dot{\eta} = \dfrac{\partial\eta}{\partial\theta}\,\dot{\theta} = \dfrac{1}{\theta}\,c_\theta\,\dot{\theta}$, together with transformations applied also in equation (4.25), we may write

$$\begin{aligned}
0 &= \rho\left(\frac{\partial\psi}{\partial\boldsymbol{b}^e} : \dot{\boldsymbol{b}}^e + \frac{\partial\psi}{\partial\alpha}\,\dot{\alpha} + \theta\,\dot{\eta}\right) - \boldsymbol{\sigma} : \boldsymbol{d} \\
0 &= \rho\left(\frac{\partial\psi}{\partial\boldsymbol{b}^e} : \dot{\boldsymbol{b}}^e + \frac{\partial\psi}{\partial\alpha}\,\dot{\alpha} + c_\theta\,\dot{\theta}\right) - \boldsymbol{\sigma} : \boldsymbol{d} \\
\rho\,c_\theta\,\dot{\theta} &= -\rho\left(\frac{\partial\psi}{\partial\boldsymbol{b}^e} : \dot{\boldsymbol{b}}^e + \frac{\partial\psi}{\partial\alpha}\,\dot{\alpha}\right) + \boldsymbol{\sigma} : \boldsymbol{d} \\
\rho\,c_\theta\,\dot{\theta} &= -2\rho\,\frac{\partial\psi}{\partial\boldsymbol{b}^e}\cdot\boldsymbol{b}^e : \boldsymbol{d} - 2\rho\,\frac{\partial\psi}{\partial\boldsymbol{b}^e}\cdot\boldsymbol{b}^e : \frac{1}{2}\left[\boldsymbol{\pounds}_{\boldsymbol{v}}\boldsymbol{b}^e\cdot\boldsymbol{b}^{e-1}\right] + \beta\,\dot{\alpha} + \boldsymbol{\sigma} : \boldsymbol{d} \\
\rho\,c_\theta\,\dot{\theta} &= -\boldsymbol{\sigma} : \boldsymbol{d} - \boldsymbol{\sigma} : \frac{1}{2}\left[\boldsymbol{\pounds}_{\boldsymbol{v}}\boldsymbol{b}^e\cdot\boldsymbol{b}^{e-1}\right] + \beta\,\dot{\alpha} + \boldsymbol{\sigma} : \boldsymbol{d} \\
\rho\,c_\theta\,\dot{\theta} &= \boldsymbol{\sigma} : \boldsymbol{d}^p + \beta\,\dot{\alpha} \\
\rho\,c_\theta\,\dot{\theta} &= \mathcal{D}^{TP}.
\end{aligned} \tag{4.40}$$

It can be observed that the rate of the temperature is driven by the dissipation $\mathcal{D}^{TP}$ derived before. The dissipation contains the plastic stress power and a part coupled to the evolution of the internal variable, which can be interpreted as negative dissipation due to modification in the crystalline lattice of the material. The latter one is then responsible for strain hardening behavior. This part is also known as cold work, which is not transferred into heat.

Nevertheless, a more simple approach to model the dissipation term for computing the temperature increase is very popular and used in a variety of commercial software tools, such as LS-DYNA and ABAQUS, see Hallquist [50], Simulia [135]. The one we follow in this material is based on the works of Taylor and Quinney, compare Taylor and Quinney [145]. The actual dissipation is therein approximated via the dimensionless, so-called Taylor Quinney coefficient γ_{TQ} via

$$\dot{\theta} = \frac{1}{\rho\,c_\theta}\,\mathcal{D}^{\mathrm{TP}} \quad \text{with} \quad \mathcal{D}^{\mathrm{TP}} \approx \gamma_{\mathrm{TQ}}\,\boldsymbol{\sigma} : \boldsymbol{d}^p. \tag{4.41}$$

The same approximation for heatgenerating plastic work can be found in Bröcker [15] and Simo and Miehe [134], to name just a few. Typically the range for γ_{TQ} in the scope of metals is set with 0.8 to 1.0. Notice that for an iso-thermal model, the local temperature is constant.

4.3.5 Numerical integration of evolution equations

For the thermoplastic model, evolution equations for the plastic internal variables were derived. For the implementation into numerical codes, the evolution equations have to be integrated in time. Especially the integration of the plastic flow requires special attention in order to satisfy the incompressibility condition. In the following, we recall the common strategy proposed by Simo [132].

To describe the evolution of the plastic variables in time, we introduce the notation for a time-step of size Δt, which ranges from a discrete time t_n to t_{n+1}. In numerical schemes, t_n usually denotes a present equilibrium solution, while the algorithm attempts to achieve equilibrium for the next time t_{n+1}. To differentiate quantities of different time states, the subscript $(\bullet)_n$ is used to indicate known quantities of t_n, while no subscript always indicates a quantity at t_{n+1}. Applying this concept to the current problem of integration of plastic flow, we assume a multiplicative update of the plastic deformation gradient in time

$$\boldsymbol{F}^p = \int_{t_n}^{t_{n+1}} \tilde{\boldsymbol{L}}^p \; \mathrm{d}t \cdot \boldsymbol{F}^p_n. \tag{4.42}$$

Notice that the temporal update follows the same logic as the one outlined in (7.37). Consistency demands that the update of the plastic part of the deformation gradient, uses the plastic velocity gradient of the intermediate configuration.

The temporal integral is now approximated with an exponential mapping, which is volumepreserving to the plastic flow

$$\boldsymbol{F}^p = \exp(\tilde{\boldsymbol{L}}^p \, \Delta t) \cdot \boldsymbol{F}^p_n. \tag{4.43}$$

Using the multiplicative split of the deformation gradient including the identity $\boldsymbol{F}^{e-1} \cdot \boldsymbol{F}^e = \boldsymbol{I}$, we can also shift the statement to the current configuration, using the push-forward operation from equation (4.19) and the consequence of vanishing plastic spin

$$\begin{aligned}
\boldsymbol{F} &= \boldsymbol{F}^e \cdot \exp(\tilde{\boldsymbol{L}}^p \, \Delta t) \cdot \boldsymbol{F}^{e-1} \cdot \boldsymbol{F}^e \cdot \boldsymbol{F}^p_n \\
&= \exp(\boldsymbol{F}^e \cdot \tilde{\boldsymbol{L}}^p \cdot \boldsymbol{F}^{e-1} \, \Delta t) \cdot \boldsymbol{F}^e \cdot \boldsymbol{F}^p_n \\
&= \exp(\boldsymbol{L}^p \, \Delta t) \cdot \boldsymbol{F}^e \cdot \boldsymbol{F}^p_n \\
&= \exp(\boldsymbol{d}^p \, \Delta t) \cdot \boldsymbol{F}^e \cdot \boldsymbol{F}^p_n.
\end{aligned} \tag{4.44}$$

For the application of an implicit Euler integration algorithm, the plastic variables are fixed. Hence a further reformulation of equation (4.45) is done in order to provide an update for the elastic part of the deformation gradient based on a frozen plastic state $\boldsymbol{F} = \boldsymbol{F}^e \cdot \boldsymbol{F}^p{}_n$ and $\boldsymbol{F}^e = \boldsymbol{F} \cdot \boldsymbol{F}^{p-1}_n$

$$\begin{aligned}
\boldsymbol{F} &= \exp(\boldsymbol{d}^p \, \Delta t) \cdot \boldsymbol{F}^e \cdot \boldsymbol{F}^p_n \\
\boldsymbol{F}^e \cdot \boldsymbol{F}^p_n &= \exp(\boldsymbol{d}^p \, \Delta t) \cdot \boldsymbol{F} \cdot \boldsymbol{F}^{p-1}_n \cdot \boldsymbol{F}^p_n \\
\boldsymbol{F}^e &= \exp(\boldsymbol{d}^p \, \Delta t) \cdot \boldsymbol{F} \cdot \boldsymbol{F}^{p-1}_n \cdot \boldsymbol{F}^p_n \cdot \boldsymbol{F}^{p-1}_n \\
\boldsymbol{F}^e &= \exp(\boldsymbol{d}^p \, \Delta t) \cdot \boldsymbol{F} \cdot \boldsymbol{F}^{p-1}_n.
\end{aligned} \tag{4.45}$$

Hereby it is crucial to recognize that $\boldsymbol{F}^e$ renders the independent variable, assuming the overall deformation $\boldsymbol{F}$ is given.

Now we apply the associated plastic evolution equation (4.34) to achieve the implicit update formula

$$\boldsymbol{F}^e = \exp(\lambda \, \frac{\partial \phi}{\partial \boldsymbol{\sigma}} \, \Delta t) \cdot \boldsymbol{F} \cdot \boldsymbol{F}_n^{p-1}. \tag{4.46}$$

Simultaneously, the time integration of the internal variable α, following the evolution equation (4.34), in an implicit Euler scheme yields

$$\alpha = \int_{t_n}^{t_{n+1}} \dot{\alpha} \; \mathrm{d}t + \alpha_n = \lambda \frac{\partial \phi}{\partial \beta} \, \Delta t + \alpha_n. \tag{4.47}$$

Since the evolution of the plastic flow and the internal hardening variable are coupled by the Lagrangian multiplier, they can be combined, i.e.

$$\begin{aligned} \boldsymbol{F}^e \quad &= \exp((\alpha - \alpha_n) \frac{1}{\Delta t} \left(\frac{\partial \phi}{\partial \beta} \right)^{-1} \, \frac{\partial \phi}{\partial \boldsymbol{\sigma}} \, \Delta t) \cdot \boldsymbol{F} \cdot \boldsymbol{F}_n^{p-1} \\ &= \exp(-(\alpha - \alpha_n) \sqrt{\frac{3}{2}} \mathbb{N}) \cdot \boldsymbol{F} \cdot \boldsymbol{F}_n^{p-1}. \end{aligned} \tag{4.48}$$

In order to arrive at equation (4.48), equation (4.47) was reformulated towards an expression for the multiplier $\lambda = (\alpha - \alpha_n) \frac{1}{\Delta t} \left(\frac{\partial \phi}{\partial \beta} \right)^{-1}$, and the partial derivatives of equation (4.38) are inserted.

Notice that equation (4.48) represents the final algorithmic representation of the implicit time integration of plastic flow, for the herediscussed types of plasticity. Its implementation was subject to multiple publications. Originally proposed by Simo [132], it is very common to use symmetric properties of deformation tensors, e.g., $\boldsymbol{b}^e$ or $\boldsymbol{C}^{p-1}$. Further, a stress function based on logarithmic strains can be applied to avoid the evaluation of the matrix exponent as present in equation (4.48). In order to make use of this, the plastic flow is updated in their spectral decomposition, respectively. This is important, as this implicit scheme yields a nonlinear system of equations, which requires the first-order derivative of the matrix exponent in order to solve it with a consistent tangent. An overview of plastic algorithms following this ansatz can be found, e.g., in Simo and Hughes [133], a specific implementation is discussed in Klinkel [77].

The evolution equation for the temperature (4.41) is integrated in time to achieve the temperature increment within the time-step Δt via

$$\theta = \int_{t_n}^{t_{n+1}} \dot{\theta} \; \mathrm{d}t + \theta_n = \int_{t_n}^{t_{n+1}} \frac{\gamma_{\mathrm{TQ}}}{\rho \, c_\theta} \, \boldsymbol{\sigma} : \boldsymbol{d}^p \, \mathrm{d}t + \theta_n. \tag{4.49}$$

In this thesis, we make use of a temperature dependent yield surface, which causes thermal softening. This softening is known to cause problems in combination with implicit Euler integration, as applied for the plastic flow. For this reason, the temperature evolution is integrated using an explicit scheme. A similar ansatz is followed in Simo and Miehe [134], where additionally, the plastic work was reformulated in terms of the internal plastic variable, i.e.

$$\theta = \frac{\gamma_{\mathrm{TQ}}}{\rho \, c_\theta} \, (\alpha_n - \alpha_{n-1}) \, \beta_n + \theta_n. \tag{4.50}$$

Notice that algorithmically there is no need to store additional history variables. Update formula (4.50) can be executed at the end of a time-step. Consequently, the material uses the last updated temperature θ_n throughout the time-step Δt.

4.3.6 Implementation of non-isothermal elasto-visco-plastic materials

The algorithms used for the implementation in this thesis indeed solve the nonlinear equation system that arises from equation (4.48) directly, using the exact close form solution representation of the matrix exponent as proposed in Hudobivnik and Korelc [65]. Their method uses automatic differentiation in order to provide consistent derivatives at machine precision. For the implementation of constitutive laws, the automatic code generation tool *AceGen* was used. *AceGen* provides a front end for symbolic expressions inside the symbolic programming language *Mathematica* in the equally named software. This setup can be used to generate numerical codes in performant, low level languages such as FORTRAN or C. The advantages of this tool lie in the simultaneous expression optimization, which makes the resulting code extremely efficient, see Korelc [78]. Further, *AceGen* provides automatic differentiation on the symbolic expressions, which are classically error-prone to derive and implement by hand. Further, the finite element software *AceFEM* is used in this thesis in order to provide reference solutions and test the subroutines generated by *AceGen* directly. The combination of material- and finite-element subroutines generated in the same framework leads to well tested, reliable codes, see Korelc [79].

To specifically implement the previously discussed elasto-plastic constitutive response, two function prototypes need to be defined. On the one hand, a stress function based on elastic deformations as given in algorithm 1, and a function for the yield function, as outlined in algorithm 2.

```
Function $\hat{\boldsymbol{\tau}}(\boldsymbol{F}^e)$:
    return $\boldsymbol{\tau}$, $\dfrac{\partial \boldsymbol{\tau}}{\partial \boldsymbol{F}^e}$
```

Algorithm 1: Generic template of a stress function for implementation of constitutive behavior. The actual computation of the stresses with respect to the elastic deformation is material specific.

```
Function $\hat{\phi}(\boldsymbol{\tau}, \alpha)$:
    return $\phi$, $\dfrac{\partial \phi}{\partial \boldsymbol{\tau}}$, $\dfrac{\partial \phi}{\partial \alpha}$
```

Algorithm 2: Generic template of a yield function for the implementation of elasto-plastic materials. The computation of the actual yield function based on the provided arguments may vary.

Notice that implementation into the symbolic environment directly resolves all symbolic dependencies. However, for a better understanding, the dependencies required for the determination of the plastic flow are given here specifically.

```
Input: local variables of previous time-step: $\boldsymbol{F}_n^{p-1}, \alpha_n$
current deformation gradient: $\boldsymbol{F}$
Output: $\boldsymbol{F}^{p-1}$, $\alpha$
/* check for violation of the yield surface
under the assumption of an elastic step                                  */
$\boldsymbol{F}^e \leftarrow \boldsymbol{F} \cdot \boldsymbol{F}_n^{p-1}$
$\boldsymbol{\tau} \leftarrow \hat{\boldsymbol{\tau}}(\boldsymbol{F}^e)$, $\phi \leftarrow \hat{\psi}(\boldsymbol{\tau}, \alpha_n)$
if $\phi <= 0$ then
    /* elastic response                                                  */
    $\boldsymbol{F}^{p-1} \leftarrow \boldsymbol{F}_n^{p-1}, \; \alpha \leftarrow \alpha_n, \; \frac{\partial \boldsymbol{F}^{p-1}}{\partial \boldsymbol{F}} \leftarrow \boldsymbol{0}, \; \frac{\partial \alpha}{\partial \boldsymbol{F}} \leftarrow \boldsymbol{0}$
else
    /* elasto-plastic response                                           */
    initialize local vector of unknwons (i=0):
    $\boldsymbol{d}^i \leftarrow \left[F^{p-1}_{n\,11}, F^{p-1}_{n\,12}, F^{p-1}_{n\,13}, F^{p-1}_{n\,21}, F^{p-1}_{n\,22}, F^{p-1}_{n\,23}, F^{p-1}_{n\,31}, F^{p-1}_{n\,32}, F^{p-1}_{n\,33}, \alpha_n\right]^T$

    /* solve nonlinear local system of equation:                         */

    $\boldsymbol{r}(\boldsymbol{d}^i) = \begin{bmatrix} \boldsymbol{F} \cdot \boldsymbol{F}^{p-1}(d_{1-9}) - \exp(-(d_{10} - \alpha_n)\sqrt{\frac{3}{2}}\mathbb{N}(d_{1-9})) \cdot \boldsymbol{F} \cdot \boldsymbol{F}_n^{p-1} \\ \phi(\boldsymbol{d}^i) \end{bmatrix} \overset{!}{=} \boldsymbol{0}$

    /* using a local Newton scheme:                                      */
    for $i \leftarrow 0$ to $30$ do
        $\boldsymbol{K}^i(\boldsymbol{d}^i) \leftarrow \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{d}}(\boldsymbol{d}^i)$
        $\Delta\boldsymbol{d} = -\boldsymbol{K}^{i^{-1}} \cdot \boldsymbol{r}(\boldsymbol{d}^i)$

        if $\|\Delta\boldsymbol{d}\| <= tol$ then
            /* compute additional dependencies                               */
            $[\partial_{\boldsymbol{F}} \boldsymbol{d}^i] \leftarrow -\boldsymbol{K}^{i^{-1}} \cdot \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{F}}(\boldsymbol{d}^i)$
            /* quit iterative procedure                                      */
            $\boldsymbol{F}^{p-1} \leftarrow d_{1-9}, \; \alpha \leftarrow d_{10}$
            $\frac{\partial \boldsymbol{F}^{p-1}}{\partial \boldsymbol{F}} \leftarrow [\partial_{\boldsymbol{F}} \boldsymbol{d}^i]_{1-9,1-9,1-9}, \; \frac{\partial \alpha}{\partial \boldsymbol{F}} \leftarrow [\partial_{\boldsymbol{F}} \boldsymbol{d}^i]_{10,1-9,1-9}$
        else
            /* update and continue iterative solution procedure:             */
            $\boldsymbol{d}^{i+1} \leftarrow \boldsymbol{d}^i + \Delta\boldsymbol{d}$
        end
    end

end
return $\boldsymbol{F}^{p-1}, \; \alpha, \; \frac{\partial \boldsymbol{F}^{p-1}}{\partial \boldsymbol{F}}, \; \frac{\partial \alpha}{\partial \boldsymbol{F}}$
```

Algorithm 3: Algorithm for the determination of plastic flow for J_2 plasticity. In the case of an elasto-plastic response, the locally coupled system of equations arising from (4.48) is solved using a Newton-Raphson scheme. For clarity, the temporary iterative variables are highlighted in blue.

Algorithm 3 represents the plastic part of a constitutive subroutine, with primary variables being the deformation gradient. Its output needs to be the updated local variables. In order to be used in a globally implicit solution scheme, also their dependencies with respect to the primary variable, here $\boldsymbol{F}$, are part of the function output. The overall evaluation of constitutive equations is described in algorithm 4. This algorithm is also valid for a purely elastic response as for the hyper-elastic material from 4.2, where the deformation gradient is purely elastic a priorly. Also, in the case of rate-independent and isothermal elastic- or elasto-plastic materials, the concerning updates can simply be skipped.

For some illustrative purposes, the FEM is used in this thesis. In these cases, the same material subroutines are used as described here for the MPM. Due to the automatic differentiation capabilities in *AceGen*, it is straightforward to provide the overall consistent material tangents as required in implicit solution schemes. While most dependencies are visible for the automatic differentiation, the algorithmic treatment in determining the plastic flow (algorithm 3) requires to explicitly computing some dependencies. The consistent algorithmic tangents can be derived, by *AceGen*.

```
Input:
    $\boldsymbol{m}$ set of material parameter and system parameter e.g. $\Delta t$
    $\boldsymbol{F}$ current deformation gradient
    $\boldsymbol{h}_n$ set of local material history (previous state)
Output:
    $\boldsymbol{\sigma}$ Cauchy stress tensor
    $\boldsymbol{h}$ set of updated local material history

/* determine elastic part of the deformation gradient                */
$\boldsymbol{F}^{p-1}, \alpha, \leftarrow$<evaluate algorithm 3> $(\boldsymbol{F}, \boldsymbol{F}_n^{p-1}, \alpha_n)$
$\boldsymbol{F}^e \leftarrow \boldsymbol{F} \cdot \boldsymbol{F}_n^{p-1}$, $J \leftarrow \det \boldsymbol{F}$

/* compute stress response                                           */
$\boldsymbol{\tau} \leftarrow \hat{\boldsymbol{\tau}}(\boldsymbol{F}^e)$
$\boldsymbol{\sigma} \leftarrow \frac{1}{J} \boldsymbol{\tau}$

/* update plastic rate and temperature                               */
$\dot{\alpha} \leftarrow \frac{\alpha - \alpha_n}{\Delta t}$
$\theta \leftarrow \frac{\gamma_{\text{TQ}}}{\rho\, c_\theta} (\alpha - \alpha_n)\, \sigma_{\texttt{VM}} + \theta_n$
return $\boldsymbol{\sigma}$, $\boldsymbol{h}$
```

Algorithm 4: Algorithmic flow to determine the material response.

4.3.7 Rate-independent J$_2$-plasticity

For some applications, an isothermal rate-independent plasticity can be used. In this thesis, such materials are implemented using the previously discussed elasto-plastic framework at finite deformations and specifying the yield function

$$\phi(\boldsymbol{\tau}) = \tau_{\mathtt{VM}} - \tau_y(\alpha), \qquad \text{with} \qquad \tau_y(\alpha) = \tau_\infty + (\tau_0 - \tau_\infty)\,\exp(-\eta\,\alpha) + h\,\alpha \tag{4.51}$$

which is formulated in the Kirchhoff stresses for compatibility reasons. Notice that the formally introduced stresslike internal variable β, is now identified as a current yield strength $\tau_y(\alpha)$ in $\mathrm{N\,m^{-2}}$. The presented form contains a set of material parameters, such as the initial yield strength τ_0 in $\mathrm{N\,m^{-2}}$, a saturated yield strength τ_∞ in $\mathrm{N\,m^{-2}}$, a dimensionless parameter for exponential strain hardening η and a linear strain hardening modulus h, in $\mathrm{N\,m^{-2}}$. For a better understanding, the yield strength function is plotted in figure 4.3 over the accumulated plastic arc length α.

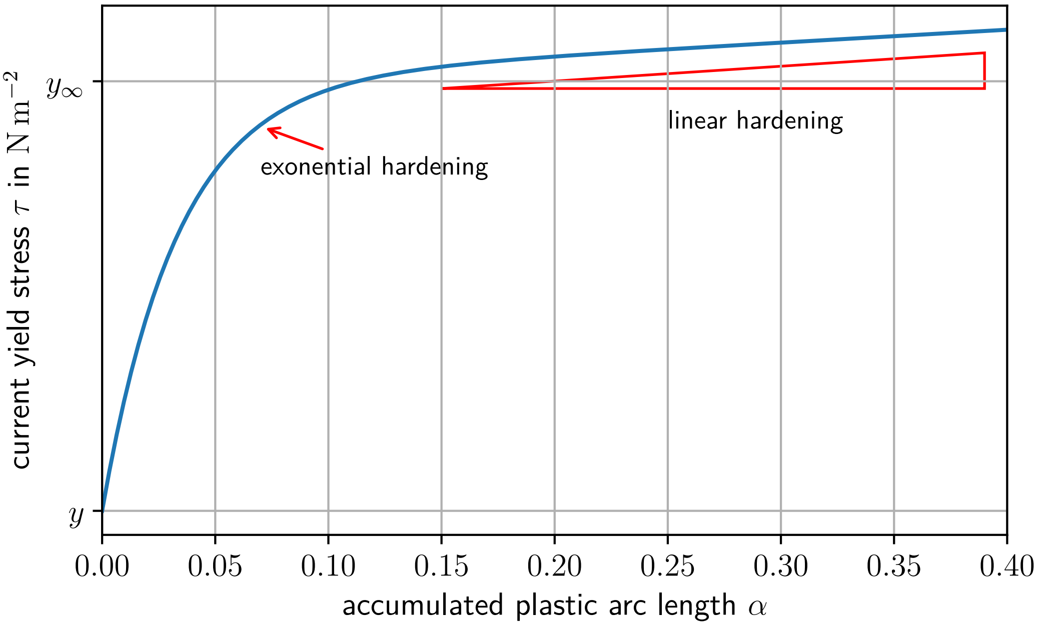


Figure 4.3: Visualisation of the rate-independent yield strength (4.51).

4.3.8 Johnson-Cook J$_2$-plasticity

For application at varying strain rates, the influence of plastic strain rates should be considered. Additionally, these applications are usually fast processes that induce plastic heat. The Johnson-Cook material law accounts for these effects, and it formulates yield stress based on the current temperature θ and the current plastic rate $\dot{\alpha}$. The original version was proposed in Johnson and Cook [70; 71], and reads

$$\tau_{y,JC}(\alpha,\,\dot{\alpha},\,\theta) = \left[\mathcal{A} + \mathcal{B}\,\alpha^{\mathcal{N}}\right]\left[1 + \mathcal{C}\,\ln\frac{\dot{\alpha}}{\dot{\alpha}_0}\right]\left[1 - \left(\frac{\theta - \theta_0}{\theta_m - \theta_0}\right)^{\mathcal{M}}\right]. \tag{4.52}$$

It can be observed that the Johnson-Cook flow stress $\tau_{y,JC}$ is composed of three terms and relies on five new material parameters. These are $\mathcal{A}$ in $\mathrm{N\,m^{-2}}$, $\mathcal{B}$ in $\mathrm{N\,m^{-2}}$, $\mathcal{N}$, $\mathcal{C}$, and $\mathcal{M}$. The first term in square brackets represents the current flow stress of the material at a quasi-static condition, without any temperature influence. Hereby $\mathcal{A}$ can be identified to represent the initial yield strength of the material, $\mathcal{B}$ is an isotropic hardening modulus, and the exponent $\mathcal{N}$ is used to describe nonlinear hardening. The parameter $\mathcal{N}$ is used to describe work hardening behavior of a material, as shown, e.g., in Huang et al. [63]. The second term acts as a rate dependency modifier, where parameter $\mathcal{C}$ is used to describe the reversible hardening effect at high strain rates observed in experiments. The last term further scales the yield strength by means of temperature weakening. Notice that at the melting temperature of the material θ_m, the material has no yield strength anymore. The parameter $\mathcal{M}$ is hereby used to define the evolution of yield stress in between. For a better understanding, figure 4.4 visualize the effects. The terms of equation (4.52) are plotted individually, to discuss the parameters on the model.

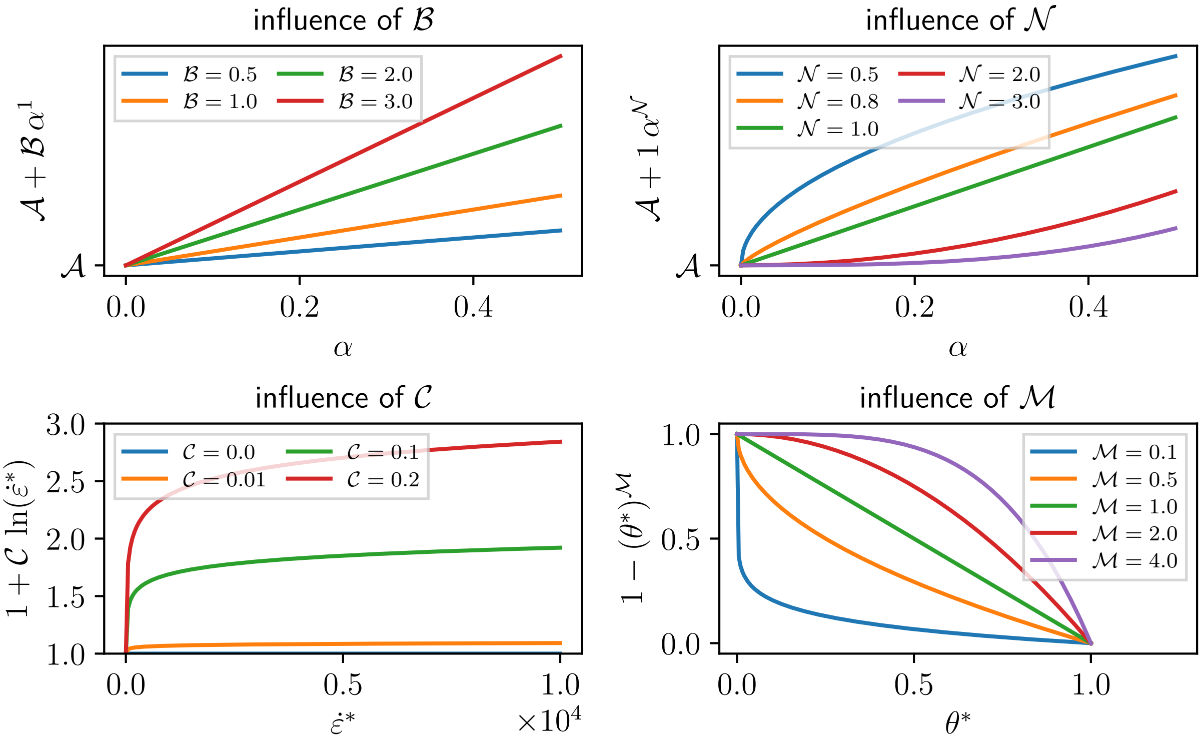


Figure 4.4: Plots of the individual terms of the Johnson-Cook yield strength (4.52), to visualize the effects of the five material parameters in the model.

It is worth mentioning that the Johnson-Cook model was originally designed to be fitted with least effort while reproducing experiments to a satisfying degree. For this reason, the model is widely used in commercial codes and research papers. The Johnson-Cook model was compared to other available constitutive models in Hor et al. [58]. The model is widely used for the simulation of cutting processes, e.g., in the particle finite element method, see Carbonell et al. [19] and also the material point method, see Ambati et al. [7], Huang et al. [63]. Its capabilities, especially at high strain rates, were further used in impact problems, as in Manes et al. [97]. Determination, especially of the strain rate parameter $\mathcal{C}$, is usually done using steady-state tests in combination with dynamic tests, such as the Split-Hopkinson-Pressure-Bar (SHPB) experiment as done in Zhao and Gary [164] or Manes et al. [96], but also using inverse methods such as described in Ning and Liang [106].

In the scope of this thesis, the Johnson-Cook model of equation (4.52) is implemented in the framework described in section 4.3.6. To benchmark, the rate and temperature dependency, finite element analysis of a homogeneous tensile test are conducted. Considered is a cubic specimen of unit size (1 m×1 m×1 m) under boundary conditions that allow for unconstrained deformation and hence a constant deformation and stress state over the whole specimen. Further, an engineering strain ε_{xx} is applied to the specimen via boundary conditions. The numerical analysis uses the Johnson-Cook model with material parameters given in table 4.1.

elastic modulus	E	$2.07 \times 10^{11}\,\mathrm{N\,m^{-2}}$
material density	ρ_0	$7830\,\mathrm{kg\,m^{-3}}$
Poisson's ratio	ν	0.3
Tayor-Quinney coefficient	γ_{TQ}	0.0
initial yield strength	$\mathcal{A}$	$792 \times 10^{6}\,\mathrm{N\,m^{-2}}$
strain hardening modulus	$\mathcal{B}$	$510 \times 10^{6}\,\mathrm{N\,m^{-2}}$
work hardening modulus	$\mathcal{N}$	0.26
plastic rate dependency parameter	$\mathcal{C}$	0.014
temperature dependency parameter	$\mathcal{M}$	1.03
reference plastic rate	$\dot{\varepsilon}_0^{pl}$	$1.0\,\mathrm{s^{-1}}$
reference temperature	θ_0	293.15 K
melting temperature	θ_m	1573.15 K
specific heat capacity	c_θ	$480\,\mathrm{J\,kg^{-1}\,K^{-1}}$
heat expansion coefficient	α_σ	$1.35 \times 10^{-5}\,\mathrm{K^{-1}}$

Table 4.1: Material parameter for AISI 4340 Steel, taken from Ambati et al. [7].

For the numerical analysis, ε_{xx} is applied at a constant rate $\dot{\varepsilon}_{xx}$. This way the effects of plastic strain rate can be observed by two speeds of the deformation. After maximum loading of $\varepsilon_{xx} = 0.6$ is reached, the load stays constant for some time to verify the drop of plastic deformation and its consequences to the yield strength. The simulations are further conducted without temperature increase, but for two constant temperatures, respectively. This way, the temperature influence can be studied more easily. The resulting stress strain diagrams of that analysis are given in figure 4.5.

It can easily be concluded that the material shows larger stress amplitudes at higher strain rates. Also, with more temperature applied, the stress response is weaker. The smaller difference due to rate increase at the lower temperature tests compared to the distance observed for room temperature can be explained in the structure of the Johnson-Cook yield stress. The influence terms act multiplicatively, and thus the rate-dependency scales with the rate independent part. It can also be seen that the von Mises stresses exceed a drop at the end of the strain spectrum, which is reasonable, as the load is kept constant at the end of the analysis. The rate modifying term consequently drops down to one.

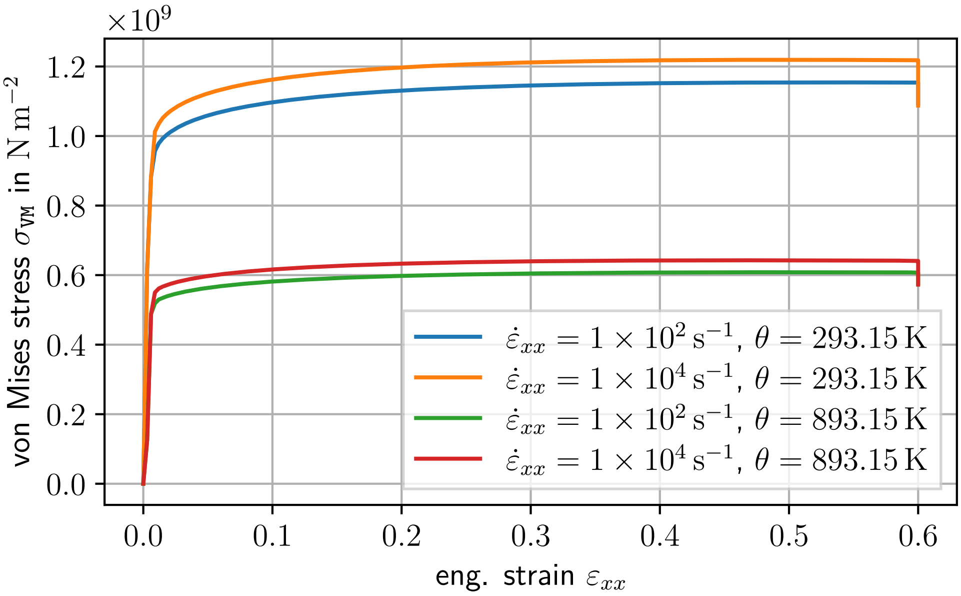


Figure 4.5: Results of a numerical tensile test with the Johnson-Cook material at different temperatures and strain rates.

5 Numerical time integration

Temporal integration occurs in the context of solving differential equations frequently. Numerical time integration is thus of great importance. There exist many time integration schemes of various complexity, computational efficiency, and accuracy. In this chapter, three kinds of time integration are introduced and used in the scope of this thesis. The goal of all schemes is to provide values (displacements, accelerations, velocities) as the result of a time-step, based on the history of the system. We assume that the history is generally known, as outlined already in chapter 4.3.5. The result of a time-step is characterized by static or dynamic equilibrium. At this point, we recall the notation for discrete time-steps t_n representing previous equilibrated time, while the algorithm attempts to achieve equilibrium for the next time t_{n+1}. The time-step size is denoted by Δt. To differentiate quantities of these time states, the subscript $(\bullet)_n$ is used to indicate known quantities of t_n, while no subscript always indicates a quantity at t_{n+1}.

In this section we aim to provide insights inot different time integration schemes in a solid mechanics framework. That reflects especially in wether the scheme is implicit or explicit. Implicit schemes are generally more stable, but a tangent to the system force is required. Explicit schemes offer update formulas to compute the solution of a time-step based on a constant system force. Also of interest is the data requirement, i.e., regarding previous values that need to be stored for the integration.

We discuss the schemes at a mass-spring problem, which allows for a discussion with respect to stability and energy conservation.

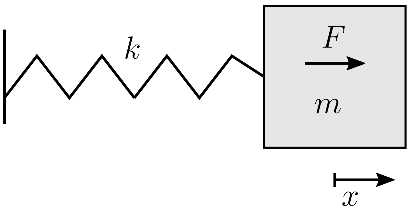


Figure 5.1: Illustration of the mass-spring problem.

We observe the motion of the single mass $m = 1$ in time from $t = 0$ to $t = 50$ discretized by steps of a constant size Δt. The force acting on the mass due to the attached spring is a function of its position and velocity $F(x, \dot{x}) = -k\,x$. Hereby we introduce the system stiffness $k = 1$. Note that while f formally depends on $\dot{x}$, we skip the actual influence of the velocity here, which typically results in damping effects. The initial configuration renders a position of $x_0 = 1$ and a velocity $v_0 = 0$. Notice that the discretized problem uses variables x, v, and a for the mass position, velocity, and acceleration. Further, we compute the system energy consisting of the kinetic- and strain energy as $E = \frac{1}{2}\,m\,\dot{x}^2 + \frac{1}{2}\,k\,x^2$.

5.1 Explicit Euler

The explicit Euler or forward Euler time integration is based on a simple first-order approximation in time,

$$\begin{aligned} x &= x_n + v_n\,\Delta t \\ v &= v_n + a_n\,\Delta t \end{aligned} \tag{5.1}$$

leaving a second-order error $\mathrm{O}(\Delta t^2)$ with respect to the step size Δt. The scheme is illustrated in figure 5.2.

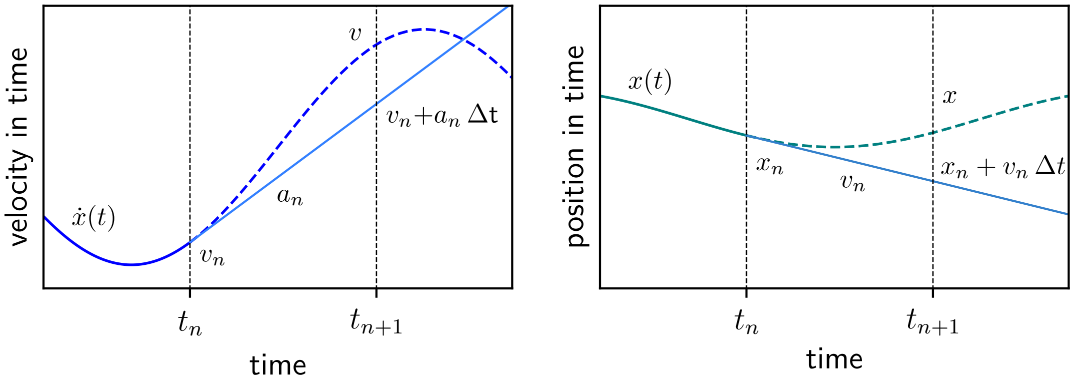


Figure 5.2: Illustration of the explicit Euler time integration. Both position and velocity are individually updated using their tangent at the known state.

The acceleration is computed based on the force from the last known equilibrium.

$$a_n = \frac{1}{m}\,F(x_n,\,v_n,\,t) \tag{5.2}$$

That leaves the requirements of this scheme to store positions and velocities at the end of the last time-steps. However, the great benefit of this method is that position and velocity both depend on known values and are not coupled. Also, no tangent to the system force is required. Notice that in order to compute the solution at time t_n no solution needs to be solved, but rather evaluate formulas, which characterizes this scheme as an explicit one.

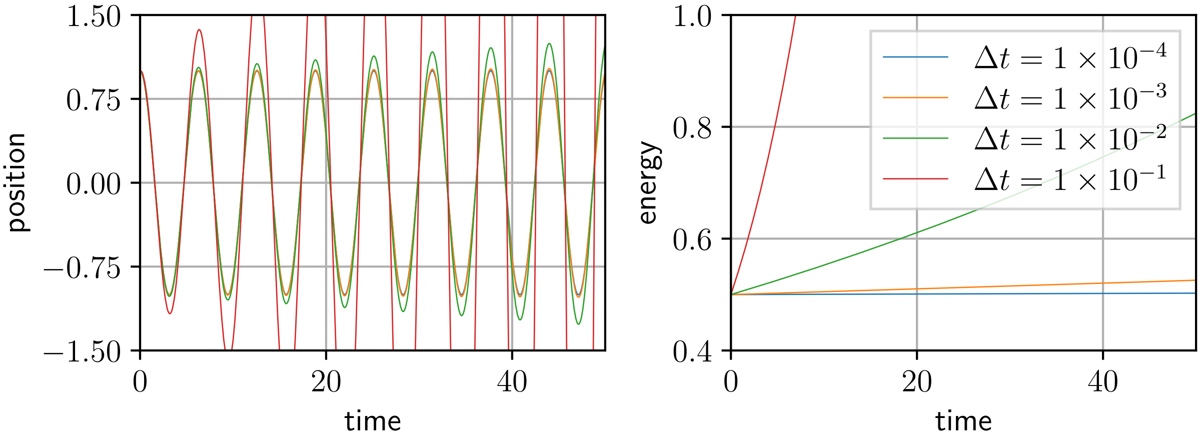


Figure 5.3: Solution of the mass-spring problem with the explicit Euler scheme. Given are the position and energy over time for different time-step sizes.

Simulation results for the mass-spring problem in terms of the explicit Euler scheme are given in figure 5.3. Here the problem was analyzed for varying time-step sizes. The position plot shows rapidly raising amplitudes for the position and raising system energy for larger step sizes. The system energy should stay constant but raises nonlinearly as expected from the error of order $O(\Delta t^2)$. Consequently, the scheme is quite sensitive to Δt and remains stable only for relatively small values.

5.2 Leapfrog Algorithm / Velocity-Verlet

The leapfrog algorithm, also known as Verlet- or Velocity-Verlet time integration, represents another explicit scheme. Unlike the Explicit Euler, which uses a first-order approximation, this method is based on a second-order expansion. Additionally, the time discretization now includes virtual half-step solutions $(\bullet)_{n+1/2}$. It is possible to gain similarly simple and computationally expensive formulas as the Euler scheme but provide a higher level of stability.

$$\begin{aligned} a_n &= \frac{1}{m} F(x_n,\, v_{n-1/2},\, t) \\ v_{n+1/2} &= v_{n-1/2} + a_n\, \Delta t \\ x &= x_n + v_{n+1/2}\, \Delta t \end{aligned} \tag{5.3}$$

Using the old half step-velocity $v_{n-1/2}$ as the argument in the force function, does not critically spoil the overall error in this method. In fact, the error is still in the order of $O(\Delta t^3)$. Notice that this method does not use or compute any full-step velocities but uses the half step values instead. However, the leapfrog algorithm does require the initial half-step velocity to be known. Usually, this is not provided but can be estimated by

$$v_{0-1/2} = v_0 - \frac{1}{2}\, a_0\, \Delta t. \tag{5.4}$$

The formula does imply one additional step backward in time before integrating forward.

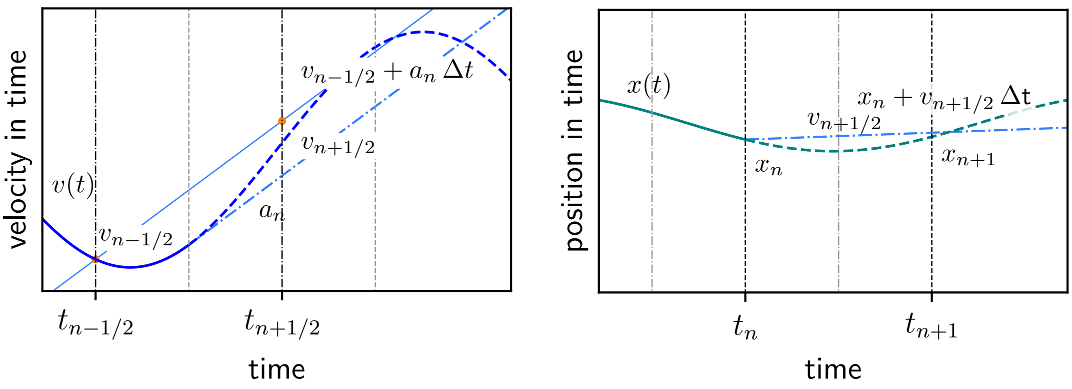


Figure 5.4: Illustration of the Velocity-Verlet time integration. At first the half step velocity is updated based on the accelerations at last known state. This velocity then acts as a tangent to update the position.

Numerical results for the mass-spring problem using the same parameter set as for the Explicit Euler scheme are given in figure 5.5. One can see the advantage of this scheme over the aforementioned immediately, compare, figure 5.3. Even for $\Delta t = 0.1$, where the Explicit Euler scheme quickly diverges, the leapfrog algorithm remains stable. Notice that the energy plot provides the same scale as that in figure 5.3. Despite some oscillation, even for $\Delta t = 0.1$, the leapfrog scheme is energy conserving.

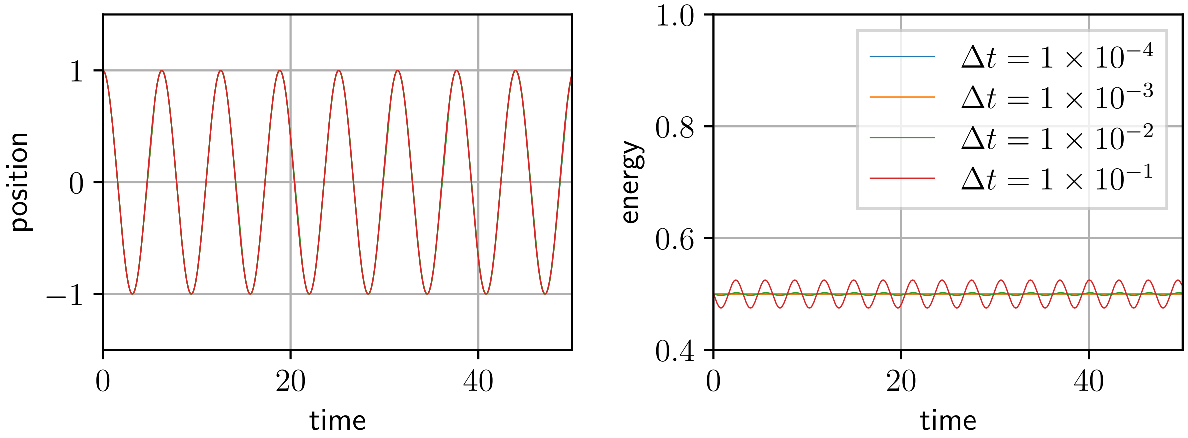


Figure 5.5: Solution of the mass-spring problem with the leapfrog scheme. Given are the position and energy over time for different time-step sizes. The results for position over time (left) align almost perfectly, only for the largest time step an energy oscillation (right) can be identified at these scales.

5.3 Implicit Euler

The implicit Euler method again relies on a first-order approximation, similar to the one used in the equation (5.1). In this case, however, the tangent used to update the velocity and position is not computed from known values but from the result itself,

$$\begin{aligned} x &= x_n + v\,\Delta t \\ v &= v_n + a\,\Delta t \end{aligned} \quad . \tag{5.5}$$

Note that the system force implicitly depends on the solution of the time-step $a(x^{n+1}, v^{n+1})$ and thus equation (5.5) represents a coupled system. It is not possible to compute the velocity and then the position as for the aforementioned methods. To provide a monolithic solution scheme, we choose for one of the motion variables, position, acceleration, or velocities, to be the primary one which can be expressed as a function of the other. Here we choose for velocities.

$$\begin{aligned} & v = v_n + a\,\Delta t \\ \Leftrightarrow \quad & v = v_n + a(x(v),\, v,\, t)\,\Delta t \\ \Leftrightarrow \quad & v = v_n + \frac{\Delta t}{m}\, f(x(v),\, v,\, t) \end{aligned} \tag{5.6}$$

Equation 5.6 requires additional information of the system force to solve, i.e., its consistent tangent to one of its arguments. Computation of this tangent is generally not straightforward and might be computationally expensive, representing one of the major downsides for an implicit time integrator. Also, if the system force provides nonlinearities for one of its arguments, solving equation (5.6) becomes more complicated and probably

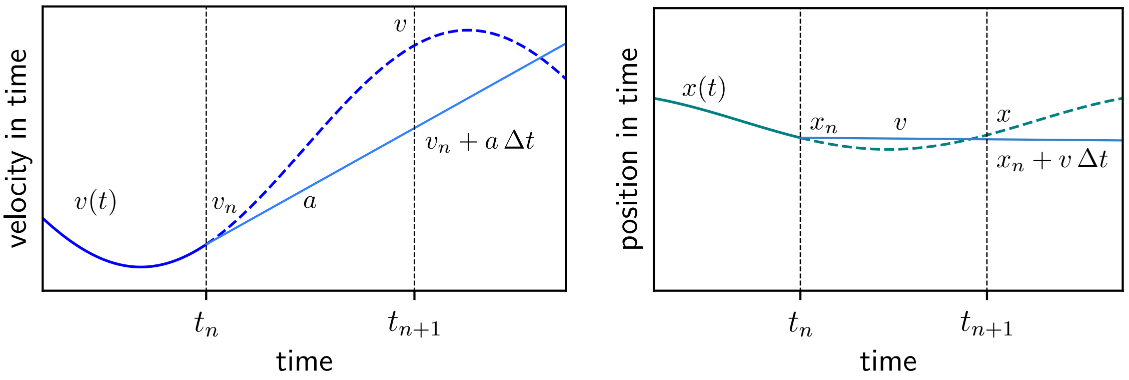


Figure 5.6: Illustration of the implicit Euler time integration. Here the implicit character is given by using the unknown acceleration to update the velocities.

an iterative scheme. For these reasons, one time-step in an implicit scheme is generally more expensive and less straightforward than one time-step of an explicit time integrator. The system force is linear in the mass-spring problem, and hence with the solution of the implicit equation (5.6), the update formulas appear as

$$\begin{aligned} v &= \left(1 + \frac{k\,\Delta t^2}{m}\right)^{-1} \left(v_n - \frac{k\Delta t}{m} x_n\right) \\ x &= x_n + v\,\Delta t. \end{aligned} \tag{5.7}$$

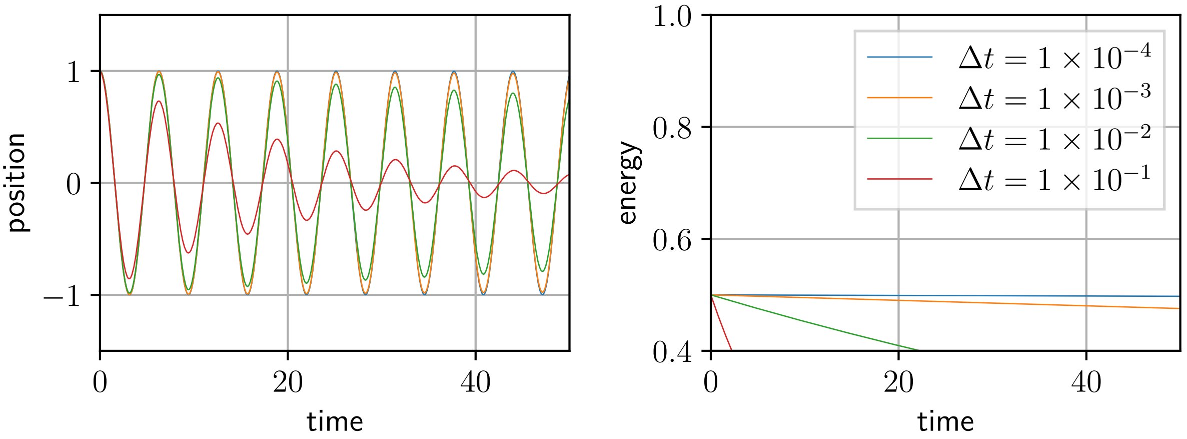


Figure 5.7: Solution of the mass-spring problem with the implicit Euler scheme. Given are the position and energy over time for different time-step sizes. A different position over time behaviour can not be identified for the smallest time-step sizes.

The results of figure 5.7 are obtained using equation (5.7) in a numerical scheme. It can be concluded that as for the implicit Euler scheme in figure 5.7, the first-order approximation leads to a noticeable error accumulation with quadratic order. However, where the explicit scheme is thermodynamically inconsistent in producing energy, this variant is rather dissipative. However, this behavior might be preferred in numerical schemes as it is beneficial regarding numerical stability.

6 Implicit-Explicit solution schemes

In order to solve differential equations numerically, various solution procedures may be applied. Most equations of concern are time-dependent, such as the balance of momentum or the balance of energy, compare equations (3.23,3.30). Generally, two strategies exist. Such differential equations can be solved in an explicit or implicit manner. These terms were used in the previous chapter in time integration schemes and share the same implications in the present context. For a discussion, we propose that a discretized boundary value problem can be described by a generally nonlinear operator $\boldsymbol{R}$ of its free/unknown parameters $\boldsymbol{d}$. The solution at any point in time is then defined by

$$\boldsymbol{R}(\boldsymbol{d}) = \boldsymbol{0}. \tag{6.1}$$

Considering time dependent processes, we allow for an alternative representation of equation (6.1)

$$\boldsymbol{M} \cdot \ddot{\boldsymbol{d}} + \boldsymbol{C} \cdot \dot{\boldsymbol{d}} + \boldsymbol{K} \cdot \boldsymbol{d} - \boldsymbol{P} = \boldsymbol{0}, \tag{6.2}$$

by introducing nonlinear matrix operators $\boldsymbol{M}$, $\boldsymbol{C}$ and $\boldsymbol{K}$ which are often denoted as global mass- damping- and stiffness matrix. Notice that this notation implies physical associations which are only reasonable in the scope of the balance of momentum. However, this form is widely used, see for example WRIGGERS [158]. Besides the balance of momentum the present form can be derived for any underlying differential equation. Further, it also holds for non-time-dependent variants, in which cases only the global stiffness matrix is a nonzero operator.

Considering the notation for temporal discretization, where no subscript refers to the current time, and the subscript $(\bullet)_n$ denotes known quantities of the previous discrete point in time, it is clear that equations (6.1, 6.2) define an equilibrium state for the current time-step.

For clarity, the general dependencies of the introduced quantities are given with

$$\begin{gathered} \boldsymbol{M}(t, \boldsymbol{d}, \dot{\boldsymbol{d}}, \ddot{\boldsymbol{d}}, t_n, \boldsymbol{d}_n, \dot{\boldsymbol{d}}_n, \ddot{\boldsymbol{d}}_n, ...), \quad \boldsymbol{C}(t, \boldsymbol{d}, \dot{\boldsymbol{d}}, \ddot{\boldsymbol{d}}, t_n, \boldsymbol{d}_n, \dot{\boldsymbol{d}}_n, \ddot{\boldsymbol{d}}_n, ...), \\ \boldsymbol{K}(t, \boldsymbol{d}, \dot{\boldsymbol{d}}, \ddot{\boldsymbol{d}}, t_n, \boldsymbol{d}_n, \dot{\boldsymbol{d}}_n, \ddot{\boldsymbol{d}}_n, ...), \quad \boldsymbol{P}(t), \end{gathered} \tag{6.3}$$

which highlights the nonlinear character of equations (6.1,6.2).

As described in chapter 5, it is very often possible to describe the rates of the unknowns as functions of each other. Describing $\dot{\boldsymbol{d}}(\boldsymbol{d}, \boldsymbol{d}_n, \dot{\boldsymbol{d}}_n, \ddot{\boldsymbol{d}}_n ...)$ and $\ddot{\boldsymbol{d}}(\boldsymbol{d}, \boldsymbol{d}_n, \dot{\boldsymbol{d}}_n, \ddot{\boldsymbol{d}}_n ...)$ which leaves the vector of unknowns $\boldsymbol{d}$ as the only unknowns to be solved for, is commonly done.

In the case of an implicit solution scheme, equation (6.1) is solved for the current time-step. This does require solving a coupled system of equations. Further, in the case of nonlinearities, an iterative procedure may be required. Most commonly used is a Newton-Raphson scheme, where the coupled system of equations is formed from the first-order Taylor expansion of equation (6.1)

$$\boldsymbol{R}(\boldsymbol{d}) + \boldsymbol{K}^{\text{eff}}(\boldsymbol{d}) \cdot \Delta \boldsymbol{d} = \boldsymbol{0}, \tag{6.4}$$

where the global effective stiffness matrix $\boldsymbol{K}^{\text{eff}}(\boldsymbol{d})$ is computed from the total differential $\dfrac{\mathrm{d}\,\boldsymbol{R}}{\mathrm{d}\,\boldsymbol{d}}$ and $\Delta \boldsymbol{d}$ denotes an increment of the vector of unknowns. Especially in the framework of variational calculus, $\boldsymbol{K}^{\text{eff}}$ is often referred to as the consistent linearization. With respect to the form of equation (6.2) it is clear that $\boldsymbol{K}^{\text{eff}}$ contains contributions from the nonlinear operators $\boldsymbol{M}$, $\boldsymbol{C}$, and $\boldsymbol{K}$. Equation (6.4) represents a coupled system of equations that is linear in $\Delta\boldsymbol{d}$. Its solution must be computed within each iteration within a Newton-Raphson scheme. A standard form of the algorithm is given in algorithm 5, respectively.

The implicit procedure provides a perfect equilibrium solution for each time-step. Furthermore, it allows for descent time-step sizes. In the case of ill-conditioned systems, it might be hard to find a solution. It is worth mentioning that it is computationally expensive to compute a solution considering the iterative procedure and the required memory for the global effective stiffness matrix and potential overhead for its solution. As the last point, even with automatic differentiation tools such as e.g. *AceGen* in hand, it might be difficult to provide the consistent linearization to complex algebraic equations which are required for an effective solution. In short, an implicit scheme best suits, where large time-steps can be applied without losing important information or where even steady-state solutions are required. In these cases, the rather expensive solution procedure is compensated for the few required time-steps to compute.

The main idea of explicit solution schemes is to assume that equation (6.2) only linearly depend on the vector of unknowns. This follows the forward time integration concepts. In this case, the representation from equation (6.2) is modified

$$\boldsymbol{M}_n \cdot \ddot{\boldsymbol{d}} + \boldsymbol{C}_n \cdot \dot{\boldsymbol{d}} + \boldsymbol{K}_n \cdot \boldsymbol{d} - \boldsymbol{P} = \boldsymbol{0}, \tag{6.5}$$

with the matrix operators evaluated at the last known time

$$\boldsymbol{M}(t_n, \boldsymbol{d}_n, \dot{\boldsymbol{d}}_n, \ddot{\boldsymbol{d}}_n, ...), \quad \boldsymbol{C}(t_n, \boldsymbol{d}_n, \dot{\boldsymbol{d}}_n, \ddot{\boldsymbol{d}}_n, ...), \quad \boldsymbol{K}(t_n, \boldsymbol{d}_n, \dot{\boldsymbol{d}}_n, \ddot{\boldsymbol{d}}_n, ...). \tag{6.6}$$

The arising coupled equation system is linear with respect to the vector of unknowns, eliminating the need for an iterative solution procedure. In the special case of the balance of momentum, it is common to drastically apply further modifications to drastically enhance the efficiency of a numerical scheme. In this scope, the vector of unknowns contains displacements, velocities, and accelerations. The starting point of the modifications is to treat the accelerations as primary variables. Additionally, the displacements and velocities are taken from the last converged state, manifesting the concept of forward time integration further. Displacements and velocities are then computed from the accelerations using an explicit update procedure, e.g., an explicit Euler (compare equation 5.1). The accelerations are computed from the solution of the linear system

$$\boldsymbol{M}_n \cdot \ddot{\boldsymbol{d}} + \boldsymbol{C}_n \cdot \dot{\boldsymbol{d}}_n + \boldsymbol{K}_n \cdot \boldsymbol{d}_n - \boldsymbol{P} = \boldsymbol{0}. \tag{6.7}$$

It is worth mentioning, that the satisfaction of equation (6.1) is usually not achieved by this solution scheme. Rather, an error is expected, which is coupled to the choice of the explicit time integration as discussed in chapter 5. It can be observed that this error can be controlled by using very small time-step sizes. Nevertheless, the error accumulates in time. It must be taken into account that the linear system (6.7) must be solved very often due to time-step size limitations.

It can be assumed that for many discretization techniques, the global mass matrix is constant in time for solid mechanics, considering equation (6.7). Also, a typical mass matrix is positive definite and of small band width, which is very beneficial for solvers of linear systems. One can take advantage of this property by providing, e.g., an LU-decomposition of the mass matrix which represents the coupled linear system. It can then be used throughout the time-steps, which replaces the effort for solving the linear system with vector, matrix multiplications. Notice that with this strategy, the quality of the solution is not affected.

However, another strategy is often chosen, called mass lumping. Here, the discretization is modified such that the mass matrix only has non-zero entries on the diagonal. The linear system is uncoupled. Such systems can be solved extremely efficiently with respect to memory consumption and computation time. This modification leads to errors in the approximation of higher-order modes of the simulated solid. As the analysis of these modes is usually not subject to structural analysis, lumped mass matrices are very popular.

It is worth mentioning that the described explicit solution strategies are also capable of finding steady-state solution, even if dynamics are completely not considered in the discretization. In order to still work, an explicit time integration is chosen as a mathematical tool, while also a suitable mass matrix is chosen. Then, using mass scaling, the solution may be obtained, see for example Schäuble [124]. Often, artificial damping is included, to stabilize the system while it is integrated towards the intended steady-state solution.

While there exist various versions of explicit schemes, a potential algorithm is not given here. A detailed few on the explicit scheme used in this thesis are discussed in the upcoming chapters.

```
Input:           $\boldsymbol{d}_n$  vector of unknowns from previously converged state
          $i_{\max}$  maximum number of iterative steps
 $O(\Delta\boldsymbol{d}, \boldsymbol{d})$  suitable abort criterion
Output:          $\boldsymbol{d}$  all unknowns from equilibrium of current time-step
/* initialize primary variables:                                   */
$\boldsymbol{d}^0 \leftarrow \boldsymbol{d}_n$
/* iterative solution procedure:                                  */
for $i \leftarrow 0$ to $i_{max}$ do
    /* form linear system                                         */
    $\boldsymbol{K}^{\text{eff}} \leftarrow \boldsymbol{K}^{\text{eff}}(\boldsymbol{d}^i)$
    $\boldsymbol{R} \leftarrow \boldsymbol{R}(\boldsymbol{d}^i)$
    /* solve linear system                                        */
    $\Delta\boldsymbol{d} = -\boldsymbol{K}^{-1} \cdot \boldsymbol{R}$
    if $O(\Delta\boldsymbol{d}, \boldsymbol{d})$ then
        /* quit iterative procedure                               */
        $\boldsymbol{d} \leftarrow \boldsymbol{d}^i$
    else
        /* update and continue iterative solution procedure:      */
        $\boldsymbol{d}^{i+1} \leftarrow \boldsymbol{d}^i + \Delta\boldsymbol{d}$
    end
end
return $\boldsymbol{d}$
```

Algorithm 5: Newton-Raphson iterative algorithm in order to solve equation (6.1).

7 The Material Point Method

7.1 Concept and notation of the material point method

In the Material Point Method (MPM), any material body, i.e., fluid or solid, is represented by a set of particles. These particles are called material points and are, throughout this thesis, denoted by the identifier `MP`. They carry all the properties of the material they represent permanently. During the discretization, properties such as the mass, and occupied volume of the material are assigned to material points. Consequently, the mass is conserved perfectly in the standard balance of momentum, respectively. The geometry of the body, e.g, defined by its surface, can only be approximated by the chosen cloud of material points. However, as each material point represents a fraction of the material, it also has properties such as position and velocity. Further, depending on the physical fields subject to the analysis, they might carry temperature, deformation state, and even history variables. It is generally a feature that these properties are kept locally at the material points. In the case of internal history variables, as, e.g., used in plasticity, this enables a consistent constitutive relation. Also, numerical stability is achieved. In these properties, a material point acts similar to an integration point as used in the finite element method. To denote any property being owned by a material point, the index $(\bullet)_{\mathtt{MP}}$ is used in this thesis. Usually, an MPM simulation uses a finite set of material points, where its whole number is denoted with `NoMP`. In the MPM, the material points do not interact directly, but rather via a computational background grid (CBG).

The CBG represents a computational mesh, similar to the finite element method. It is an important feature that this grid, however, does not carry any permanent information. The design and placement of a CBG in the MPM are generally arbitrary. Nevertheless, there are various aspects of its appearance to consider, which are reflected in the simulation. In this thesis, a CBG is defined by grid cells with indexes `NoC`, that are defined by a finite set of grid nodes. Nodes are indexed globally, using indexes `I` or `J`. The maximum index for nodes in a CBG is bounded by `NoI` and `NoJ`, and the number of cells is given by `NoC`. In reference to the notation for material points, a property discretely attached to a grid node is referred to by $(\bullet)^{\mathtt{I}}$ or $(\bullet)^{\mathtt{J}}$.

In the MPM, the differential equations are solved on the CBG. Usually, weak forms are used for that purpose. Again, very much alike a finite element mesh, the CBG discretizes the weak form with respect to nodal unknowns, which results in a matrix-vector representation of the equation systems. In order to assemble these systems, a link between the material point which carries the material state and the underlying grid is used. Various versions of the resulting mapping algorithm are known. However, this assembly process for the CBG from the material points is similar to a standard procedure where finite element systems are assembled from integration point contributions. This aspect is covered later on in more detail. The CBG also allows for boundary conditions to be set. Although the grid does not represent the geometry of bodies in the simulation, it may include additional forces or prescribed unknowns at the nodes. The solution is then computed using implicit or explicit methods. The solution for the weak form is obtained, on the CBG first.

At the end of a time-step in an MPM simulation, the solution is projected from grid nodes to material points. Usually, the inverse procedure is used as before in the assembly step. However this is not necessarily the case. With the grid solution available at the

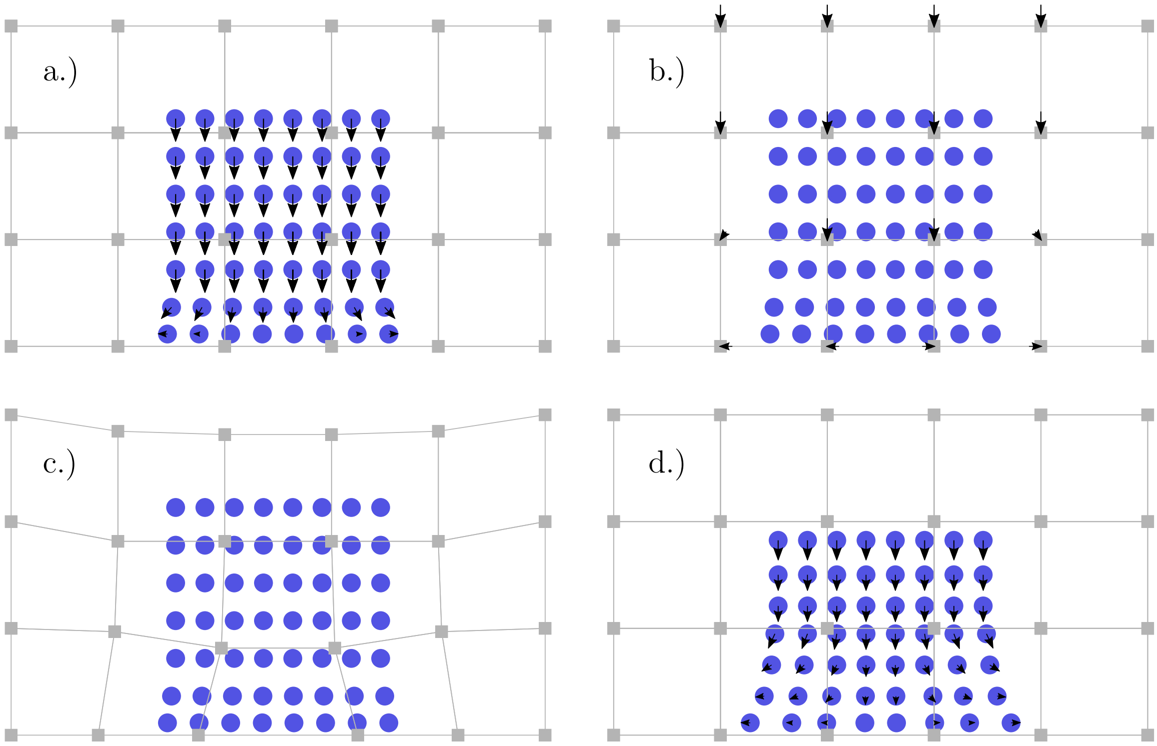


Figure 7.1: Illustration of the MPM standard algorithm. At the beginning of a time-step (a.), a computational grid is set and the material points carry the solution of the last time-step, e.g., in terms of material point velocity. In the next step, relevant properties are projected onto the nodes of the computational grid (b.). With the weak from assembled on the computational grid, a solution for the current time-step is computed on the grid (c.). In the last step, the grid nodal solution is mapped onto the material points, which are updated accordingly, e.g., in terms of their position (d.). Also, the grid might be reset in advance of the next time-step.

level of each material point, it can be used to update the material points properties. For example, in the discretization of the balance of momentum used in this thesis, the solution is velocities which are then used to update the material points position using time integration.

It was mentioned before that the CBG does not carry any permanent data. As such, at the end of each time-step, the grid is completely discarded. This is done after the solution is already mapped to the material points and thus not needed anymore. It is common practice in MPM to use the same design of the CBG throughout the time-steps. The main motivation is a boost in performance, and by not accumulating mesh deformation as in the finite element method, the MPM is not prone to mesh distortion errors.

A visual representation of the described steps is given in figure 7.1. The state denoted by a.) describes the initial state at the beginning of a time-step. Towards state .b), the material point's properties are mapped onto the nodes, i.e., assembly of the vector-matrix representation is done. Snapshot c.) illustrates the solution on the grid, visualized by a temporary mesh deformation. Notice that the bottom of the grid is subject to a boundary condition, as it does not deform in the vertical direction. The deformation is discarded at the end of the time-step d.). Before the solution is wiped, it was used to update the material points, illustrated by a change in position.

Using the described scheme, the MPM combines the Eulerian and Lagrangian mechanisms. The material points represent physical material in a Lagrangian sense, as their position directly follows its motion. With no direct interaction between them, very large deformations can easily be described. Even material discontinuities such as cracks may be modeled. Further, as material internal variables are attached to the material points, another Lagrangian concept is included for the aforementioned benefits. Still, in solving the balance equation on the CBG without having the grid following the motion of the material permanently, this part represents Eulerian aspects. The major benefit of typical Eulerian reference frames is the stability of the solution as they are not subject to mesh distortion. With the need for mapping operations between the material points and the CBG, additional overhead is produced compared to pure mesh-based or particle-based methods.

7.2 Variational formulation of the balance of momentum in MPM

In this thesis, the MPM is used to solve the balance of momentum in a so-called updated Lagrangian frame, which means it is represented with respect to the current configuration of body $\mathcal{B}$. To recall, the global statement of this was defined already in equation (3.23), reading

$$\operatorname{div}\boldsymbol{\sigma} + \rho\,(\mathfrak{b} - \ddot{\boldsymbol{x}}) = \boldsymbol{0}, \quad \forall \boldsymbol{x} \in \mathcal{B}. \tag{7.1}$$

By integrating equation (3.23) over a body it might be used to describe an equilibrium state. Further, classical boundary conditions might be defined

$$\boldsymbol{d} = \bar{\boldsymbol{d}} \quad \text{on} \quad \partial\mathcal{B}_d \qquad \text{and} \qquad \boldsymbol{\sigma}\cdot\boldsymbol{n} = \bar{\boldsymbol{t}} \quad \text{on} \quad \partial\mathcal{B}_{\bar{t}}, \tag{7.2}$$

although not as straightforward to use in the MPM as the surface geometry of the body is not well defined. Hereby, the whole boundary of body $\mathcal{B}$ is decomposed into a prescribed motion and traction part, i.e., $\mathcal{B} := \mathcal{B}_d \cup \mathcal{B}_{\bar{t}}$ with $\mathcal{B}_u \cap \mathcal{B}_{\bar{t}} = \emptyset$.

Following Galerkin's method, a weak form of equation (3.23) is achieved by introducing an arbitrary, vector-valued test function $\delta\boldsymbol{d}$, in the scope of a virtual displacement. It is subject to the prescribed boundary condition, and thus $\delta\boldsymbol{d} = \boldsymbol{0}$ on $\mathcal{B}_d$. At first we obtain

$$\int_{\mathcal{B}} \operatorname{div}\boldsymbol{\sigma}\cdot\delta\boldsymbol{d} + \rho\,(\mathfrak{b} - \ddot{\boldsymbol{x}})\cdot\delta\boldsymbol{d}\;\mathrm{dv} = 0. \tag{7.3}$$

While equation (7.3) represents a second-order differential with respect to the motion, it is possible to use the Gauss theorem in the relation

$$\begin{aligned} -\int_{\mathcal{B}} \operatorname{div}\,\boldsymbol{\sigma}\cdot\delta\boldsymbol{d}\;\mathrm{dv} &= \int_{\mathcal{B}} \boldsymbol{\sigma} : \operatorname{grad}\,\delta\boldsymbol{d}\;\mathrm{dv} - \int_{\mathcal{B}} \operatorname{div}\,(\boldsymbol{\sigma}\cdot\delta\boldsymbol{d})\;\mathrm{dv} \\ &= \int_{\mathcal{B}} \boldsymbol{\sigma} : \operatorname{grad}\,\delta\boldsymbol{d}\;\mathrm{dv} - \int_{\partial\mathcal{B}} (\boldsymbol{\sigma}\cdot\boldsymbol{n})\cdot\delta\boldsymbol{d}\;\mathrm{da} \\ &= \int_{\mathcal{B}} \boldsymbol{\sigma} : \operatorname{grad}\,\delta\boldsymbol{d}\;\mathrm{dv} - \int_{\partial\mathcal{B}_{\bar{t}}} \bar{\boldsymbol{t}}\cdot\delta\boldsymbol{d}\;\mathrm{da}, \end{aligned} \tag{7.4}$$

to shift a part of the divergence onto the boundary of the body. With this, the weak form G of the balance of momentum in the current configuration reads

$$G = \underbrace{\int_{\mathcal{B}} \boldsymbol{\sigma} : \text{grad}\ \delta\boldsymbol{d}\ \text{dv}}_{G^{\text{int}}} \underbrace{- \int_{\mathcal{B}} \rho\, \mathfrak{b} \cdot \delta\boldsymbol{d}\ \text{dv}}_{G_b^{\text{ext}}} \underbrace{+ \int_{\mathcal{B}} \rho\, \ddot{\boldsymbol{x}} \cdot \delta\boldsymbol{d}\ \text{dv}}_{G^{\text{mass}}} \underbrace{- \int_{\partial\mathcal{B}_{\bar{t}}} \bar{\boldsymbol{t}} \cdot \delta\boldsymbol{d}\ \text{da}}_{G_t^{\text{ext}}} = 0. \tag{7.5}$$

The weak form G, as specified here, is solved on the CBG in an MPM simulation.

7.3 Weak form discretization on the CBG

In MPM, the weak form is discretized in terms of material points and in fashion of support points with help of CBG nodes. The boundary term in equation (7.5) is modified to avoid the non-trivial evaluation of the required boundary integral. Once a body is discretized in material points, no sharp definition for its geometry is given anymore. Consequently, a surface integral on its boundary is not straightforward. It can be concluded that accurate tracking of domain boundaries is a weak point of the MPM compared with mesh-based Lagrangian methods as the FEM.

As known from other mesh-based approaches, the grid nodes render support points for approximated fields. Following Galerkin's principle, we use weighted residuals for the approximation. Mainly, this strategy is applied for the test function and the acceleration field

$$\delta\boldsymbol{d}(\boldsymbol{x}) \approx \delta\boldsymbol{d}^h(\boldsymbol{x}) = \sum_{\mathtt{I}=1}^{\mathtt{NoI}} \mathbb{N}^{\mathtt{I}}(\boldsymbol{x})\, \delta\boldsymbol{d}^{\mathtt{I}} \quad \text{and} \quad \ddot{\boldsymbol{x}}(\boldsymbol{x}) \approx \boldsymbol{a}^h(\boldsymbol{x}) = \sum_{\mathtt{I}=1}^{\mathtt{NoI}} \mathbb{N}^{\mathtt{I}}(\boldsymbol{x})\, \boldsymbol{a}^{\mathtt{I}}. \tag{7.6}$$

Here, node-wise vectors $(\bullet)^{\mathtt{I}}$ contain arbitrary discrete tests or nodal accelerations. Notice that the geometry, i.e., the position field, is not required to be approximated in that sense as the volume integrals rely on the associated volume of the material points. For the approximation, weighting- or shape-functions $\mathbb{N}$ are used. There are various versions of shape functions, a more detailed view is given later.

In order to provide the spatial derivatives which are required for the test functions, the spatial derivatives of the shape functions $\mathbb{N}$ are introduced as the B-vector

$$\text{grad}\ \delta\boldsymbol{d} \approx \text{grad}\ \delta\boldsymbol{d}^h(\boldsymbol{x}) = \sum_{\mathtt{I}=1}^{\mathtt{NoI}} \text{grad}\ \mathbb{N}^{\mathtt{I}}(\boldsymbol{x})\, \delta\boldsymbol{d}^{\mathtt{I}} = \sum_{\mathtt{I}=1}^{\mathtt{NoI}} \mathbb{B}^{\mathtt{I}}(\boldsymbol{x}) \otimes \delta\boldsymbol{d}^{\mathtt{I}}. \tag{7.7}$$

The system of equations must be formed to obtain a solution for the nodal unknowns of equation (7.5). In order to do this, we start by shifting the test functions onto the left side of the equation

$$G = \int_{\mathcal{B}} \text{grad}^T\ \delta\boldsymbol{d} : \boldsymbol{\sigma}\ \text{dv} - \int_{\mathcal{B}} \rho\, \delta\boldsymbol{d}^T \cdot \mathfrak{b}\ \text{dv} + \int_{\mathcal{B}} \rho\, \delta\boldsymbol{d}^T \cdot \ddot{\boldsymbol{x}}\ \text{dv} - \int_{\partial\mathcal{B}_{\bar{t}}} \delta\boldsymbol{d}^T \cdot \bar{\boldsymbol{t}}\ \text{da} = 0. \tag{7.8}$$

Further, approximations (7.6) and (7.7) for the test functions are inserted

$$\begin{aligned} G = &\sum_{\mathtt{I}=1}^{\mathtt{NoI}} \int_{\mathcal{B}} \left(\mathbb{B}^{\mathtt{I}}(\boldsymbol{x}) \otimes \delta\boldsymbol{d}^{\mathtt{I}}\right)^T : \boldsymbol{\sigma}\ \text{dv} &&- \sum_{\mathtt{I}=1}^{\mathtt{NoI}} \int_{\mathcal{B}} \rho\, \left(\mathbb{N}^{\mathtt{I}}(\boldsymbol{x})\, \delta\boldsymbol{d}^{\mathtt{I}}\right)^T \cdot \mathfrak{b}\ \text{dv} \\ &+ \sum_{\mathtt{I}=1}^{\mathtt{NoI}} \int_{\mathcal{B}} \rho\, \left(\mathbb{N}^{\mathtt{I}}(\boldsymbol{x})\, \delta\boldsymbol{d}^{\mathtt{I}}\right)^T \cdot \ddot{\boldsymbol{x}}\ \text{dv} &&- \sum_{\mathtt{I}=1}^{\mathtt{NoI}} \int_{\partial\mathcal{B}_{\bar{t}}} \left(\mathbb{N}^{\mathtt{I}}(\boldsymbol{x})\, \delta\boldsymbol{d}^{\mathtt{I}}\right)^T \cdot \bar{\boldsymbol{t}}\ \text{da} \quad = 0, \end{aligned} \tag{7.9}$$

which, after some algebraic manipulations results in an isolation of the nodal test vectors for the individual terms

$$\begin{aligned} G = \;& \sum_{\mathtt{I}=1}^{\mathtt{NoI}} \delta \boldsymbol{d}^{\mathtt{I}^T} \cdot \int_{\mathcal{B}} \mathbb{B}^{\mathtt{I}^T}(\boldsymbol{x}) \cdot \boldsymbol{\sigma} \text{ dv} && - \sum_{\mathtt{I}=1}^{\mathtt{NoI}} \delta \boldsymbol{d}^{\mathtt{I}^T} \cdot \int_{\mathcal{B}} \rho \, \mathbb{N}^{\mathtt{I}}(\boldsymbol{x}) \, \mathfrak{b} \text{ dv} \\ & + \sum_{\mathtt{I}=1}^{\mathtt{NoI}} \delta \boldsymbol{d}^{\mathtt{I}^T} \cdot \int_{\mathcal{B}} \rho \, \mathbb{N}^{\mathtt{I}}(\boldsymbol{x}) \, \ddot{\boldsymbol{x}} \text{ dv} && - \sum_{\mathtt{I}=1}^{\mathtt{NoI}} \delta \boldsymbol{d}^{\mathtt{I}^T} \cdot \int_{\partial \mathcal{B}_{\bar{t}}} \mathbb{N}^{\mathtt{I}}(\boldsymbol{x}) \, \bar{\boldsymbol{t}} \text{ da} \quad = 0 \end{aligned} \tag{7.10}$$

and for the whole equation,

$$\begin{aligned} G = \sum_{\mathtt{I}=1}^{\mathtt{NoI}} \delta \boldsymbol{d}^{\mathtt{I}^T} \cdot \Bigg(& \int_{\mathcal{B}} \mathbb{B}^{\mathtt{I}^T}(\boldsymbol{x}) \cdot \boldsymbol{\sigma} \text{ dv} \quad - \int_{\mathcal{B}} \rho \, \mathbb{N}^{\mathtt{I}}(\boldsymbol{x}) \, \mathfrak{b} \text{ dv} \\ & + \int_{\mathcal{B}} \rho \, \mathbb{N}^{\mathtt{I}}(\boldsymbol{x}) \, \ddot{\boldsymbol{x}} \text{ dv} \quad - \int_{\partial \mathcal{B}_{\bar{t}}} \mathbb{N}^{\mathtt{I}}(\boldsymbol{x}) \, \bar{\boldsymbol{t}} \text{ da} \Bigg) = 0. \end{aligned} \tag{7.11}$$

Up to this point, equation (7.11) can be found for lots of discretization methods and solution schemes. At this point we may isolate expressions for nodal forces in N

$$\boldsymbol{f}^{\mathtt{I}}_{\text{int}} = \int_{\mathcal{B}} \mathbb{B}^{\mathtt{I}^T}(\boldsymbol{x}) \cdot \boldsymbol{\sigma} \text{ dv}, \qquad \boldsymbol{f}^{\mathtt{I}}_{\text{ext},b} = \int_{\mathcal{B}} \rho \, \mathbb{N}^{\mathtt{I}}(\boldsymbol{x}) \, \mathfrak{b} \text{ dv}, \qquad \boldsymbol{f}^{\mathtt{I}}_{\text{ext},t} = \int_{\partial \mathcal{B}_{\bar{t}}} \mathbb{N}^{\mathtt{I}}(\boldsymbol{x}) \, \bar{\boldsymbol{t}} \text{ da}, \tag{7.12}$$

which appear in this context. The term including the accelerations is not treated this way, although it could be seen as nodal forces from inertia. Rather, we use the approximation from equation (7.6)

$$\begin{aligned} \int_{\mathcal{B}} \rho \, \mathbb{N}^{\mathtt{I}}(\boldsymbol{x}) \, \ddot{\boldsymbol{x}} \text{ dv} &= \sum_{\mathtt{J}=1}^{\mathtt{NoJ}} \int_{\mathcal{B}} \rho \, \mathbb{N}^{\mathtt{I}}(\boldsymbol{x}) \, \mathbb{N}^{\mathtt{J}}(\boldsymbol{x}) \, \boldsymbol{a}^{\mathtt{J}} \text{ dv} \\ &= \sum_{\mathtt{J}=1}^{\mathtt{NoJ}} \int_{\mathcal{B}} \rho \, \mathbb{N}^{\mathtt{I}}(\boldsymbol{x}) \, \mathbb{N}^{\mathtt{J}}(\boldsymbol{x}) \, \boldsymbol{I} \text{ dv} \cdot \boldsymbol{a}^{\mathtt{J}} \\ &= \sum_{\mathtt{J}=1}^{\mathtt{NoJ}} \boldsymbol{m}^{\mathtt{IJ}} \cdot \boldsymbol{a}^{\mathtt{J}}, \end{aligned} \tag{7.13}$$

with the consistent nodal mass matrix $\boldsymbol{m}^{\mathtt{IJ}} = \int_{\mathcal{B}} \rho \, \mathbb{N}^{\mathtt{I}}(\boldsymbol{x}) \, \mathbb{N}^{\mathtt{J}}(\boldsymbol{x}) \, \boldsymbol{I} \text{ dv}$.

Using the abbreviations from equations (7.12) and (7.13) the weak from on the CBG can be written as

$$G = \sum_{\mathtt{I}=1}^{\mathtt{NoI}} \delta \boldsymbol{d}^{\mathtt{I}^T} \cdot \left(\boldsymbol{f}^{\mathtt{I}}_{\text{int}} - \boldsymbol{f}^{\mathtt{I}}_{\text{ext},b} - \boldsymbol{f}^{\mathtt{I}}_{\text{ext},t} + \sum_{\mathtt{J}=1}^{\mathtt{NoJ}} \boldsymbol{m}^{\mathtt{IJ}} \cdot \boldsymbol{a}^{\mathtt{J}} \right) = 0. \tag{7.14}$$

In order to obtain the generally nonlinear, coupled system of equations, global vectors for nodal tests and unknowns are defined, i.e.,

$$\delta \boldsymbol{D} = \left[\delta \boldsymbol{d}^{1^T}, \, ..., \, \delta \boldsymbol{d}^{\mathtt{NoI}^T} \right]^T, \quad \boldsymbol{D} = \left[\boldsymbol{a}^{1^T}, \, ..., \, \boldsymbol{a}^{\mathtt{NoI}^T} \right]^T. \tag{7.15}$$

Now the Galerkin weak form can be differentiated with respect to the global vector of nodal tests $\delta\boldsymbol{D}$. As it depends linearly on the nodal test,

$$G = \delta\boldsymbol{D} \cdot \boldsymbol{R} = \delta\boldsymbol{D} \cdot \boldsymbol{0} \quad \text{with} \quad \boldsymbol{R}(\boldsymbol{D}) := \frac{\partial\, G(\delta\boldsymbol{D}, \boldsymbol{D})}{\partial\, \delta\boldsymbol{D}} \tag{7.16}$$

the resulting vector-valued system can now be solved with respect to the unknowns $\boldsymbol{D}$. In equation (7.16) the dependency of the nonlinear operator $\boldsymbol{R}(\boldsymbol{D})$ is stated explicitly.

It is now straightforward to extract the actual system of equations from equation (7.16) that yields

$$\boldsymbol{R}(\boldsymbol{D}) = \boldsymbol{0} \tag{7.17}$$

which representation is analog to equation (6.1). Various strategies can be employed in order to compute a solution with respect to the vector of unknowns. With the detailed discussion from chapter 6, we further use the strictly explicit form given in equation (6.7), with some minor modifications

$$\boldsymbol{M}_n \cdot \boldsymbol{D} = \boldsymbol{F}_n. \tag{7.18}$$

The main modification here is that all terms depending on known values only are summarized in the global force vector $\boldsymbol{F}$. That does include not only inner stress contributions but also potential damping effects and external loads. It can be observed that equation (7.18) actually represents Newton's second law for explicit time integration.

As this system is linear with respect to the current nodal accelerations gathered in the global vector $\boldsymbol{D}$, it can be solved directly. The global mass matrix $\boldsymbol{M}_n$ and global force vector $\boldsymbol{F}$ are formed using suitable assembly operators $\mathbf{A}$ on the nodal variants of mass matrix and force vectors

$$\boldsymbol{F} = \underset{\mathtt{I}=1}{\overset{\mathtt{NoI}}{\mathbf{A}}} \left(-\boldsymbol{f}^{\mathtt{I}}_{\text{int}} + \boldsymbol{f}^{\mathtt{I}}_{\text{ext},b} + \boldsymbol{f}^{\mathtt{I}}_{\text{ext},t}\right) \quad \text{and} \quad \boldsymbol{M} = \underset{\mathtt{I}=1,\mathtt{J}=1}{\overset{\mathtt{NoI},\mathtt{NoJ}}{\mathbf{A}}} \boldsymbol{m}^{\mathtt{IJ}}. \tag{7.19}$$

7.4 Aspects on the explicit solution on the CBG

In order to provide a computationally effective solution scheme, the proposed frameworks in this thesis utilize mass lumping. The beneficial effects were discussed before in chapter 6. The key aspect is a modification to equation (7.13)

$$\sum_{\mathtt{J}=1}^{\mathtt{NoJ}} \boldsymbol{m}^{\mathtt{IJ}} \cdot \boldsymbol{a}^{\mathtt{J}} \approx \sum_{\mathtt{J}=1}^{\mathtt{NoJ}} \boldsymbol{m}^{\mathtt{I}} \cdot \boldsymbol{a}^{\mathtt{J}} \tag{7.20}$$

with the lumped nodal mass matrix $\boldsymbol{m}^{\mathtt{I}} = \displaystyle\int_{\mathcal{B}} \rho\, \mathbb{N}^{\mathtt{I}}(\boldsymbol{x})\, \boldsymbol{I}\ \mathrm{dv}$.

With this modification, the coupled system of equations (7.18) relaxes to an uncoupled one. With inverion of a diagonal matrix being trivial, it is possible to compute each solution individually via

$$\boldsymbol{a}^{\mathtt{I}} = \boldsymbol{m}^{\mathtt{I}^{-1}} \cdot \boldsymbol{f}^{\mathtt{I}}_n \quad \text{with} \quad \boldsymbol{f}^{\mathtt{I}}_n = \left(-\boldsymbol{f}^{\mathtt{I}}_{\text{int}} + \boldsymbol{f}^{\mathtt{I}}_{\text{ext},b} + \boldsymbol{f}^{\mathtt{I}}_{\text{ext},t}\right)_n \tag{7.21}$$

It was discussed before that an explicit solution scheme demands small time-steps to stay stable. Therefore, time-step size estimators are proposed, to predict the time-step size for

a stable scheme based on multiple criteria. While there exist sophisticated, discretization-dependent approaches to determining the critical time-step size Δt, see, e.g., BERZINS [13], there are also some general concepts applicable. One of them is condensed in the so-called Courant-Friedrichs-Lewy constant.
The idea is to provide a sufficiently dense resolution of a process in time, such that no information is lost. For example, in solid mechanics the critical time-step scales with the speed of sound of a material and a characteristic length of the spatial discretization. In FEM, for example, this length might be the smallest dimension of an element.
In MPM analysis, these criterion is modified of course. For instance, in ZHANG ET AL. [163] the authors suggest an MPM critical estimator with

$$\Delta t_{cr} = \frac{l_{cr}}{\max\limits_{\texttt{NoMP}} \left(c_{\texttt{MP}} + \sqrt{\boldsymbol{v}_{\texttt{MP}} \cdot \boldsymbol{v}_{\texttt{MP}}} \right)}. \tag{7.22}$$

Here, l_{cr} is the shortest dimension of all cells in the CBG. Furthermore, the material elastic wave speed is used with,

$$c_{\texttt{MP}} = \sqrt{\frac{E_{\texttt{MP}}\,(1 - \nu_{\texttt{MP}})}{(1 + \nu_{\texttt{MP}})(1 - 2\,\nu_{\texttt{MP}})\,\rho_{\texttt{MP}}}} \tag{7.23}$$

where $E_{\texttt{MP}}$, $\nu_{\texttt{MP}}$ are the elastic modulus and Poission's ratio of the material at any material point MP, and $\rho_{\texttt{MP}}$ denotes the density.

The term artificial damping describes the introduction of viscous effects to a time-dependent solution that are not physically motivated but rather used to dissipate kinematic energy from a system. It is usually applied to stabilize a dynamic solution or to enforce a steady state solution. One way of numerical damping is described in LI ET AL. [83], resulting in an artificial damping force

$$\boldsymbol{f}^{\texttt{I}}_{\text{ext},d} = -\alpha_d\, \boldsymbol{v}^{\texttt{I}}\, m^{\texttt{I}}. \tag{7.24}$$

Here the artificial damping parameter α_d in s^{-1} is introduced. The new nodal force vector is assembled in the same fashion as described in equation (7.19); the required nodal velocity vector $\boldsymbol{v}^{\texttt{I}}$ must be assembled as well. It is worth mentioning, that this approach is not very selective in its damping behavior, nor is it physically motivated. It must always be considered always with caution and with adoption to the specific boundary value problem to get reasonable results.

Motivated by investigating different update strategies in MPM, NAIRN [103] proposed a numerical damping scheme that uses more advanced features in MPM algorithms. By introducing a scaling between two exchange strategies of the CBG and material points, which either use just the grid acceleration to update the material points velocity or the grid velocity is directly utilized. The author refers to the latter as a special from of numerical damping as it was observed that it does smooth out acceleration peaks for the cost of global momentum.

7.5 Leapfrog time integration on the CBG

The balance of momentum is usually solved to obtain the position of particles or control points, i.e. the aim is to analyze motion. However, the solution of equation (7.18) results in nodal accelerations for the current time-step only. In order to provide all necessary information for the next time-step, velocities and displacement must be computed as well. In chapter 5, time integration schemes were discussed, which provide approximations for the differentials of displacements. The leapfrog algorithm introduces sub-time-steps to improve accuracy and stability while still providing excellent performance. The main concept is set by a central differencing scheme in time. This time integration method is chosen now for obtaining velocities and displacements on the CBG.

For each time-step, for each node of the grid, the following solution and update formulas would result from the scheme as introduced in (5.3)

$$\begin{aligned} \boldsymbol{a}_n^{\mathtt{I}} &= \boldsymbol{m}^{\mathtt{I}^{-1}} \cdot \boldsymbol{f}_n^{\mathtt{I}} \\ \boldsymbol{v}_{n+1/2}^{\mathtt{I}} &= \boldsymbol{v}_{n-1/2}^{\mathtt{I}} + \boldsymbol{a}_n^{\mathtt{I}}\, \Delta t \\ \boldsymbol{x}^{\mathtt{I}} &= \boldsymbol{x}_n^{\mathtt{I}} + \boldsymbol{v}_{n+1/2}^{\mathtt{I}}\, \Delta t \end{aligned} \tag{7.25}$$

In MPM, the CBG does not carry any information throughout a single time-step. Hence, previous solutions, such as $\boldsymbol{v}_{n-1/2}^{\mathtt{I}}$ are not available and need to be computed from the material point discretization

$$\boldsymbol{v}_{n-1/2}^{\mathtt{I}} = \boldsymbol{m}^{\mathtt{I}^{-1}} \cdot \boldsymbol{p}_{n-1/2}^{\mathtt{I}} \tag{7.26}$$

where $\boldsymbol{p} = \displaystyle\int_{\mathcal{B}} \rho\, \mathbb{N}^{\mathtt{I}}(\boldsymbol{x})\, \boldsymbol{v}\; \mathrm{dv}$ represents the nodal vector of linear momentum. Notice that this additional computation does not produce significant computational overhead. As given in equation (7.25) the nodal position is not updated actually, so that a grid deformation in terms of displacements for a time-step is computed here only for post-processing purposes. Thus, the actual leapfrog time integration on the CBG for the material point method renders

$$\begin{aligned} \boldsymbol{a}_n^{\mathtt{I}} &= \boldsymbol{m}^{\mathtt{I}^{-1}} \cdot \boldsymbol{f}_n^{\mathtt{I}} \\ \boldsymbol{v}_{n+1/2}^{\mathtt{I}} &= \boldsymbol{m}^{\mathtt{I}^{-1}} \cdot \boldsymbol{p}_{n-1/2}^{\mathtt{I}} + \boldsymbol{a}_n^{\mathtt{I}}\, \Delta t \\ \boldsymbol{u}^{\mathtt{I}} &= \boldsymbol{v}_{n+1/2}^{\mathtt{I}}\, \Delta t \end{aligned} \tag{7.27}$$

A collection of temporary properties at each node in the CBG with respect to the chosen leapfrog time integration is given in table 7.1.

acceleration	$\boldsymbol{a}^{\mathtt{I}}$	$\mathrm{m\,s^{-2}}$
velocity	$\boldsymbol{v}^{\mathtt{I}}$	$\mathrm{m\,s^{-1}}$
linear momentum	$\boldsymbol{p}^{\mathtt{I}}$	$\mathrm{kg\,m\,s^{-1}}$
stress resultants (force)	$\boldsymbol{f}^{\mathtt{I}}$	N
mass	$m^{\mathtt{I}}$	kg

Table 7.1: Persistent properties of a material point for solid mechanics.

7.6 Boundary conditions on the CBG

The CBG allows for applying boundary conditions. Besides the technical possibility to add external force contributions to each node, this technique is not very popular. The reason is that external loads are usually applied to either surface, e.g., as traction, or in terms of volumetric loads. As the nodes of the CBG do not represent any material interface, or even indicate the presence of any material, it is hard to find any use for this in solid mechanics. Nevertheless, it is very common to prescribe any set of unknowns on the CBG, which influences the solution similar to essential boundary conditions in FEM. Notice that in the case of the discussed leapfrog time integration, the solution of the two fields, acceleration, and velocity, must be prescribed

$$\boldsymbol{v}^{\mathtt{I}}_{n+1/2} = \bar{\boldsymbol{v}}^{\mathtt{I}}_{n+1/2}, \qquad \text{and} \qquad \boldsymbol{a}^{\mathtt{I}}_{n} = \bar{\boldsymbol{a}}^{\mathtt{I}} \tag{7.28}$$

Although this is not relevant on the CBG, affects the material points, as the nodal acceleration is also used. The prescribed movement of a node is mostly used to model rigid surfaces, either with perfect slip or stick conditions.

7.7 Interface between material points and the CBG

Up to this point, the computation of nodal vectors on the CBG relies on the analytical integration of continuous fields, e.g. stresses. It is fundamental to the MPM that these integrals are approximated numerically. Unlike most FEM implementations which use the efficient quadrature rules of Gauss-Legendre or Gauss-Lobatto, the MPM is based on the Riemann sum. In this single point quadrature rule, the integration over a volume is approximated by the sum of the integrant evaluated at discrete points in space, weighted by associated volumes. In MPM, these discrete points are the material points

$$\int_{\mathcal{B}} (\bullet)\ \mathrm{dv} \approx \sum_{\mathtt{MP}=1}^{\mathtt{NoMP}} (\bullet)_{\mathtt{MP}}\ \mathrm{v}_{\mathtt{MP}}. \tag{7.29}$$

In other words, a material point discretizes a continuous field as a constant inside its associated volume. With this approximation the nodal vectors and mass matrix, to compute the solution on the computational grid can be evaluated.

For the lumped nodal mass matrix, we obtain

$$\boldsymbol{m}^{\mathtt{I}} = \int_{\mathcal{B}} \rho\, \mathbb{N}^{\mathtt{I}}(\boldsymbol{x})\, \boldsymbol{I}\ \mathrm{dv} \approx \sum_{\mathtt{MP}=1}^{\mathtt{NoMP}} \rho_{\mathtt{MP}}\, \mathrm{v}_{\mathtt{MP}}\, \mathbb{N}^{\mathtt{I}}(\boldsymbol{x}_{\mathtt{MP}})\, \boldsymbol{I}, \tag{7.30}$$

where $\rho_{\mathtt{MP}}$ represents the current density at the material point and $\boldsymbol{x}_{\mathtt{MP}}$ its current spatial position. Notice that this implies a certain set of properties, a material point needs to carry. Later in this chapter, we give a summary on the necessary properties. Also, the indicated updates as indicated by the term "current" are to be discussed.

The individual contributions to the nodal force vector are given by

$$\begin{aligned}
\boldsymbol{f}^{\mathtt{I}}_{\text{int}} &= \int_{\mathcal{B}} \mathbb{B}^{\mathtt{I}^T}(\boldsymbol{x}) \cdot \boldsymbol{\sigma} \text{ dv} &&\approx \sum_{\mathtt{MP}=1}^{\mathtt{NoMP}} \mathbb{B}^{\mathtt{I}^T}(\boldsymbol{x}_{\mathtt{MP}}) \cdot \boldsymbol{\sigma}_{\mathtt{MP}} \, \text{v}_{\mathtt{MP}}, \\
\boldsymbol{f}^{\mathtt{I}}_{\text{ext},b} &= \int_{\mathcal{B}} \rho \, \mathbb{N}^{\mathtt{I}}(\boldsymbol{x}) \, \boldsymbol{b} \text{ dv} &&\approx \sum_{\mathtt{MP}=1}^{\mathtt{NoMP}} \rho_{\mathtt{MP}} \, \mathbb{N}^{\mathtt{I}}(\boldsymbol{x}_{\mathtt{MP}}) \, \boldsymbol{b}_{\mathtt{MP}} \, \text{v}_{\mathtt{MP}}, \\
\boldsymbol{f}^{\mathtt{I}}_{\text{ext},t} &= \int_{\partial\mathcal{B}_{\bar{t}}} \mathbb{N}^{\mathtt{I}}(\boldsymbol{x}) \, \bar{\boldsymbol{t}} \text{ da}, &&\approx \sum_{\mathtt{MP}=1}^{\mathtt{NoMP}} \mathbb{N}^{\mathtt{I}}(\boldsymbol{x}_{\mathtt{MP}}) \, \bar{\boldsymbol{t}}_{\mathtt{MP}} \, \bar{h}_{\mathtt{MP}} \, \text{v}_{\mathtt{MP}}.
\end{aligned} \tag{7.31}$$

Notice that the current Cauchy stresses $\boldsymbol{\sigma}_{\mathtt{MP}}$ and external volumetric acceleration $\boldsymbol{b}_{\mathtt{MP}}$ at the material point $\mathtt{MP}$ are used in equation (7.31). Further, the nodal force contribution from external traction $\bar{\boldsymbol{t}}$ is just approximated, caused by the lack of a proper determination of the loaded surface $\partial\mathcal{B}_{\bar{t}}$ in an MPM discretization. Rather, the approximation is based on a volume integral over the whole body $\mathcal{B}$, of an equivalent volume acceleration $\bar{\boldsymbol{t}}_{\mathtt{MP}}$ at the material point, modified by a correction factor $\bar{h}_{\mathtt{MP}}$ in m^{-1}. One can think of $\bar{h}_{\mathtt{MP}}$ as the thickness of the layer between the loaded virtual surface of a body and the material points closest to this surface. The application of traction boundary conditions in MPM is not straightforward. At last, the linear momentum vector is computed as

$$\boldsymbol{p} = \int_{\mathcal{B}} \rho \, \mathbb{N}^{\mathtt{I}}(\boldsymbol{x}) \, \boldsymbol{v} \text{ dv} \approx \sum_{\mathtt{MP}=1}^{\mathtt{NoMP}} \rho_{\mathtt{MP}} \, \mathbb{N}^{\mathtt{I}}(\boldsymbol{x}_{\mathtt{MP}}) \, \boldsymbol{v}_{\mathtt{MP}} \, \text{v}_{\mathtt{MP}}, \tag{7.32}$$

with the current velocity vector of each material point $\boldsymbol{v}_{\mathtt{MP}}$.

It can be observed that to project properties from the set of material points onto the computational grid an integral equation needs to be carried out. This is reasonable, as any property is based on the presence of material inside the CBG, is defined for. Further, not all material points represent the same fraction of material, e.g., via their associated volume. If a property is assembled on the grid, i.e., computed at the grid nodes from a set of material points, it can be done by considering the mass as a weighting factor for the presence of material. Logically, this leads to the more expensive computation of, e.g., grid velocities using the linear momentum as shown in equation (7.32). In a more general scheme, this mapping is done via

$$(\bullet)^{\mathtt{I}} = \left(\sum_{\mathtt{MP}=1}^{\mathtt{NoMP}} \rho_{\mathtt{MP}} \, \mathbb{N}^{\mathtt{I}}(\boldsymbol{x}) \, \text{v}_{\mathtt{MP}} \right)^{-1} \left(\sum_{\mathtt{MP}=1}^{\mathtt{NoMP}} \rho_{\mathtt{MP}} \, (\bullet)_{\mathtt{MP}} \, \mathbb{N}^{\mathtt{I}}(\boldsymbol{x}) \, \text{v}_{\mathtt{MP}} \right). \tag{7.33}$$

The inverse projection represents an equally common task in the MPM. This field simply needs to be evaluated at the positions of the material points in order to obtain the requested value, assuming the quantities on the computational grid act as support points to an interpolated field, i.e.,

$$(\bullet)_{\mathtt{MP}} = \sum_{\mathtt{I}=1}^{\mathtt{NoI}} \mathbb{N}^{\mathtt{I}}(\boldsymbol{x}_{\mathtt{MP}}) \, (\bullet)^{\boldsymbol{I}}. \tag{7.34}$$

With the set of shape functions $\mathbb{N}$ defined on the CBG, it is straightforward to provide spatial gradients of the fields interpolated by them

$$\text{grad}\,(\bullet)_{\mathtt{MP}} = \sum_{\mathtt{I}=1}^{\mathtt{NoI}} \text{grad}\, \mathbb{N}^{\mathtt{I}}(\boldsymbol{x}_{\mathtt{MP}}) \, (\bullet)^{\boldsymbol{I}} = \sum_{\mathtt{I}=1}^{\mathtt{NoI}} \mathbb{B}^{\mathtt{I}}(\boldsymbol{x}_{\mathtt{MP}}) \boxdot (\bullet)^{\boldsymbol{I}}. \tag{7.35}$$

In order to account for the variation of necessary operation with respect to the usage of the $\mathbb{B}$-vector, the placeholder $\boxdot$ is used in equation (7.35). In the case of $(\bullet)$ being a scalar, $\boxdot$ renders a vector-scalar product, or in the common case of $(\bullet)$ representing a vector quantity, $\boxdot \to \otimes$.

Notice that the order of spatial derivative is restricted by the ansatz-space of the shape functions used. With the weak form of the balance of momentum (7.5) containing only first-order derivatives, standard Lagrangian shape functions are technically sufficient.

Throughout this thesis, shape functions and $\mathbb{B}$-vectors are evaluated at material points position, with index MP. In order to provide leaner equations, the shorthand notation $\mathbb{N}^{\mathtt{I}}(\boldsymbol{x}_{\mathtt{MP}}) := \mathbb{N}^{\mathtt{I}}_{\mathtt{MP}}$ and $\mathbb{B}^{\mathtt{I}}(\boldsymbol{x}_{\mathtt{MP}}) := \mathbb{B}^{\mathtt{I}}_{\mathtt{MP}}$ is introduced at this point, respectively.

7.8 Material point level equations

In MPM, the material points represent the underlying material itself. Consequently, they carry properties of the material they represent. As the material deforms, these properties need to updated. The kinematic quantities, position and time, are updated from the back-projected solution at the nodes of the CBG. The performed update formulas using the leapfrog algorithm (5.3) and the previously discussed mapping technique (7.34), are given with

$$\begin{aligned} \boldsymbol{v}_{\mathtt{MP}_{n+1/2}} &= \boldsymbol{v}_{\mathtt{MP}_{n-1/2}} + \boldsymbol{a}_{\mathtt{MP}}\,\Delta t \quad \text{with} \quad & \boldsymbol{a}_{\mathtt{MP}} &= \sum_{\mathtt{I}=1}^{\mathtt{NoI}} \mathbb{N}^{\mathtt{I}}_{\mathtt{MP}}\,\boldsymbol{a}^{\boldsymbol{I}}_{n}, \\ \boldsymbol{x}_{\mathtt{MP}} &= \boldsymbol{x}_{\mathtt{MP}_n} + \boldsymbol{u}_{\mathtt{MP}} \quad \text{with} \quad & \boldsymbol{u}_{\mathtt{MP}} &= \Delta t \sum_{\mathtt{I}=1}^{\mathtt{NoI}} \mathbb{N}^{\mathtt{I}}_{\mathtt{MP}}\,\boldsymbol{v}^{\boldsymbol{I}}_{n+1/2}. \end{aligned} \tag{7.36}$$

It is crucial to recognize, that in order to update the material point's position, the grid velocity is used. The velocity of the material point itself is updated using the grid accelerations. This scheme represents the central differencing in time, down to the material point.

While the movement of the individual material points also implies deformation of the body, the material deformation state is also carried by each material point individually. The deformation property in that regard is the deformation gradient. Throughout the simulation, incremental deformation gradients for each time-step are accumulated locally at the material points. As the deformation gradient also includes rotation, a multiplicative update scheme is chosen

$$\boldsymbol{F}_{\mathtt{MP}} = \left(\frac{\partial\,\boldsymbol{x}}{\partial\,\boldsymbol{X}}\right)_{\mathtt{MP}} = \left(\frac{\partial\,\boldsymbol{x}}{\partial\,\boldsymbol{x}_n} \cdot \frac{\partial\,\boldsymbol{x}_n}{\partial\,\boldsymbol{X}}\right)_{\mathtt{MP}} = \Delta\boldsymbol{F}_{\mathtt{MP}} \cdot \boldsymbol{F}_{\mathtt{MP}_n}. \tag{7.37}$$

Here, the incremental deformation gradient describes the deformation from the previously equilibrated state to the current one. The update procedure is consistent as long as it is continuously performed. An illustration of the update procedure throughout temporal configuration, i.e., time-steps, is given in figure 7.2. In the context of the leapfrog algorithm from equation (7.36), resulting in nodal half step velocities as the solution available on the nodes, the deformation increment can be computed as

$$\begin{aligned}\Delta \boldsymbol{F}_{\mathtt{MP}} = \left(\frac{\partial\, \boldsymbol{x}}{\partial\, \boldsymbol{x}_n}\right)_{\mathtt{MP}} &= \left(\frac{\partial\, \boldsymbol{x}_n + \boldsymbol{u}}{\partial\, \boldsymbol{x}_n}\right)_{\mathtt{MP}} \\ &= \left(\frac{\partial\, \boldsymbol{x}_n}{\partial\, \boldsymbol{x}_n} + \frac{\partial\, \boldsymbol{u}}{\partial\, \boldsymbol{x}_n}\right)_{\mathtt{MP}} \\ &= \left(\boldsymbol{I} + \operatorname{grad}_n \boldsymbol{u}\right)_{\mathtt{MP}} .\end{aligned} \tag{7.38}$$

Further, the displacement gradient with respect to the previous configuration is also computed in accordance with equation (7.36),

$$\operatorname{grad}_n \boldsymbol{u}_{\mathtt{MP}} = \operatorname{grad}_n \left(\Delta t \sum_{\mathtt{I}=1}^{\mathtt{NoI}} \mathbb{N}^{\mathtt{I}}_{\mathtt{MP}}\, \boldsymbol{v}^{\boldsymbol{I}}_{n+1/2}\right) = \sum_{\mathtt{I}=1}^{\mathtt{NoI}} \mathbb{B}^{\mathtt{I}\,T}_{\mathtt{MP}} \otimes \boldsymbol{v}^{\boldsymbol{I}}_{n+1/2}\, \Delta t = \Delta t\, \boldsymbol{L}_{\mathtt{MP}_{n+1/2}}, \tag{7.39}$$

using the spatial velocity gradient $\boldsymbol{L}_{\mathtt{MP}_{n+1/2}}$.

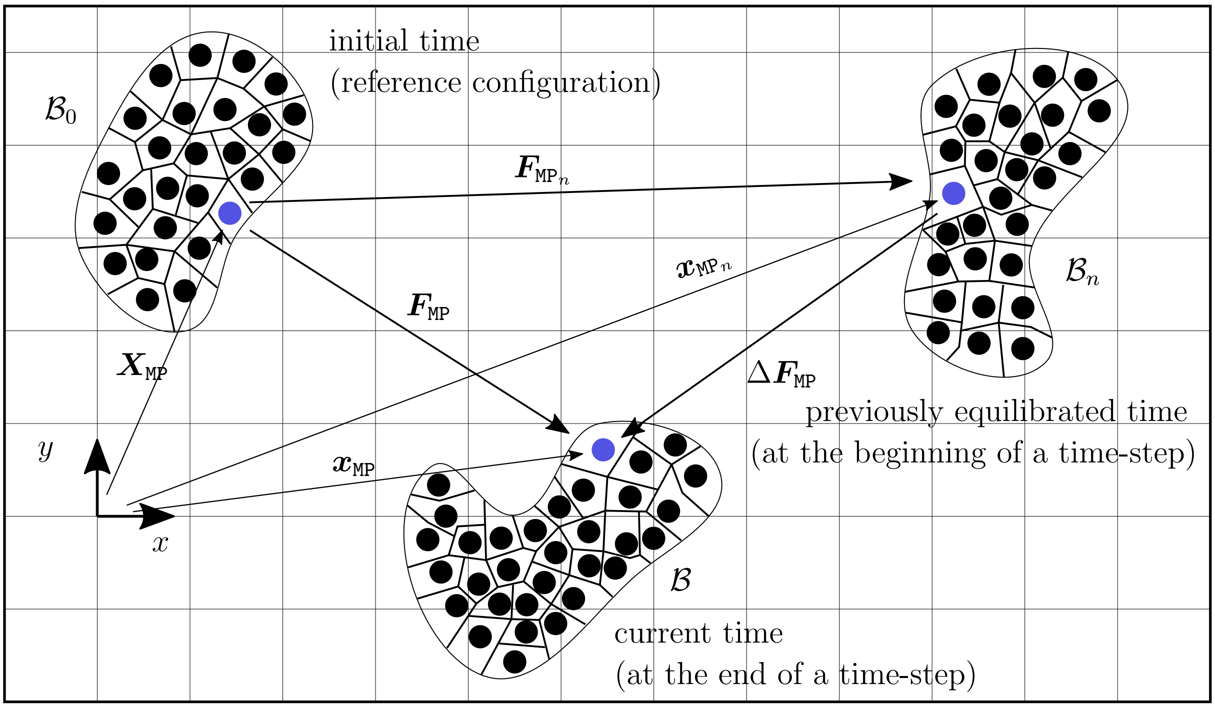

$\mathcal{B}_0$
initial time
(reference configuration)
$\boldsymbol{F}_{\mathtt{MP}_n}$
$\mathcal{B}_n$
$\boldsymbol{x}_{\mathtt{MP}_n}$
$\boldsymbol{F}_{\mathtt{MP}}$
$\boldsymbol{X}_{\mathtt{MP}}$
$\Delta\boldsymbol{F}_{\mathtt{MP}}$
previously equilibrated time
(at the beginning of a time-step)
y
$\boldsymbol{x}_{\mathtt{MP}}$
x
$\mathcal{B}$
current time
(at the end of a time-step)

Figure 7.2: Illustration of the deformation update scheme in time, throughout temporal configurations.

With the deformation gradient, representing the current state of material deformation available at the material point, it is also possible to update the associated volume according to the transport theorem (3.3) and the material density, due to the satisfaction of the balance of mass equation (3.20)

$$\mathrm{v}_{\mathtt{MP}} = J_{\mathtt{MP}}\, v_{\mathtt{MP}_0} \text{ and } \rho_{\mathtt{MP}} = J^{-1}_{\mathtt{MP}}\, \rho_{\mathtt{MP}_0}. \tag{7.40}$$

As the deformation at a material point is updated, a new stress state must also be considered. Generally, stresses can be obtained from a constitutive relation, such as described in chapters 4.2 or 4.3. In these chapters, stresses arise from an elastic or elasto-plastic response, described by a total deformation gradient, describing the deformation of the body since its stress-free reference configuration. As with the before-mentioned update scheme (7.37), this property is available at each material point. Thus, for the formulations in this thesis, stresses are computed straight by evaluating a constitutive equation in the form

$$\boldsymbol{\sigma}_{\mathtt{MP}} = \hat{\boldsymbol{\sigma}}(\boldsymbol{F}_{\mathtt{MP}}, \boldsymbol{F}_{\mathtt{MP}_n}, \boldsymbol{h}_{\mathtt{MP}}, \boldsymbol{h}_{\mathtt{MP}_n}) \tag{7.41}$$

with additional time-dependent input $\boldsymbol{h}_{\mathtt{MP}}$. The execution of a constitutive equation is explicitly reflected by the executed algorithms; see algorithm 4. Notice that in some cases, e.g., in plasticity a fraction of the time-dependent variables are updated itself.

It is worth noticing that especially in the solution scheme based on an updated configuration, as used in this thesis, gives an attractive use case for updating stresses in a hypo-elastic fashion. With this ansatz, stresses are actually updated using a stress increment

$$\boldsymbol{\sigma}_{\mathtt{MP}} = \boldsymbol{\sigma}_{\mathtt{MP}_n} + (\boldsymbol{\sigma}_{\mathtt{MP}})^{\cdot}\ \Delta t \tag{7.42}$$

computed from a stress rate $(\boldsymbol{\sigma}_{\mathtt{MP}})$. A feature of this approach is that no continuous update of the deformation state is required, as the stress rate is usually based on the incremental deformation $\Delta\boldsymbol{F}$. Nevertheless, the objectivity of the stress rate must be given, which is not straightforward and mostly requires additional overhead. For example, when using the famous Jaumann rate, which is an objective rate to the Cauchy stress tensor, the current spin of the material point must still be updated continuously in time.

position	$\boldsymbol{x}_{\mathtt{MP}}$	m
velocity	$\boldsymbol{v}_{\mathtt{MP}}$	$\mathrm{m\,s^{-1}}$
associated volume	$\mathrm{v}_{\mathtt{MP}}$	$\mathrm{m^{3}}$
material density	$\rho_{\mathtt{MP}}$	$\mathrm{kg\,m^{-3}}$
deformation gradient	$\boldsymbol{F}_{\mathtt{MP}}$	
Cauchy stresses	$\boldsymbol{\sigma}_{\mathtt{MP}}$	$\mathrm{N\,m^{-2}}$
vector of history variables	$\boldsymbol{h}_{\mathtt{MP}}$	1

Table 7.2: Persistent properties of a material point for solid mechanics.

7.9 An update stress last (USL) scheme in MPM

As discussed at the beginning of this thesis, the MPM knows several variants. With respect to the applied algorithm, namely, the flow for the computation of the solution of a time-step, the most popular variants are the Update Stress First (USF) and Update Stress Last (USL) algorithms. The difference between these variants is when the deformation and stresses are updated, either at the beginning of a time-step, or its end. A consequence is that the USF scheme requires an additional exchange between material points and the CBG, which is more expensive. However, the USF scheme is more energy conserving than the USL variant, where only one exchange between material points and CBG is done, to update deformation at the end of the time-step. In Bardenhagen [10], the author strongly investigated the differences of both schemes and concluded that the USL scheme, though more dissipative, might be the better choice regarding numerical stability. The Modified Update Stress Last scheme (MUSL) bridges the two original variants in updating the stresses at the end of a time-step but using a correction from an additional mapping step.

In order to approach the highly dynamic engineering problems considered in this thesis, the USL scheme was considered. Algorithm 6 represents the steps in an explicit leapfrog time integration, as in the MPM simulations in this thesis.

```
Input: MPM discretization, CBG, time discretization ($t_{\text{start}}$, $t_{\text{end}}$, $\Delta t$)
for $t \leftarrow t_{start}$ to $t_{end}$ do
    /* perform time-step */
    $(\bullet)_{\mathtt{MP}_n} \leftarrow (\bullet)_{\mathtt{MP}}$

    /* compute shape functions for each material point */
    $\mathbb{N}^{\mathtt{I}}_{\mathtt{MP}} \leftarrow \mathbb{N}^{\mathtt{I}}(\boldsymbol{x}_{\mathtt{MP}_n})$, $\mathbb{B}^{\mathtt{I}}_{\mathtt{MP}} \leftarrow \mathbb{B}^{\mathtt{I}}(\boldsymbol{x}_{\mathtt{MP}_n})$

    /* compute nodal vectors/ assemble global system using $\mathbb{N}^{\mathtt{I}}_{\mathtt{MP}}$, $\mathbb{B}^{\mathtt{I}}_{\mathtt{MP}}$ */
    $\boldsymbol{f}^{\mathtt{I}}_n$, $\boldsymbol{p}^{I}_n$, $m^{\mathtt{I}}_n$

    /* solve for nodal vectors/ solve global system */
    $\boldsymbol{a}^{\mathtt{I}}_n$, $\boldsymbol{v}^{\mathtt{I}}_{n+1/2}$

    /* map CBG-solution to the material points using $\mathbb{N}^{\mathtt{I}}_{\mathtt{MP}}$, $\mathbb{B}^{\mathtt{I}}_{\mathtt{MP}}$ */
    $\boldsymbol{a}_{\mathtt{MP}}$, $\boldsymbol{u}_{\mathtt{MP}}$, $\boldsymbol{L}_{\mathtt{MP}_{n+1/2}}$

    /* update material points kinematic and constitutive state */
    $\boldsymbol{x}_{\mathtt{MP}}$, $\boldsymbol{v}_{\mathtt{MP}}$, $\boldsymbol{F}_{\mathtt{MP}}$, $\rho_{\mathtt{MP}}$, $\mathrm{v}_{\mathtt{MP}}$, $\boldsymbol{\sigma}_{\mathtt{MP}}$, $\boldsymbol{h}_{\mathtt{MP}}$
end
```

Algorithm 6: Algorithmic flow of the USL algorithm, using the explicit leapfrog time integration.

8 The ELSE code

To date, the MPM is not part of standard commercial simulation software. Although the method can be implemented in FEM codes to some degree Chen and Brannon [26], dedicated codes provide more flexibility required in research and are usually more performant. Modern programs often make use of Object Oriented Programming (OOP) in languages such as C++. In Ma et al. [93], the authors proposed a class layout for MPM simulations. To date, their code is not available to the public. de Vaucorbeil et al. [35] present their open source MPM framework *Karamelo*, based on the widely used LAMMPS software architecture. LAMMPS is a massively parallel particle simulation code, but without the opportunity to incorporate computational meshes. Another promising project to name here is *Anura3D*, see e.g. Ceccato and Simonini [20], which is not publicly available either and focused on soil-water-structure interaction. Nevertheless, open-source projects for MPM simulations do exist. The *Unitah* project provides a set of software components and libraries for MPM simulations, along with documentation. Also, the research group around Prof. J.A. Nairn from the Oregon-State-University provides code used in their publications, e.g., *Nairn-MPM*. However, most research publications on MPM rely on in-House developed codes, compare e.g. Dong and Grabe [38] among others.

For the numerical analysis in this thesis, a computational framework for the material point method (MPM) was developed. In this chapter, a brief review of the code structure is given, as it reflects the necessary layers for the implementation of this method. Also the implementation details towards the implemented USL scheme are discussed. The presented solver, body, and grid class are used for all examples in this document (if not mentioned explicitly otherwise). The ELSE (Eulerian-Lagrangian-Simulation-Environment) code represents a computational framework for the material point method primarily. The code design is oriented towards:

- The **flexibility** to provide a common basis for the implementation of MPM variants and even beyond that.
- Providing a **user-friendly** experience. In order to be used in research, the framework provides simple problem definitions (input script) and versatility in post-processing.
- Using a class versatility that is intuitive to provide a good **programmer-friendly** experience. The clear goal is to prefer structures that enable new contributions to aspects of the present implementation using pre-defined interfaces.
- **Performance** and scalability as major aspects of any simulation tool.

The above stated design aspects are listed in the order of their priority for the ELSE code. The main intensive on the development of the framework was to provide software that does meet the requirements of research practice.

The first aspect in the context of research is flexibility. Trying new things, ranging from the implementation of a new solution strategy down to an effective way for doing simple numerical parameter studies, is set to be the main use case of ELSE.

In order to serve this purpose, a Python (van Rossum [150]) application programming interface (API) is provided. While for performance reasons, the compiled C++ programming language is actually used for the program, the python bindings are added using the pybind11 library (Jakob et al. [68]). This ansatz is inspired by the rise of python API also for numerical tools, such as the *FEniCS* project or the *NGSolve* project, see Logg et al. [89], Abali [1], and Schöberl [125]. The advantage of this approach is that Python as an interpreted language allows for rapid implementation of various computational problems and direct post-processing of most data without the need for an additional program. Further Python is open source, such that no expensive licenses are required. The actual computational code is written in C++ and must be compiled, resulting in the desired performance for a numerical code. Python and C++ can share the same memory at runtime, such that no additional overhead is introduced. This ansatz is followed by the Python API of ELSE.

In research, development is an ongoing process. Any code framework must be easy to understand and extend by new developers in this context. C++ as a programming language is not only widely used, but also known for its maintainability. This strength is used in ELSE, by relying on an Object Oriented Programming (OOP). The base design on classes in ELSE is chosen towards the mechanic's community. Here, it is common to develop code for entities of a discretization, such as finite elements. In MPM and ELSE, this focus is shifted towards material points. Nevertheless, the interchange mechanisms between material points and computational grids demand a more complex structure, outlined in detail later. It is worth noting that the standard structure of ELSE is not chosen towards performance primarily but rather to reflect known concepts from popular codes of the community.

Any simulation software should be as computationally performant as possible. Aspects of the performance of the ELSE code were mentioned before, e.g., regarding the use of C++ on numerically demanding operations. Also, for the implemented classes, memory layout is taken into account. The MPM offers huge parallelization capabilities as shown, e.g., in Huang et al. [64]. To date, the ELSE code is not shared or distributed memory parallelized, respectively. However, the code is built with parallelization in mind, and parallel speedup can be achieved in the future.

8.1 OOP framework

In order to provide an intuitive computational representation of the MPM algorithm, a set of base classes in an OOP framework is developed. These base classes can be seen as fundamental entities in an MPM algorithm. The classes are further meant to describe their interactions in an intuitive way, which is formalized by interfaces for these communications in ELSE. For an extension of the ELSE code, it is possible to develop derived classes, which minimizes efforts in these cases. For example, for a new MPM variant, a derived class can be created which inherits all required interfaces from the MPMBody core class. This is exemplarily done for all included MPM variants in ELSE, which are also used throughout this thesis. With this general structure, good maintainability and expandability are targeted.

The illustration in figure 8.1 gives an overview of the core classes in ELSE and their main interactions. The MPMMaterial class is the base class for material models to be used to compute stresses. They carry material parameters and provide a stress function. In principle, they interact exclusively with any MPMBody to update the stress tensors at the material points. The MPMBody represents a physical body in the numerical model, discretized by material points. Hence, it stores all material point data. Further, it has to provide routines to compute contributions to the global system, map solutions to the material points, and update them. Hereby, an MPMBody has to provide algorithms to compute mapping relations by means of the core class MPMMap. The MPMMap class is a variable container. This class supports variable numbers of grid nodes, a material point is connected to for a more general mapping relation. To create an MPMMap, the MPMBody needs a description of the computational grid to describe the connectivity. Hence the main aspect for an MPMGrid is to detect which grid nodes are connected to a specific material point, i.e., coordinate, and provide the associated grid shape functions. A connection is thereby defined by a nonzero shape function evaluated at that coordinate. Notice that the MPMBody requests this information from the MPMGrid but may modify them before actually using them for the MPMMap. This aspect enables the flexible implementation of many MPM variants. A great advantage of the MPMMap is that it provides a set of node indexes with nonzero shape functions, which benefits computational efficiency when it is used. The MPMRigidBody base class represents rigid bodies that are included in the simulation. Their representation is flexible, e.g., boundary meshes might be used. In this thesis, derived classes of type MPMRigidBody contain contact algorithms to compute contact forces for each material point of an MPMBody in contact.

The MPMSolver represents the solution algorithm for a simulation. In this thesis, the USL Leapfrog integration scheme is used, which is also outlined in the upcoming paragraphs in more detail. A potential alternative might be implemented deriving from the MPMSolver class, which provides methods that, for example, time integrate over a given number of time-steps. The CBG represents the global system for the simulation. The design decision to have a CBG class and an MPMGrid class is reasoned by flexibility. While the MPMGrid represents the geometric grid and shape functions, the CBG is rather just a mathematical representation. It is used to provide an efficient assembly and solution mechanism to the system of equations that arise in each time-step.

The top-level authority for an `ELSE` simulation is represented by the MPMDomain. It is implemented as a singleton, which guarantees the uniqueness of its data. This class keeps track of simulation-sensitive data such as current time and the number of simulation steps. Further, it provides pointers to all entities, i.e., objects, inside the running simulation. Although with performance penalties, all data is available through the MPMDomain. Contrary to all other classes, inheritance is not intended from the MPMDomain.

8.2 Specific implementation of USL algorithm in `ELSE`

In this section, the solution algorithm is outlined which is used throughout this thesis. Hereby we make use of the derived classes `RegularGrid_A` for the grid, `Explicit_A` as the MPMSolver and `SMP` as an MPMBody implementation for generic MPM. For other variants of MPM which are discussed later in this thesis, algorithm details might vary. It can be assumed that a setup is given, where the solver is initialized with the aforementioned grid to be the frame of reference, and at least one body to be considered for the time integration.

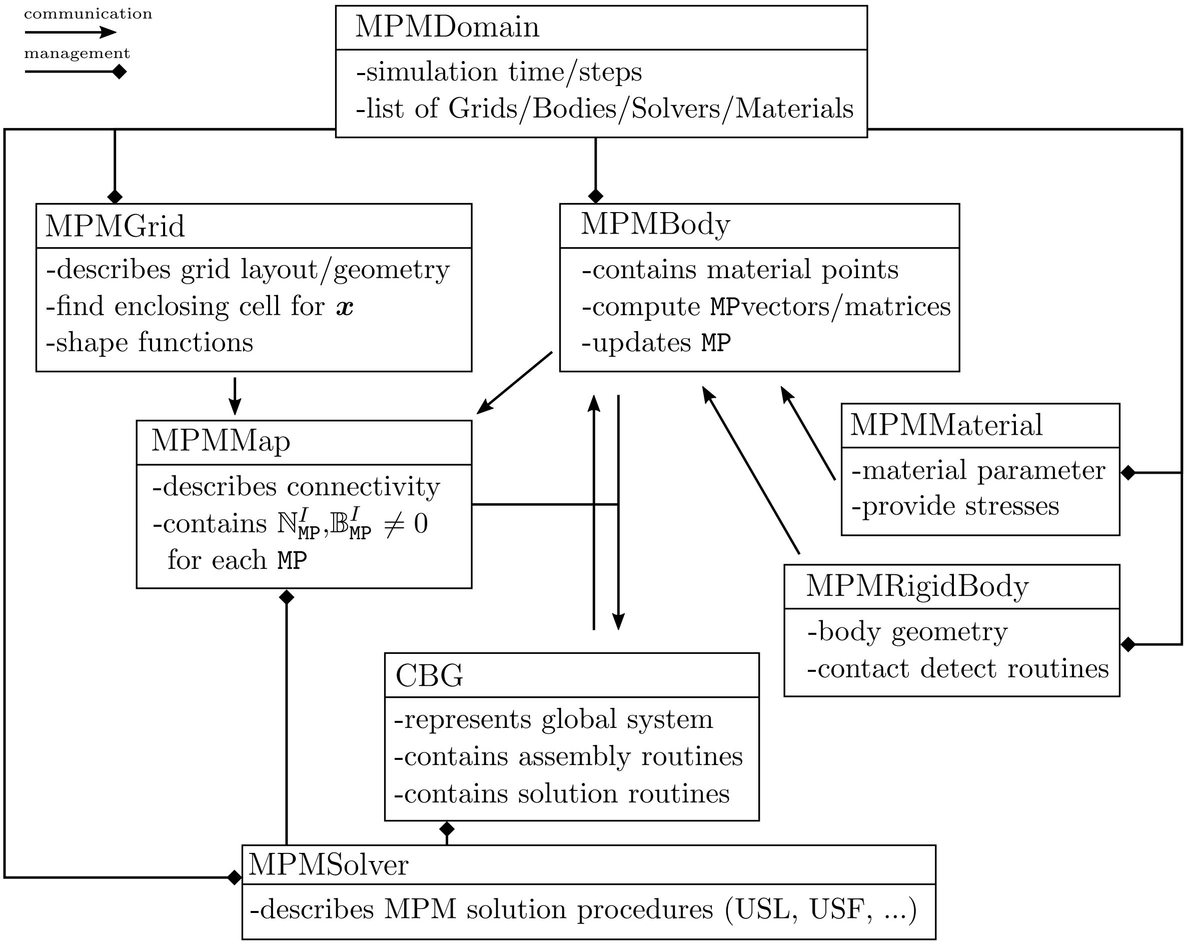

communication
management
MPMDomain
-simulation time/steps
-list of Grids/Bodies/Solvers/Materials
MPMGrid
-describes grid layout/geometry
-find enclosing cell for $\boldsymbol{x}$
-shape functions
MPMBody
-contains material points
-compute MPvectors/matrices
-updates MP
MPMMap
-describes connectivity
-contains $\mathbb{N}^I_{\mathtt{MP}}$,$\mathbb{B}^I_{\mathtt{MP}} \neq 0$
for each MP
MPMMaterial
-material parameter
-provide stresses
MPMRigidBody
-body geometry
-contact detect routines
CBG
-represents global system
-contains assembly routines
-contains solution routines
MPMSolver
-describes MPM solution procedures (USL, USF, ...)

Figure 8.1: Scheme for an interaction of the core classes in an `ELSE` simulation.

8.2.1 Time-step initialization

At the beginning of a new time-step, all entities in the simulation advance in time. That means that all persistent properties that are tagged as *current* are switched to the states *previously*. Following the class structure in `ELSE`, the MPMDomain updates the simulation clock and step counter while each MPMBody takes care of its set of associated material points. As outlined in algorithm 7, the CBG is reset. That does delete all nodes from the CBG following its temporary character.

```
/* advance domain in time                                          */
$t \leftarrow t + \Delta t$, $step \leftarrow step + 1$

/* advance MPMBodies in time                                       */
foreach MP do
    $\boldsymbol{x}_{\mathtt{MP}_n} \leftarrow \boldsymbol{x}_{\mathtt{MP}}$
    $\boldsymbol{v}_{\mathtt{MP}_n} \leftarrow \boldsymbol{v}_{\mathtt{MP}}$
    $\boldsymbol{F}_{\mathtt{MP}_n} \leftarrow \boldsymbol{F}_{\mathtt{MP}}$
end

/* reset CBG                                                       */
CBG.reset() /* deletes all nodes and nodal information             */
```

Algorithm 7: Procedures at the beginning of a time-step for the USL algorithm in `ELSE`.

By design, all classes in `ELSE` provide a `NextStep` function, called by the solver at the beginning of each time-step, to trigger this behavior.

8.2.2 Update mapping operators

In the USL solution scheme, shape functions are evaluated once at the beginning of a time-step. For each material point in the simulation, a temporary MPMMap is allocated. Every material point then requests the MPMGrid for all nonzero shape functions defined on the grid and the node indexes where they occur. Section E of the appendi, provides the specific implementation of a regular grid currently implemented in `ELSE`. For the actual implementation of algorithm 8, the MPMBody class performed the loop over its associated material points. While it requests the MPMGrid for shape functions, an MPMBody may modify these before they are stored as an MPMMap for each material point.

```
/* update mappings                                                 */
foreach MP do
    $\mathrm{MPMMap}_{\mathtt{MP}} : \mathbb{N}^{\mathtt{I}}_{\mathtt{MP}}, \mathbb{B}^{\mathtt{I}}_{\mathtt{MP}} \leftarrow \mathbb{N}^{\mathtt{I}}(\boldsymbol{x}_{\mathtt{MP}}), \mathbb{B}^{\mathtt{I}}(\boldsymbol{x}_{\mathtt{MP}})$
end
```

Algorithm 8: Compute mapping operators for the USL algorithm in `ELSE`.

8.2.3 Global system assembly

With respect to the introduced scheme, the global system is represented by a vector of nodes. The ELSE code reflects this by using the CBG class, which provides a container for node objects. Each node class has properties as given in table 7.1. The assembly process in this way is straightforward. In ELSE, the main loop over material points in algorithm 9, is again performed by the MPMBody. Notice that for each material point, a contribution to the nodal properties of mass m^I, linear momentum $\boldsymbol{p}^{\mathtt{I}}$, and force $\boldsymbol{f}^{\mathtt{I}}$ is made. In this sense, the algorithm is very similar to a FE procedure, where nodal contributions are computed similarly. However, in comparison, the stresses are not computed as in the implicit FEM in this step. Also, the shape functions used for to compute the global contributions are computed in the previous step in MPM, as they require communications with, e.g., the MPMGrid. In the FEM, usually all these steps are combined within one subroutine, evaluated for each element.

```
Input: CBG: $m^{\mathtt{I}} = 0$, $\boldsymbol{p}^{I} = \mathbf{0}$, $\boldsymbol{f}^{\mathtt{I}} = \mathbf{0}\ \forall\ \mathtt{I} \in \mathtt{NoI}$

/* assemble global vectors                                        */
foreach MP do
    foreach I$\in$ (MPMMap)$_{MP}$ for which $\mathbb{N}^{I}_{MP} \neq 0$ do
        $m^{\mathtt{I}} \leftarrow m^{\mathtt{I}} + \mathbb{N}^{\mathtt{I}}_{\mathtt{MP}}\, \rho_{\mathtt{MP}}\, \mathrm{v}_{\mathtt{MP}}$
        for $i \leftarrow 1$ to 3 do
            $p^{\mathtt{I}}_i \leftarrow p^{\mathtt{I}}_i + \mathbb{N}^{\mathtt{I}}_{\mathtt{MP}}\, \rho_{\mathtt{MP}}(\mathrm{v}_{\mathtt{MP}})_i\, \mathrm{v}_{\mathtt{MP}}$
            $f^{\mathtt{I}}_i \leftarrow f^{\mathtt{I}}_i + \mathbb{N}^{\mathtt{I}}_{\mathtt{MP}}\, (\mathfrak{b}_{\mathtt{MP}})_i\, \mathrm{v}_{\mathtt{MP}}\, \rho_{\mathtt{MP}}$
            for $j \leftarrow 1$ to 3 do
                $f^{\mathtt{I}}_i \leftarrow f^{\mathtt{I}}_i - \left(\mathbb{B}^{\mathtt{I}}_{\mathtt{MP}}\right)_j\, (\sigma_{\mathtt{MP}})_{ij}\, \mathrm{v}_{\mathtt{MP}}$
            end
        end
    end
end
```

Algorithm 9: Form global system for the USL algorithm in ELSE.

8.2.4 Solution on the CBG

Once all material properties are available in the CBG object, it is used to compute the solution. Algorithm 10 shows the procedure, while we omit time indicators, as at this point of the algorithm, one variant of nodal properties exists. As the CBG represents the system with using a lumped mass matrix, the system is uncoupled, and the solution is computed for each node I individually, resulting in a very good performance. In the displayed algorithm, the numerical damping strategy from equation (7.24) is incorporated. Hence, artificial damping for any $\alpha_d > 0$ is introduced. Further, the application of boundary conditions is shown. Here, prescribed motion in terms of constant nodal velocity $\bar{\boldsymbol{v}}^{\mathtt{I}}$ as described in equation (7.28) is actually used. A constant velocity, which includes resting by $\bar{\boldsymbol{v}}^{\mathtt{I}} = \mathbf{0}$, covers most cases, including all examples in this thesis.

```
/* compute solution on CBG                                          */
foreach $I$ do
    if $m^I \leq tol$ then
        for $i \leftarrow 1$ to 3 do
            $a_i^{\mathtt{I}} \leftarrow (f_i^{\mathtt{I}} - \alpha_d \, p_i^{\mathtt{I}}) \, \frac{1}{m^{\mathtt{I}}}$;      $v_i^{\mathtt{I}} \leftarrow p_i^{\mathtt{I}} \, \frac{1}{m^{\mathtt{I}}}; + a^{\mathtt{I}} \, \Delta t$
        end
        if $\boldsymbol{x}^I$ is subject to BC then
            for $i \leftarrow 1$ to 3 do
                $a_i^{\mathtt{I}} \leftarrow 0$;    $v_i^{\mathtt{I}} = \bar{v}_i^{\mathtt{I}}$
            end
        end
    end
end
```

Algorithm 10: Compute solution of current time-step on the CBG for the USL algorithm in ELSE.

8.2.5 Mapping of CBG solution to each material point

At this point of the time-step, the solution from the CBG is projected onto the material points. The task is performed again completely by any MPMBody object for its managed material points in communication with the CBG. Notice that still, this operation utilizes the mapping operators $\mathbb{N}_{\mathtt{MP}}^{\mathtt{I}}$ and $\mathbb{B}_{\mathtt{MP}}^{\mathtt{I}}$ computed at the beginning of the time-step. The implementation in algorithm 11 additionally requires a temporary velocity $\tilde{\boldsymbol{v}}_{\mathtt{MP}}$. Also, the velocity gradient $\boldsymbol{L}_{\mathtt{MP}}$ and acceleration $\boldsymbol{a}_{\mathtt{MP}}$ technically only temporarily required, as they are re-computed in each time-step.

```
/* mapping solution to the material points                          */
foreach $MP$ do
    /* initialize                                                    */
    $\boldsymbol{a}_{\mathtt{MP}} \leftarrow \boldsymbol{0}$, $\boldsymbol{v}_{\mathtt{MP}} \leftarrow \boldsymbol{0}$, $\boldsymbol{L}_{\mathtt{MP}} \leftarrow \boldsymbol{0}$
    /* map solution to each MP                                       */
    foreach $I \in$ MPMMap$_{MP}$ for which $\mathbb{N}_{MP}^{I} \neq 0$ do
        for $i \leftarrow 1$ to 3 do
            $(a_{\mathtt{MP}})_i \leftarrow (a_{\mathtt{MP}})_i + \mathbb{N}_{\mathtt{MP}}^{\mathtt{I}} \, a_i^{\mathtt{I}}$
            $(\tilde{v}_{\mathtt{MP}})_i \leftarrow (\tilde{v}_{\mathtt{MP}})_i + \mathbb{N}_{\mathtt{MP}}^{\mathtt{I}} \, v_i^{\mathtt{I}}$
            for $j \leftarrow 1$ to 3 do
                $(L_{\mathtt{MP}})_{ij} \leftarrow (L_{\mathtt{MP}})_{ij} + (\mathbb{B}_{\mathtt{MP}}^{\mathtt{I}})_j \, v_i^{\mathtt{I}}$
            end
        end
    end
end
```

Algorithm 11: Mapping nodal solutions from the CBG to the material points for the USL algorithm in ELSE.

8.2.6 Update stress last - finalizing time-step

At the end of any time-step in the procedure, the solution is available at the material points. Thus, the update of all persistent material point properties can be performed, again using the MPMBody classes for the material point loops. The implementation given in algorithm 12 uses the temporary velocity $\tilde{v}_{\mathtt{MP}}$ for an update of the material points position. After this step, this property is not required anymore. Further, the velocity, deformation gradient, and associated volume are updated according to the properties from the previous step. At last, the Cauchy stress tensor is computed for each material point. Implementation-wise, the constitutive routine is provided by the MPMMaterial. The update of the material density must be done inside this routine by convention. If the material model uses history-dependent parameters, these are updated by the subroutine as well.

```
/* final update of the material points                                */
foreach MP do
    for $i \leftarrow 1$ to 3 do
        /* update position                                            */
        $(x_{\mathtt{MP}})_i \leftarrow (x_{\mathtt{MP}_n})_i + (\tilde{v}_{\mathtt{MP}})_i \, \Delta t$
        /* update velocity                                            */
        $(\mathrm{v}_{\mathtt{MP}})_i \leftarrow (x_{\mathtt{MP}_n})_i + (a_{\mathtt{MP}})_i \, \Delta t$
        /* update deformation gradient                                */
        for $j \leftarrow 1$ to 3 do
            $(F_{\mathtt{MP}})_{ij} \leftarrow 0$
            for $k \leftarrow 1$ to 3 do
                $(F_{\mathtt{MP}})_{ij} \leftarrow (F_{\mathtt{MP}})_{ij} + (\delta_{ik} + (L_{\mathtt{MP}})_{ik} \, \Delta t) \; (F_{\mathtt{MP}_n})_{kj}$
            end
        end
        /* update material point volume                               */
        $v_{MP} \leftarrow \det(\boldsymbol{F}_{\mathtt{MP}}) \; v_{MP_0}$
        /* update stress last                                         */
        $\boldsymbol{\sigma}_{\mathtt{MP}}, \boldsymbol{h}_{\mathtt{MP}} \leftarrow mate\_call(\boldsymbol{F}_{\mathtt{MP}}, \boldsymbol{h}_{\mathtt{MP}_n})$
    end
end
```

Algorithm 12: Final update of time dependent properties at material points for the USL algorithm in ELSE.

8.3 Verification of USL implementation - vibration of a material point

The simplest form of an MPM problem, is one material point placed in a single computational cell. This problem was first analyzed by Bardenhagen [10]. As loading conditions in this transient problem, an initial velocity is assigned to the material point while one side of the grid is fixed. These conditions cause the material point to vibrate.

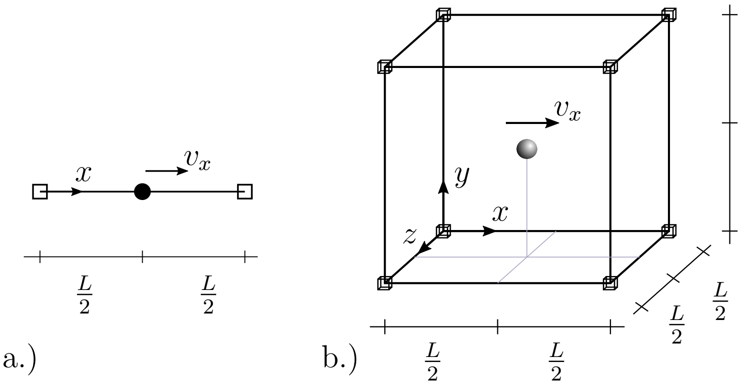


Figure 8.2: Illustration of the single material point vibration setup in the original 1-dimensional setup a.) and in the 3-dimensional framework of ELSE b.).

The discretization involves a single material point inside a single cell of length L, as depicted in figure 8.2. Initially the material point is located in the geometric center of the cell and loaded with an initial velocity v_x. The problem parameters are summarized in table 8.1. Although in Bardenhagen [10], a linear elastic material is used, whereas here, the hyper-elastic material from equation (4.13) is used. The time-step size is computed from the one-dimensional wave speed according to equation (7.22), with $\Delta t = 0.1\,L/c$, using 0.1 as a safety factor analog to Bardenhagen [10].

Table 8.1: Discretization properties of the single material point vibration problem.

$\mathrm{v}_{\mathtt{MP}}$	$= 1.0\ \mathrm{m}^3$	ρ	$= 1.0\ \mathrm{kg\,m}^{-3}$
c	$= 2\pi\ \mathrm{m\,s}^{-1}$	L	$= 1.0\ \mathrm{m}$
E	$= 4\,\pi^2\ \mathrm{N\,m}^{-2}$	ν	$= 0.0$
$\boldsymbol{X}_{\mathtt{MP}}$	$= (0.5, 0.5, 0.5)^T\ \mathrm{m}$	$\boldsymbol{v}_{\mathtt{MP}}$	$= (0.1, 0.0, 0.0)^T\ \mathrm{m\,s}^{-1}$
T_{end}	$= 10\ \mathrm{s}$	Δt	$= 0.01591\ \mathrm{s}$

Further, a reference solution for the material points velocity $v(t)$ and position $x(t)$ was derived in Bardenhagen [10], which renders a harmonic oscillation, with frequency $\omega = \sqrt{E/\rho}$,

$$v(t) = v_0\,\cos(\omega\,t), \quad x(t) = x_0 \exp\left[\frac{v_0}{L\,\omega}\,\sin(\omega\,t)\right]. \tag{8.1}$$

The results obtained for this problem with the ELSE code are given in figure 8.3. In order to fix the grid, essential boundary conditions were applied to the nodes which match the criterion $x = 0$. Compared with the reference solution from equation (8.1), the numerical results show strong damping. However, the harmonic oscillation matches the predicted frequency and hence wavelength perfectly in both velocity and position of the single material point.

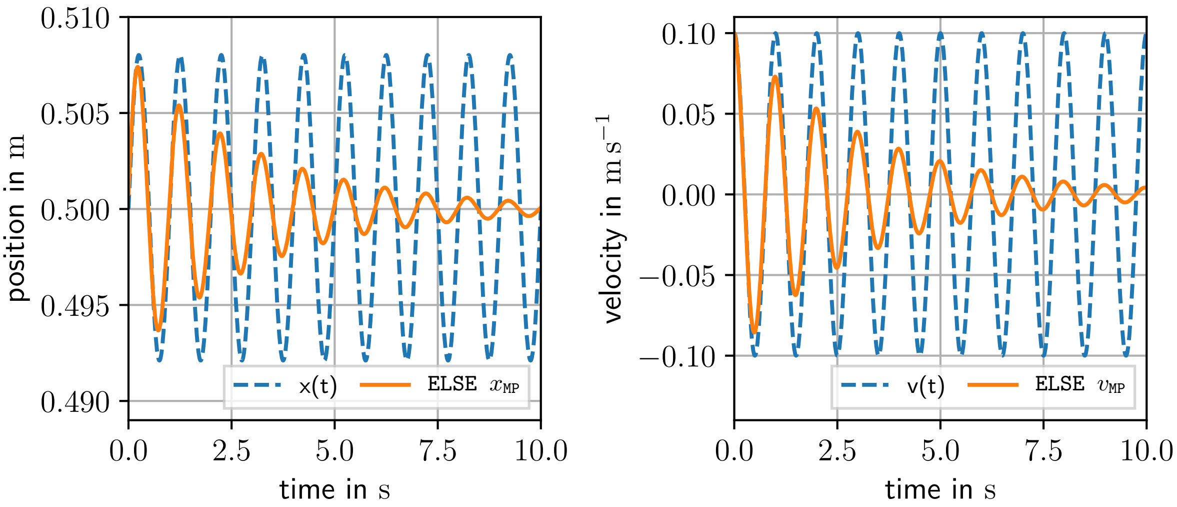


Figure 8.3: MPM results for the vibration of a single material point obtained from `ELSE`. Given are the position of the material point and its velocity over time, in comparison to the analytical solutions from equation (8.1).

In order to investigate the damping behavior further, the kinetic and internal energies of the single material point are computed via

$$\mathcal{K}_{\mathtt{MP}} = \frac{1}{2}\,\boldsymbol{v}_{\mathtt{MP}} \cdot \boldsymbol{v}_{\mathtt{MP}}\, m_{\mathtt{MP}}, \qquad \text{and} \qquad \mathcal{E}_{\mathtt{MP}} = \frac{1}{2}\,(\boldsymbol{F}_{\mathtt{MP}} - \boldsymbol{I}) : \boldsymbol{\sigma}\, \mathrm{v}_{\mathtt{MP}}, \tag{8.2}$$

compare Bardenhagen [10]. The strong dissipation of energy over time is clearly shown in figure 8.4. The observations match exactly the results from Bardenhagen [10], where the dissipation effect is a property of the USL scheme. As reported by many publications, this dissipation effect does not dominate USL simulations in the general case but rather stabilizes a numerical analysis. Also, the dissipative error can be controlled by varying numerical time-step sizes as, shown in figure 8.4. Here, the analysis was not just done with the time-step size estimated values (compare table 8.1) but also choosing significantly smaller values. As a result of this modification, the damping is reduced for this example.

The original version of the problem as introduced in Bardenhagen [10], renders a one-dimensional problem, see figure 8.2 a.). It was just shown that a three-dimensional setup could be used to reproduce the results exactly. However, the implemented explicit 3D-MPM based on the USL algorithm has a critical mesh dependency on simulations with spatial restrictions. Therefore, it is not sufficient to prescribe the motion of the grid nodes. e.g., to 1D or a 2D plane. For the vibration of a single material point, the material point was initially placed in the center of the single computational cell. One could also use a discretization with the material point initially on the x-axis, as depicted in figure 8.5. Nevertheless, with this discretization in the 3D `ELSE` code, the grid deforms differently to the first-mentioned case, introducing shearing effects on the grid. As these modes are then propagated to the material point, it exhibits spurious shear stresses. Consequently, the oscillation of the material point is different in such cases. For comparison, figure 8.6 shows the motion for the two cases discussed in figure 8.5. The differences in these grid deformations with respect to the initial placements of the material point are qualitatively illustrated in figure 8.5.

From the discussed results, it can be concluded that the USL algorithm is implemented

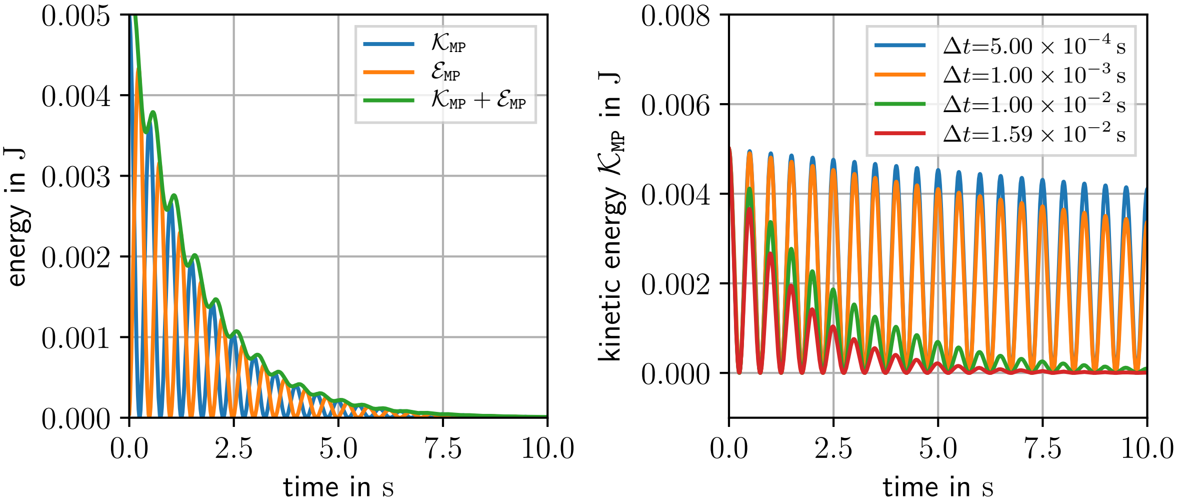


Figure 8.4: Evolution of energy over time for the vibration of a single material point in ELSE. The left graph shows kinetic and internal energy in comparison with their sum, while the right graph shows the influence of the computational time-step size on the dissipation.

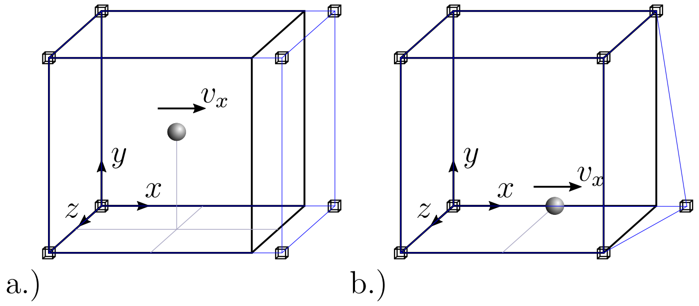


Figure 8.5: Qualitative illustration of the grid deformation within a single time-step, based on the initial placement of the single material point. In order to obtain a perfect match with the 1D-simulation, the material point must be placed in the center of the CBG (a.). Initial positions deviating from the center (b.) introduce spurious shear modes on the CBG.

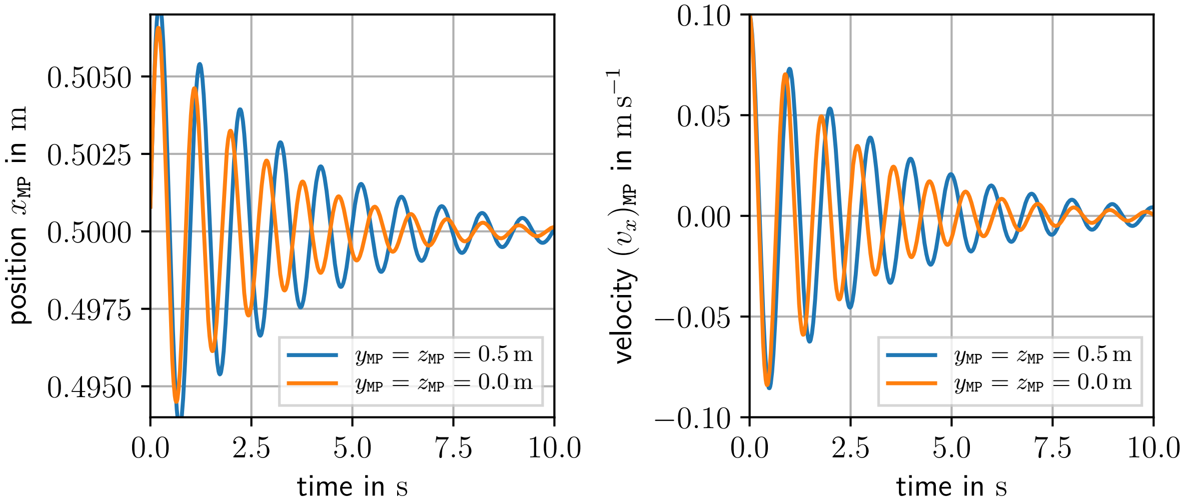


Figure 8.6: Results for the vibration of a single material point with alternating initial placement. A centered initial placement with $y_{\mathtt{MP}} = z_{\mathtt{MP}} = 0.5\,\mathrm{m}$ results in the correct damped oscillation, while $y_{\mathtt{MP}} = z_{\mathtt{MP}} = 0.0\,\mathrm{m}$ results in spurious shear and a different motion.

correctly. Also, as expected, the used nonlinear hyper-elastic material law covers the linear theory in the small deformation case. For all simulations with spatial restrictions, the material points must be placed in the symmetry lines of the grid for the restricted dimensions. The ELSE code can be used for 1D or 2D simulations following this conclusion together with appropriate boundary conditions on the grid

8.4 Verification of boundary conditions and material implementation - Cook's Membrane

The Cook's membrane problem is a common benchmark for finite elements, as it combines large bending and shear deformations with a numerically challenging stress singularity. For this analysis, the problem definition from Schröder et al. [126] is adopted. The geometry of Cook's membrane is illustrated in figure 8.7, along with the control points A and B as well as the boundary conditions. The problem demands full support at surface S_s, and surface S_t is subjected to constant traction in the y direction $p_y = 20.0 \times 10^6\,\mathrm{Pa}$.

For the MPM, the boundary conditions for the support are defined on the CBG by means of prescribed zero nodal velocities. For the CBG of this problem, a uniform cell size of $dx = dy = dz = 2.0 \times 10^{-3}\,\mathrm{m}$ is chosen. The discretization of the elastic membrane by material points follows a regular hexahedral sub-grid on the domain, where each center is taken as a material point with an associated volume that equals the volume of the hexahedral cell. Notice that this way, the sum of all material points volume is equal to the geometry, given in figure 8.7. For this example, a convergency study is carried out with respect to the number of material points. The sub-grid strategy allows for a parametrized refinement of the discretization, respectively.

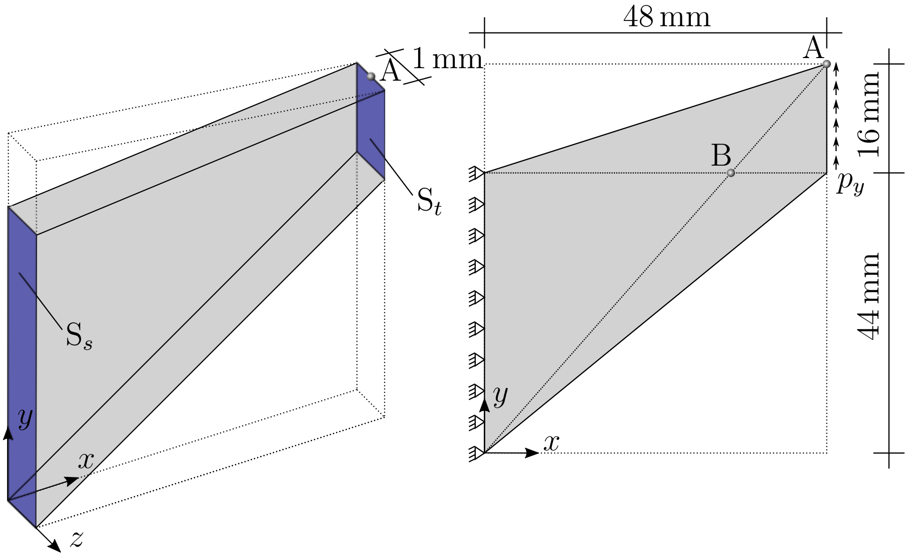


Figure 8.7: Cook's membrane problem illustration. The three-dimensional membrane domain is subject to full support on surface S_s, and surface S_t is subject to constant traction. Control points A and B are introduced analog to Schröder et al. [126].

However, the traction boundary condition in the original problem cannot be implemented in the present MPM scheme directly. The most obvious reason is that the MPM discretization does not discretize the surface S_t, where the load should be applied. In equation, (7.31), a strategy was presented to approximate external traction by volumetric force contributions $\bar{\boldsymbol{t}}_{\mathtt{MP}}$ and a correction factor $\bar{h}_{\mathtt{MP}}$ selectively on the material points closest to an edge. This strategy is adopted here. Additionally, the traction load from the original problem is defined in the reference configuration. As the here applied MPM relies on an updated Lagrangian definition of the balance of momentum, it only supports loads defined in the current configuration. This aspect is completely neglected here. Hence, the load is treated in the current configuration with the same magnitude, introducing an error when comparing with Schröder et al. [126]. We observe that this error is very subtle for the present problem. In order to compute $\bar{\boldsymbol{t}}_{\mathtt{MP}}$ and $\bar{h}_{\mathtt{MP}}$, we use the equality

$$\int_{\mathrm{S}_t} p_y \,\mathrm{da} \;\;= \sum_{\mathtt{MP}=1}^{\bar{\mathtt{MP}}} \bar{\boldsymbol{t}}_{\mathtt{MP}}\, \bar{h}_{\mathtt{MP}}\, \mathrm{v}_{\mathtt{MP}}, \tag{8.3}$$

where $\bar{\mathtt{MP}}$ defines the set of material points which are closest to the surface S_t. The underlying assumption is that the total force which is applied externally is equal. The computation is carried out for each discretization, as $\bar{\mathtt{MP}}$ directly depends on this.

The problem is analyzed on a plane strain constraint using grid boundary conditions as discussed in chapter 8.3. In contrast to Schröder et al. [126], the hyper-elastic material of equation (4.13) is used for this analysis. The formulations are similar, though, so we adopt the same elastic parameters, i.e., $E = 500.0 \times 10^6\,\mathrm{Pa}$ and $\nu = 0.35$.

In order to compute a steady-state solution, artificial density and numerical damping were used in a dynamic relaxation strategy as mentioned in chapter 6. Figure 8.8 shows the vertical displacement of point A and the stresses in point B over the virtual time, to show that a static equilibrium is actually achieved. These points were added to the simulation, to evaluate the material points solution at the control points. A very small associated volume was assigned to each of them to minimize their effects. One can think of these control points as virtual material points.

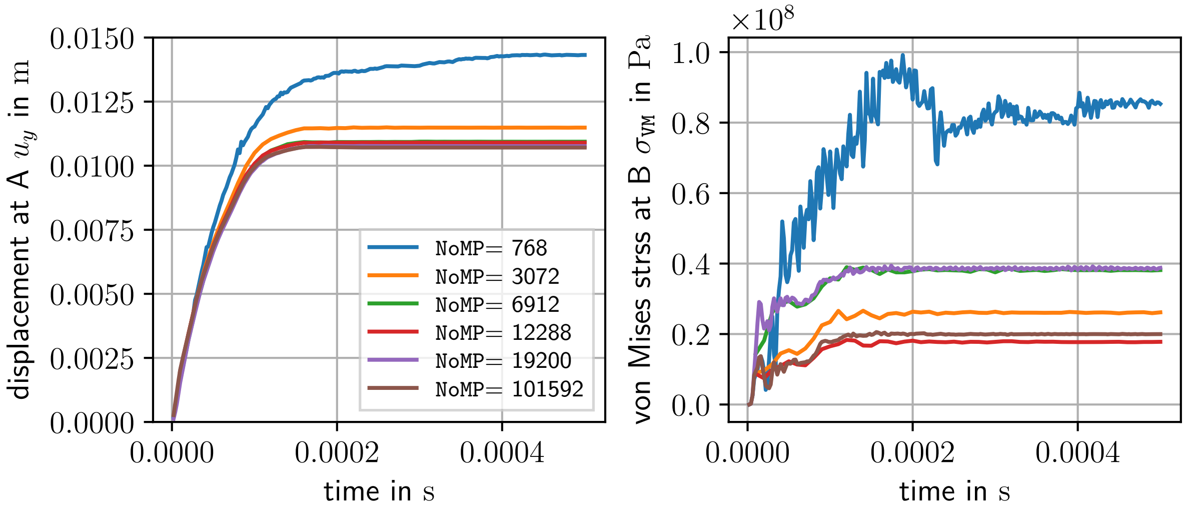


Figure 8.8: Dynamic relaxation procedure for the convergency study of Cook's membrane, to compute static equilibrium.

Although the CBG is fixed, smaller distances of the material points require different parameters for a stable simulation. For this analysis, time-step size between $\Delta t = 1.0 \times 10^{-7}\,\mathrm{s}$ and $\Delta t = 2.0 \times 10^{-8}\,\mathrm{s}$, artificial damping with $\alpha_d = 5.0 \times 10^4\,\mathrm{s}^{-1}$ and a virtual density of $100.0\,\mathrm{kg\,m^{-3}}$ were used. The set total number of material points on the discretizations considered in the convergency study are also listed in figure 8.8.

The actual convergence of the problem is analyzed in figure 8.9. It can be seen that the vertical displacement of point A shows a strong convergence with finer discretization towards $u_y = 10.708 \times 10^{-3}\,\mathrm{m}$ with NoMP= 101592. In comparison with the FEM solutions in Schröder et al. [126], where a converged displacement of $u_y \approx 10.6 \times 10^{-3}\,\mathrm{m}$ is reported, the MPM solution is quite satisfying, especially given the mentioned differences in material and boundary conditions. In this analysis, also the convergency of von Mises stresses in point B is investigated, see figure 8.9. In contrast, it does not converge as straight as the displacement in point A. Nevertheless, a quick convergence in scale can be observed as well, given that the obtained results vary by a factor of 2.0 at maximum.

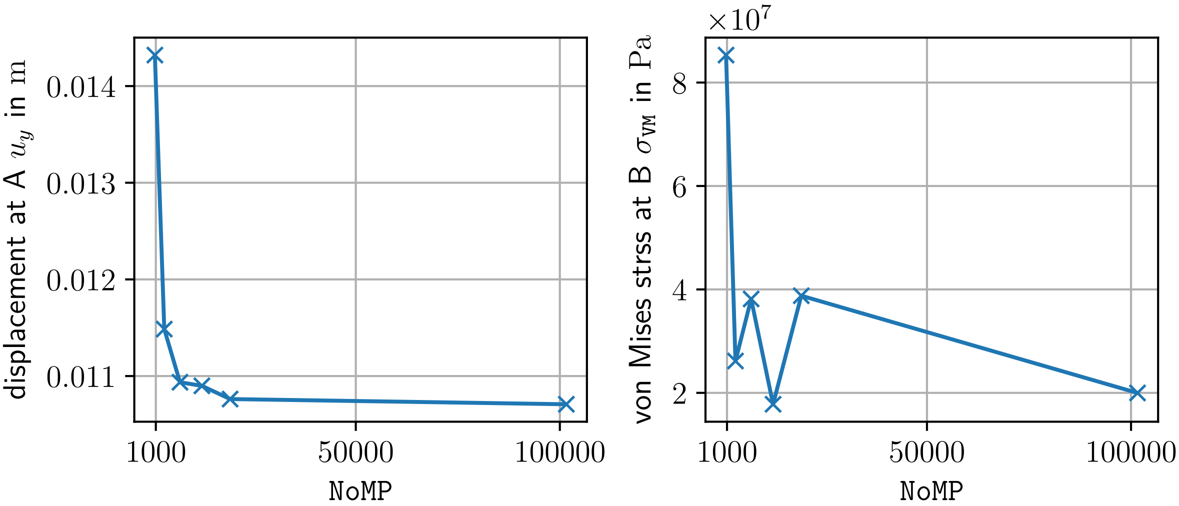


Figure 8.9: Convergency study for Cook's membrane. The vertical displacement of point A strongly converges with NoMP. The von Mises stresses in point B show more deviations.

For a better view on the stress approximation obtained with the MPM with varying NoMP, figure 8.10 shows the σ_{xx} component over the deformed elastic domain, respectively. The MPM results, already in the second discretization with NoMP= 3072 show very good agreement with the stresses in Schröder et al. [126], in both magnitude and distribution in space. For solid mechanical problems, the resolution of a body with material points also effects the approximation quality of the geometry. In each of the presented cases in figure 8.10, the deformation pattern of Cook's membrane is physically reasonable and equal to the literature.

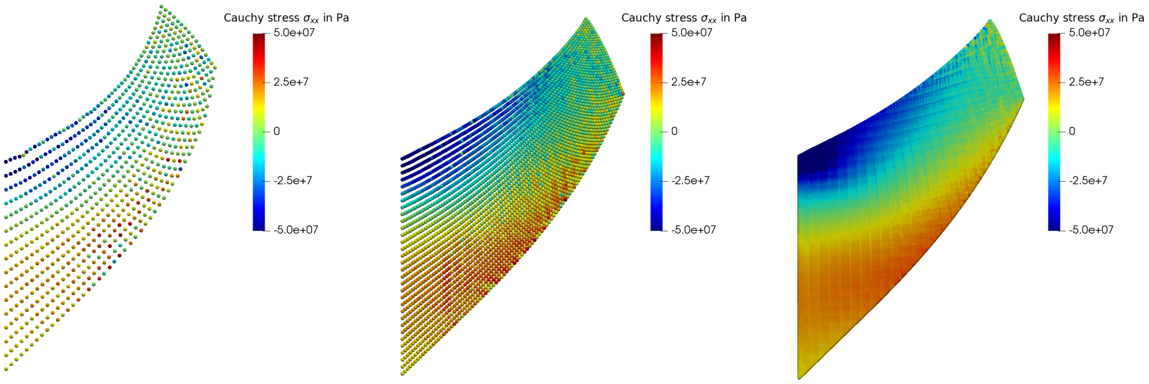


Figure 8.10: MPM results for Cook's membrane discretized with 768, 3072 and 101592 material points. The graphics show the σ_{xx} Cauchy stresses for a direct comparison with Schröder et al. [126].

9 Contact in MPM

Contact mechanics are a major requirement for numerical modeling of most engineering problems. Usually, contact algorithms are one of the most expensive aspects in a simulation with respect to computational time. In order to simulate contact conditions, two aspects must be provided. The first one is contact detection, and the second one is the actual contact model. Contact models are highly dependent on the chosen discretization method and are used to enforce certain contact conditions in a simulation. Most commonly, non penetration conditions are used in combination with a suitable friction law. Its execution, however, can be neglected in comparison to the prior contact search in terms of computational effort. The interested reader is referred to Wriggers [157] or Zienkiewicz and Taylor [165] for a general overview of both.

MPM contact variants relevant for this thesis can be categorized as

1 **MPM body to MPM body**: Two bodies, defined by sets of material points, exchange contact interactions with each other.

2 **MPM-Body to MPM-Grid**: A MPM body gets in contact with another body, which is not explicitly modeled but reflected in the CBG, where the non-penetration and sliding/sticking conditions are enforced.

3 **MPM-Body to Geometric-Body**: A MPM-Body without a defined body surface, gets in contact to another body, discretized and modeled by means of another method (e.g., FEM) interact with each other.

9.1 MPM body-body contact

In MPM, a contact interaction between two bodies, each discretized by material points can be observed without further implementation because all material points, no matter their body association, share the same velocity field on the CBG. Hence, the standard MPM enforce a non penetration, perfect stick condition for MPM bodies. Additionally, this feature does not produce any additional computational costs. The contact search, required by other discretzation methods, is not needed as the contact is naturally resolved by mapping into a single CBG. The disadvantage of this phenomenon is that the perfect stick condition is often not suitable.

In order to investigate this behavior, a standard benchmark in a modified version is employed here. Originally, the problem is defined by a pair of circular disks which collide under plane strain conditions. In this form, the setup was used as a benchmark in Sulsky et al. [142] to validate the two dimensional MPM implementation and from there on utilized in multiple publications, e.g., in Coetzee [30], de Vaucorbeil et al. [36].

In the following, the problem is transferred into three dimensions. Two spheres with centroids $\boldsymbol{x}_1^c$ and $\boldsymbol{x}_2^c$ with the same diameter r are discretized with the MPM. The spheres collide eventually, as an initial velocity $\boldsymbol{v}$ is assigned to all material points of each sphere, in opposite directions. A graphical illustration of the setup is given in figure 9.1 alongside the relevant simulation parameter. The hyper-elastic material from equation (4.13) is used. The simulation is carried out on a regular CBG as described in E with uniform grid spacing dx. The exact collision time can be computed from geometrical observations with

$$t_a = \frac{\|\boldsymbol{x}_1^c - \boldsymbol{x}_2^c\| - 2\,r}{2\,\|\boldsymbol{v}\|} = 1.845\,\mathrm{s} \tag{9.1}$$

with executed accuracy of the simulation, $\Delta t = 1 \times 10^{-3}\,\mathrm{s}$. In order to investigate the contact during the simulation, two control points A and B are defined. Considering the MPM discretization of the spheres with a regular pattern of material points, these control points are considered to be the material point of each sphere, which is closest to the centroid of the opposite one. Hereby, a decent discretization density is chosen, with 113.153×10^3 material points per sphere.

$\boldsymbol{x}_1^c$	$= (0.8,\, 0.8,\, 0.8)^T$ m
$\boldsymbol{x}_2^c$	$= (0.2,\, 0.2,\, 0.2)^T$ m
r	$= 0.2$ m
$\boldsymbol{v}$	$= (0.1,\, 0.1,\, 0.1)^T\ \mathrm{m\,s^{-1}}$
l	$= 1.0$ m
E	$= 1.0 \times 10^3\,\mathrm{N\,m^{-2}}$
ν	$= 0.3$
ρ	$= 1.0 \times 10^3\,\mathrm{kg\,m^{-3}}$
Δt	$= 1.0 \times 10^{-3}\,\mathrm{s}$
t_{end}	$= 4.0\,\mathrm{s}$

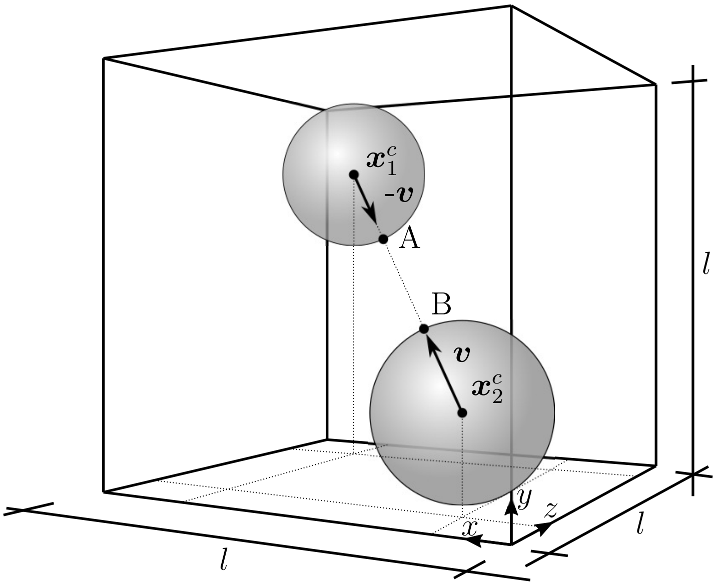


Figure 9.1: Colliding spheres problem. Simulation parameter (left) and geometrical illustration of the setup (right).

The problem parameters depicted in figure 9.1 are closely related to those in Sulsky et al. [142], to observe the same phenomena. A time series for meaning-full stages during the simulation is given in figure 9.2. The simulation can be categorized in three phases, the approaching phase, the collision phase and the separation phase. In the first phase, the spheres translate in space towards each other. Here it is essential that a constant velocity is preserved and no deformation is induced. This behavior is observed in the present simulation, considering the snapshots from $t = 0.1\,\mathrm{s}$ and $t = 1.5\,\mathrm{s}$. At a certain point, the material points associated with the two spheres interact with each other, defining the collision phase. The contact mechanism to initialize this phase was mentioned before. If two material points from different bodies happen to map their properties, e.g., linear momentum, to the same node of the CBG, these material points interact with each other. This interaction is hence very dependent on the location of the shared nodes.

Consequently, the contact is mesh-dependent. In order to study this further, a sensitivity analysis is performed on this problem. Figure 9.2 shows the snapshot during the collision at $t = 2.3\,\mathrm{s}$. Due to the contact situation, the spheres deform, which is accompanied by occurring elastic stresses. The linear momentum of the spheres shrinks, as the kinetic

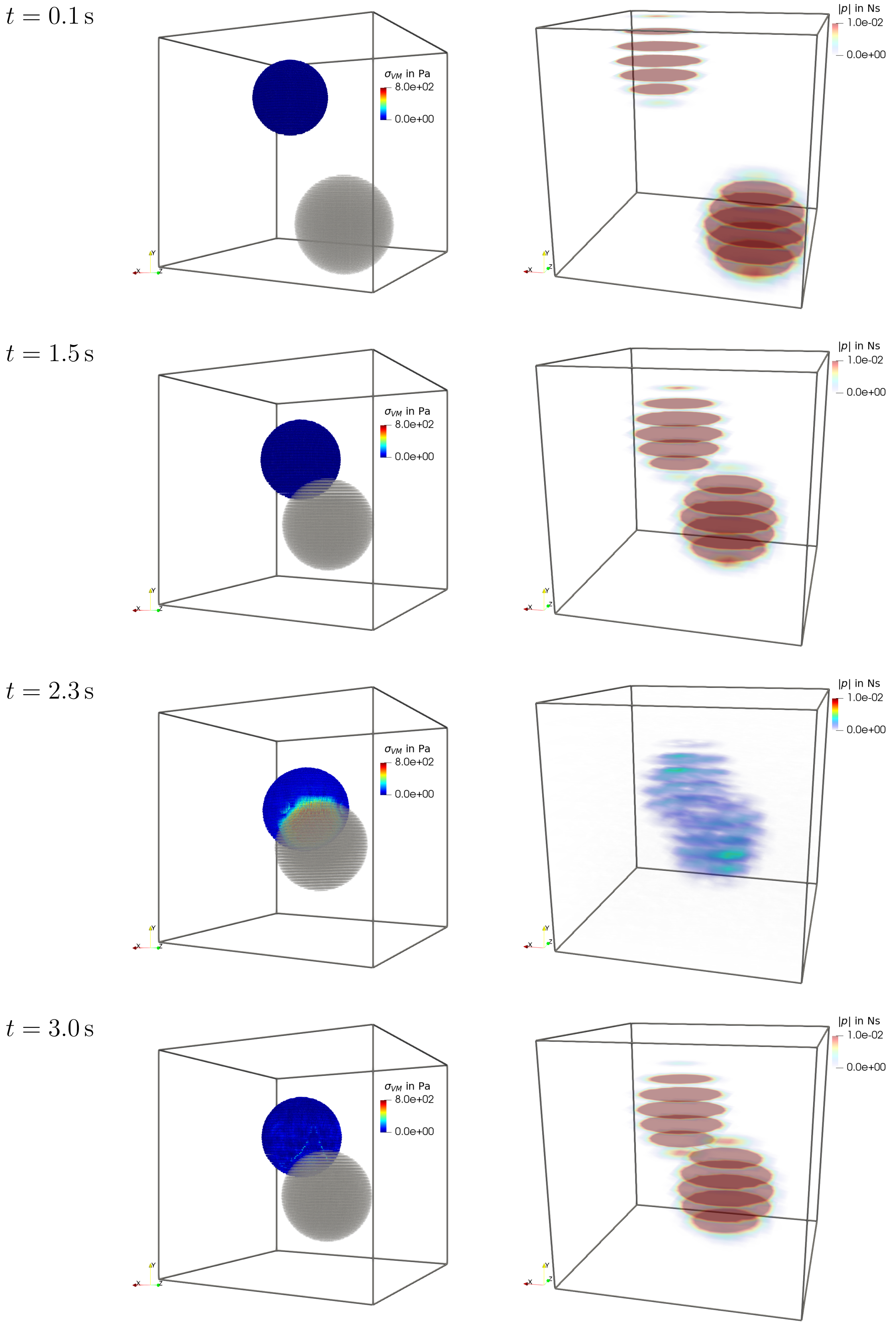


Figure 9.2: Colliding spheres problems snapshots at different time-steps with a uniform grid spacing of $\mathrm{dx} = 5 \times 10^{-2}\,\mathrm{m}$. The left column shows the material points, with von Mises stresses for the upper sphere, the right column visualizes the corresponding linear momentum field magnitude as observed on the CBG.

energy is transformed to strain energy in this phase. Notice that both spheres share the same momentum field on the CBG. Near the contact surface of the spheres, the linear momentum on the grid is actually zero, as opposite contributions from both spheres negate each other. In the separation phase, the strain energy is transferred back to kinetic energy as expected from an elastic impact, respectively. The complete separation of the spheres can be observed in the snapshot at $t = 3.0\,\mathrm{s}$. The linear momentum field on the CBG shows no connection between the contributions from both spheres anymore. However, studying the von Mises stresses of the material points, some elastic vibrations persist after the impact. This behavior is not only reasonable but also covered in the literature.

For a deeper understanding of the observed contact mechanism, the movement of the control points A, B are observed. Figure 9.3 shows their current position in the x-direction, which is exactly the same as in the y, and z direction, respectively. Four different sizes for the uniform grid spacing are chosen and given in the graph for comparison. It can be observed that the two points do not actually touch each other. This aspect is also covered by the simulation visualizations of the contact situation in figure 9.3. The reason was discussed before. It is furthermore possible to observe the mesh dependency, as, with smaller cell sizes, point A and B become closer to each other. Also, for smaller grid spacings, the interaction between the spheres becomes more abrupt. For dx = 1.0×10^{-1} m, the contact does not happen at a sharp point in time. Rather the spheres repel from each other in a smooth way, without geometrical contact.

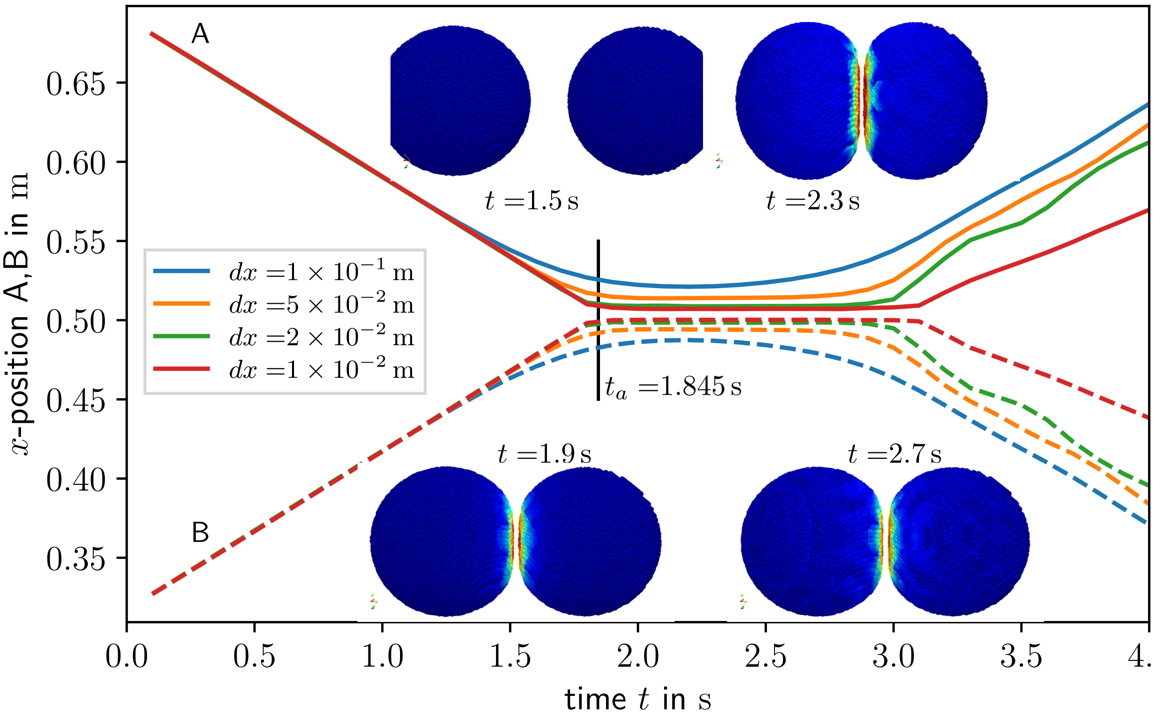


Figure 9.3: Analysis of the movement of control points A and B for varying grid spacings. Closeup visualizations of the spheres in contact are given for meaningful time-steps, the analytical collision time t_a is highlighted as well.

For the smallest conducted grid spacing dx = 1.0×10^{-2} m, the contact does follow a more sharp pattern and the spheres almost geometrically touch. Also, considering the analytical time of contact, the results indicate that it can potentially be reached as a limit case for infinitely small grid spacings. Nevertheless, the post-contact behavior shows a very different pattern for the considered spacings. The point of separation tends to become later in the simulation with smaller cells. In figure 9.4, the linear momentum of the upper sphere is analyzed over time. Interestingly, the smallest conducted grid spacing shows a huge drop of the linear momentum after the collision. Possible explanations are that with a finer CBG, higher vibration modes could be resolved and hence consume energy in the post collision phase.

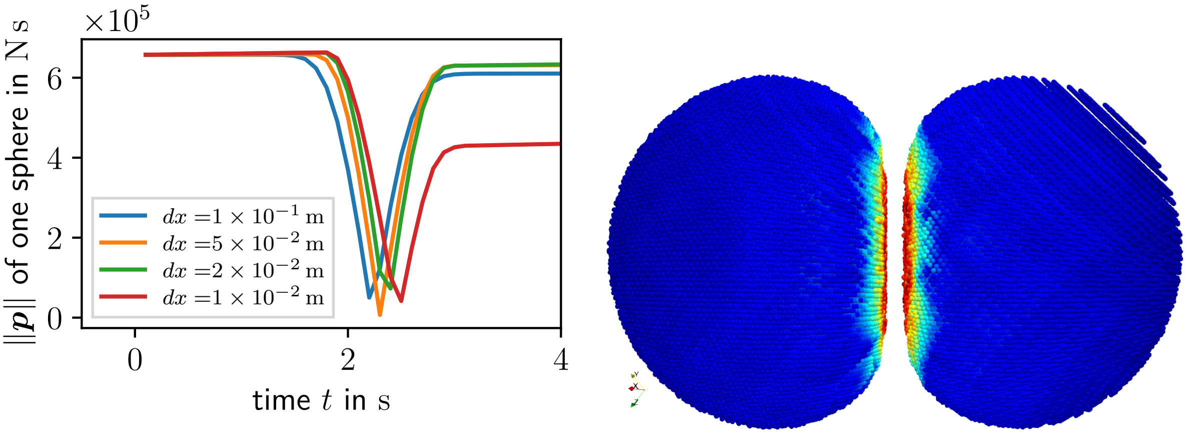


Figure 9.4: Norm of linear momentum $\|\boldsymbol{p}\|$ for the upper sphere over time (left) and closeup of the colliding spheres problem with dx $= 5 \times 10^{-2}$ m (right).

However, these vibrations are additionally triggered by a spurious adhesion the spheres exert on each other during the contact phase. Due to the shared force field on the CBG, the spheres can be seen as a single body in the contact phase. This phenomenon is one of the critical drawbacks of the standard contact in MPM. The observed case is a straight consequence of the perfect stick condition that this contact mechanism establishes.

It can be concluded that the build-in contact of MPM provides a very efficient way to analyze contact problems without further implementation efforts. However, it always establishes a perfect stick condition between contact partners. This behavior restricts its application to a limited set of problems.

9.2 Plastic impact problem

As one problem that can be examined with the MPM standard contact, an impact problem is considered. This benchmark was also considered, e.g., in Sulsky et al. [143] and Coetzee [30]. Experiments where conducted in Trucano and Grady [147]. The two dimensional problem is approximated using a plane strain condition. An AISI 52-100 steel bullet at a high initial velocity penetrates a 6061-T6 aluminum target. In reference to Coetzee [30], the bullet does not deform plastically, and the aluminum follows a perfect elasto-plastic von Mises constitutive behavior. Material parameters for the used hyperelastic material and the von Mises plasticity are given in tables 9.1 and 9.3, respectively. A graphical illustration of the boundary value problem is given in figure 9.5, alongside the problem dimensions and other important simulation parameters.

The boundary of the aluminum target is subject to standard rolling boundary conditions. The disk is always discretized always by 2836 material points, following a regular pat-

E	$= 200 \times 10^{9}\,\mathrm{Pa}$	ν	$= 0.3$	ρ	$= 7850\,\mathrm{kg\,m^{-3}}$

Table 9.1: AISI 52-100 chromium steel (hyperelasticity, see chapter 4.2).

tern, to provide a reasonable approximation of the circular geometry. For the following sensitivity study, the CBG and material point discretization of the aluminum target is parameterized by the parameter n. It scales the uniform grid spacing used for the CBG as given in table 9.2. The material points for the aluminum target are placed based on a Gauss-Legendre integration point pattern based on the CBG. This results in a maximum number of four material points per cell of the CBG, initially.

n	dx	total NoMP
50	1.2×10^{-3}	9436
80	7.5×10^{-4}	19 796
150	4.0×10^{-4}	62 836
200	3.0×10^{-4}	109 236

Table 9.2: Sensitivity parameter n and its influence for uniform grid spacing dx and total number of material points NoMP for the plastic impact problem.

$\boldsymbol{x}^c$	$= (30.0,\ 50.0,\ \mathrm{dx}/2)^T\ 10^{-3}\,\mathrm{m}$
r	$= 4.765 \times 10^{-3}\,\mathrm{m}$
$\boldsymbol{v}$	$= (0.0,\ 1160.0,\ 0.0)^T\ \mathrm{m\,s^{-1}}$
l	$= 60.0 \times 10^{-3}\,\mathrm{m}$
d	$= 40.0 \times 10^{-3}\,\mathrm{m}$
Δt	$= 1.0 \times 10^{-8}\,\mathrm{s}$ or $1.0 \times 10^{-9}\,\mathrm{s}$
t_{end}	$= 80.0 \times 10^{-6}\,\mathrm{s}$

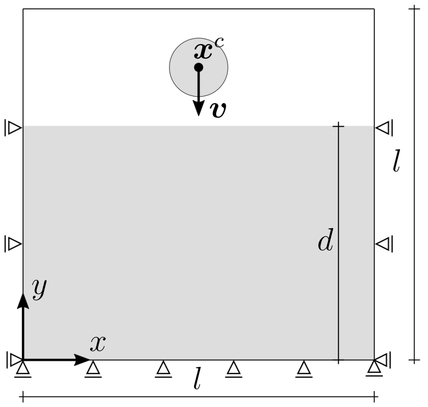


Figure 9.5: Simulation parameter (left) and illustration of the plastic impact boundary value problem.

Due to its initial velocity, the bullet penetrates the target during the analysis. As a contact mechanism, the standard MPM contact is used. Snapshots of the simulation are displayed in figure 9.3. The left side of each snapshot shows the linear momentum on the CBG, for once to highlight the shared momentum field to understand the contact mechanics, but also to visualize the vanishing momentum during the simulation. The energy dissipation is caused only by plastic deformation only, and no numerical damping is applied.

E	$= 78.2 \times 10^{9}\,\mathrm{Pa}$	ν	$= 0.3$	ρ	$= 2700\,\mathrm{kg\,m^{-3}}$		
τ_0	$= 300 \times 10^{6}\,\mathrm{Pa}$	τ_∞	$= 10 \times 10^{10}\,\mathrm{Pa}$	η	$= 0.0$	h	$= 0.9\,\mathrm{Pa}$

Table 9.3: 6061-T6 aluminum (perfect elasto-plastic, see chapter 4.3.7).

As the bullet gets in contact with the target, it transfers momentum. The stresses arising from the momentum gradients inside the target lead to plastic deformation, dissipating

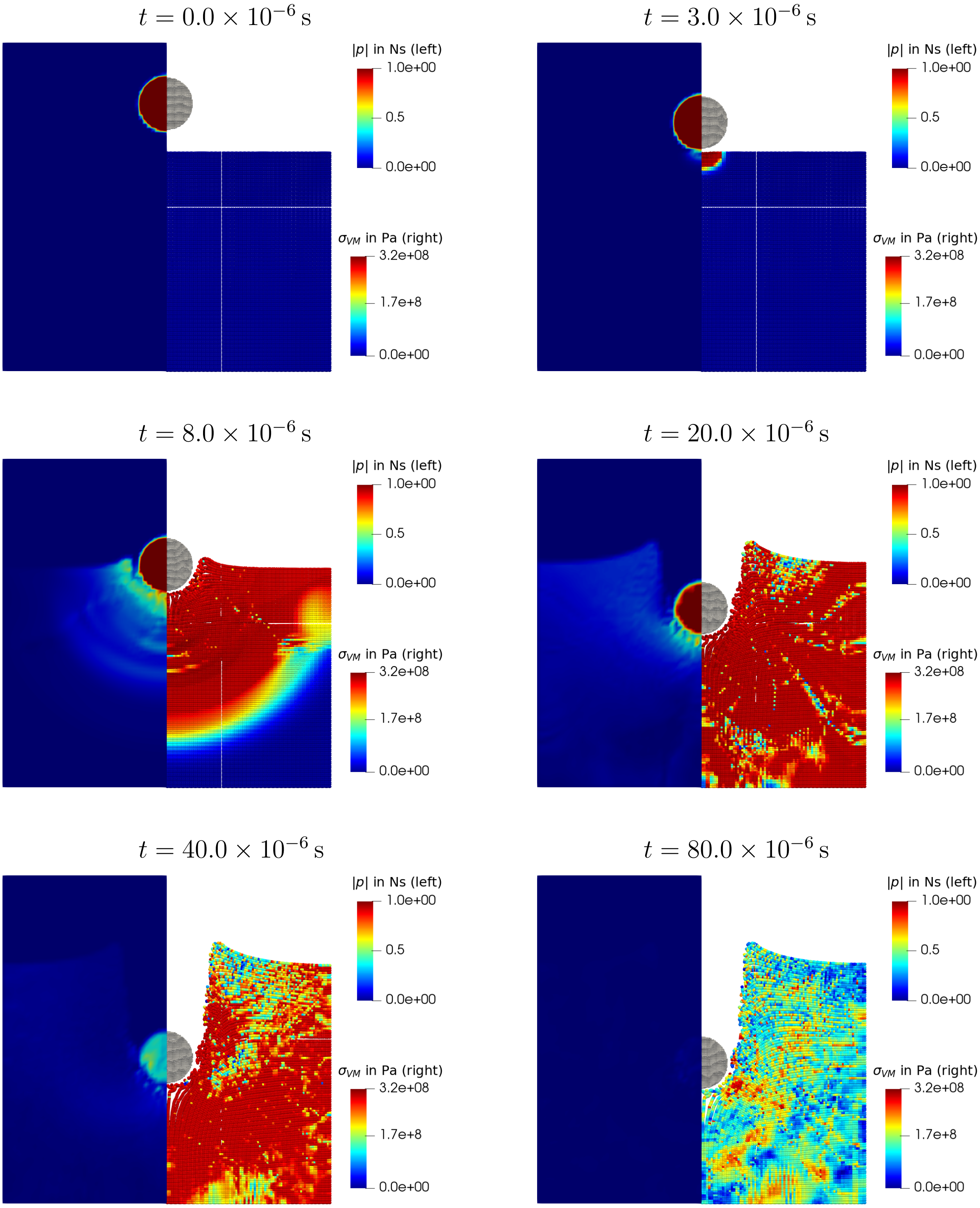


Figure 9.6: Time series of the impact problem for $n = 80$. Each snapshot shows the linear momentum on the CBG (left) and the material points (right). The material point view of the aluminum target features a von Mises stress plot.

energy. The snapshot at $t = 8.0 \times 10^{-6}$ s nicely shows the stress wavefront as it propagates through the material. At $t = 20.0 \times 10^{-6}$ s, the stress pattern is more diffused, as the stress waves, partly reflected from the boundaries of the computational domain, superimpose. Due to the penetration of the bullet, the material faces a discontinuity, forming the bullet hole. Notice that this behavior does not need any specific extensions to the MPM. Rather, material discontinuities are just possible with this method. The deformations leading to this behavior are extreme, highlighting the capabilities of the MPM in problems like this. The simulation is aimed to last until the bullet comes to rest, to conclude the final penetration depth. Sulsky et al. [143] and Coetzee [30] ran their simulations up to 40.0×10^{-6} s. In this thesis, snapshot $t = 40.0 \times 10^{-6}$ s still shows a decent amount of stresses and linear momentum; hence the simulation is continued until $t = 80.0 \times 10^{-6}$ s. At this final state, the overall stress magnitudes are much lower and the linear momentum left in the system is neglect-able.

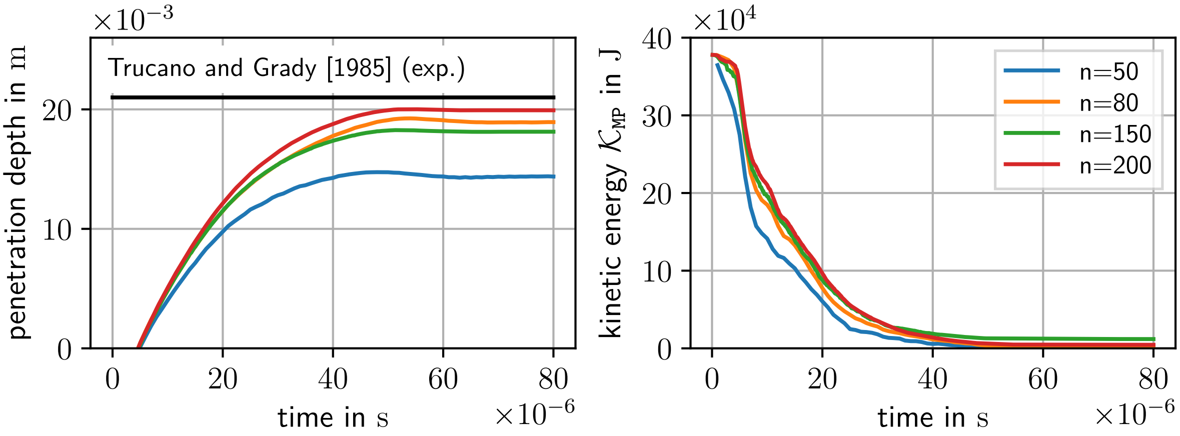


Figure 9.7: Penetration depth of the bullet (left) and total kinetic energy of the bullet (right) over time.

For a more analytical view of the simulation results, figure 9.7 shows the penetration depth and kinetic energy of the bullets during the simulation time for the analyzed discretizations. It can be concluded that after $t = 40.0 \times 10^{-6}$ s, the penetration depth does not change anymore, and also, the kinetic energy of the bullet is almost transported into the target. For a better comparison, the reported experimental results from Trucano and Grady [147] are displayed as well. Given the model approximations, e.g., of plane strain and perfect elasto-plasticity, this ELSE implementation of the MPM is capable of replicating this results.

At last, figure 9.8 shows the final configuration of the impact problem for the analyzed discretizations to visualize the morphology of the impact hole. It is well visible that for more dense discretizations, the rim of the impact crater shows more distinct features. The accumulated plastic arc length is given on the visualized material points. The observed values underline the huge deformations in this problem. Although not directly visible from the plots, there is no material point in the aluminum target which has not deformed inelastically. Nevertheless, the highest plastic deformations occur on the inner wall of the penetration wall. It can be observed that with finer discretizations, the deformation accumulates more locally.

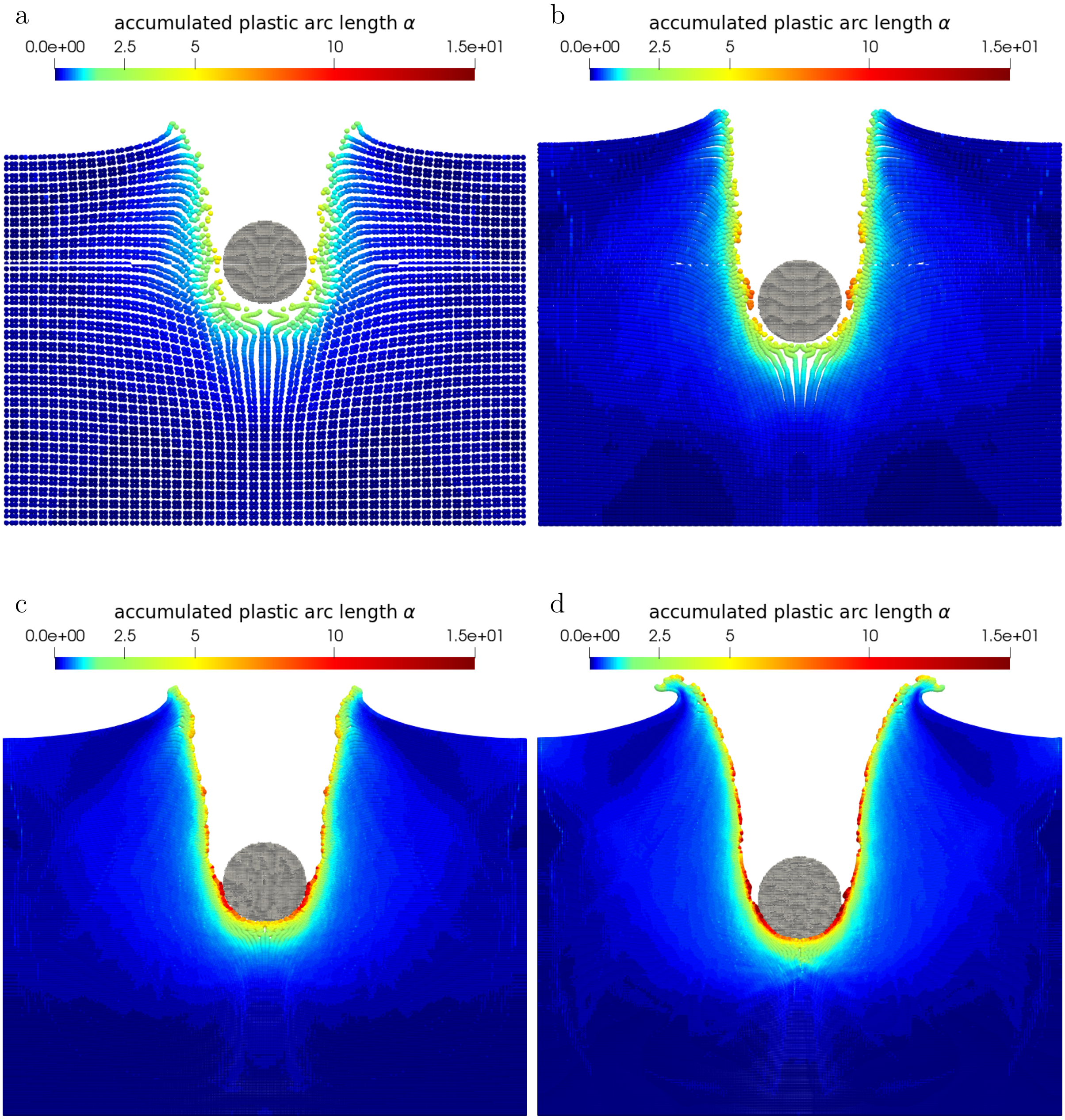


Figure 9.8: Final placements at the end of the impact simulation for discretizations a→ $n = 50$, b→ $n = 80$, c→ $n = 150$ and d→ $n = 200$.

9.3 MPM to rigid body penalty contact

In the scope of this thesis, metal processing applications are investigated. In these applications, a metallic workpiece is treated with tools which are much more durable than the processed material. Often it is possible to treat these tools as rigid in numerical analysis. With this focus, a penalty contact between MPM-Bodies and rigid bodies is discussed. The concepts and algorithms are closely related to the algorithms found in Ding and Schroeder [37], who used penalty forces in MPM before. In contrast to the standard contact between MPM-Bodies, we use contact mechanics offering frictional contact of the Coulomb type. The non-penetration condition further is satisfied using a penalty force. A penalty force may be used in contact mechanics to penalize a penetration of one body into the other. In the scope of this thesis, we define penalty force for each material point that penetrates a rigid body, defined by rigid surfaces. Notice that the penalty force is added to the material points body force. Accordingly, the material points force contribution to the grid is extended by the additional term

$$\boldsymbol{f}^{\mathtt{I}}_{\text{pen}} = \mathbb{N}^{\mathtt{I}}(\boldsymbol{x})\,\boldsymbol{f}^{\text{pen}}_{\mathtt{MP}}, \tag{9.2}$$

which is added to the contributions from equation (7.12). The penetration is compensated by the global explicit time integration due to the penalty force.

9.3.1 MPM to rigid body contact implementation

In the `ELSE` code, a rigid body is represented by the MPMRigidBody class. The derived classes contain the geometry description and algorithms to compute the penalty contact forces for each material point that penetrates this geometry. A penetration must be detected to do so. For efficiency reasons, a hierarchical contact search is implemented in `ELSE`. On the coarsest level, the global search is executed for a defined frequency. This frequency spans multiple time-steps. Currently implemented MPMRigidBody classes, such as the MPMRigidSurface used in the next example, provide an InProximity function which is called on each material point. This function provides a fast, coarse check whether a chance for penetration is given at all. This can be done, e.g., using bounding boxes or spheres for the rigid body. In every time-step, the local contact search is executed. Each potential penetration material point is now checked for actual penetration. If this is the case, we store a specific class to describe an active contact, which keeps track especially of the attachment point, as discussed later. The local contact search might also detect that a contact pair does not be in actual contact anymore, resulting in a deletion of the concerning class. This algorithmic setup does not introduce much overhead using suitable frequencies for the global search and reasonable InProximity functions.

9.3.2 Penalty force and Coulomb friction

The penalty forces that are used to handle the contact for the problems in this thesis establish a Coulomb friction condition. Specifically, as a material point penetrates a rigid body, it experiences a penalty force $\boldsymbol{f}^{\text{pen}}_{\mathtt{MP}}$ which satisfies the Coulomb friction cone

$$\boldsymbol{f}^{\text{pen}}_{\mathtt{MP}} \cdot \boldsymbol{t} \leq \mu\, \boldsymbol{f}^{\text{pen}}_{\mathtt{MP}} \cdot \boldsymbol{n}, \tag{9.3}$$

with the normal vector $\boldsymbol{n}$ and tangential vector $\boldsymbol{t}$, defined on the penetrated surface, respectively. The parameter μ denotes the dimensionless friction coefficient. The penalty force vector

$$\boldsymbol{f}^{\text{pen}}_{\mathtt{MP}} = \kappa_{\text{pen}}\,\left(\boldsymbol{x}_A - \boldsymbol{x}_{\mathtt{MP}}\right) \tag{9.4}$$

is computed using the attachment point $\boldsymbol{x}_A$ and the penalty parameter κ_{pen} in $\text{kg}\,\text{s}^{-2}$. The penalty parameter is an artificial parameter, which needs to be set problem-specific. The attachment point can be interpreted as an anchor point for the penetrated material point on the surface of the rigid body, see figure 9.9. It is initialized when the penetration is first detected, with the closest point projection $\boldsymbol{x}_C$ of the material point onto the surface of the rigid body. As indicated by equation (9.4), the penalty force drags the material point towards the attachment points in any case. The attachment point is not fixed on the surface, however. If it would be, the friction cone could potentially be violated. This case is detected in the algorithm by first computing a trail force $(\boldsymbol{f}_{tr}^{\text{pen}})_{\mathtt{MP}}$, according to the current position of the attachment point. If this trail force does not satisfy the friction cone, the attachment point is relaxed, i.e., it moves a minimum distance to satisfy equation (9.4) again. The process defines a minimization problem, which is linear in the case of planar surfaces. In the following, the relaxation of the attachment point is derived based on the definitions

$$\boldsymbol{n} = \frac{\boldsymbol{x}_C - \boldsymbol{x}_{\mathtt{MP}}}{\|\boldsymbol{x}_C - \boldsymbol{x}_{\mathtt{MP}}\|}, \quad \text{and} \quad \boldsymbol{t} = \frac{\boldsymbol{x}_A - \boldsymbol{x}_C}{\|\boldsymbol{x}_A - \boldsymbol{x}_C\|}. \tag{9.5}$$

Notice that these definitions are only valid in the case of a violated friction cone, which necessarily implies $\boldsymbol{x}_A \neq \boldsymbol{x}_C$. For the relaxation of the attachment point, we introduce a temporary parameter $\hat{\xi} \in (0, 1]$ to describe its displacement $\boldsymbol{u}_A$, with

$$\boldsymbol{u}_A = \hat{\xi}\,(\boldsymbol{x}_C - \boldsymbol{x}_A)\,. \tag{9.6}$$

This way, the attachment point always relaxes towards the closest point projection. In the frictionless case ($\mu = 0$), it necessarily relaxes to exactly this point ,i.e., $\boldsymbol{u}_A = (\boldsymbol{x}_C - \boldsymbol{x}_A)$. In the general case, the linear solution

$$\hat{\xi} = -\frac{(\boldsymbol{x}_A - \boldsymbol{x}_{\mathtt{MP}}) \cdot (\boldsymbol{t} - \mu\,\boldsymbol{n})}{(\boldsymbol{x}_C - \boldsymbol{x}_A) \cdot (\boldsymbol{t} - \mu\,\boldsymbol{n})} \tag{9.7}$$

is taken in $\boldsymbol{f}_{\mathtt{MP}}^{\text{pen}} = \kappa_{\text{pen}}\,(\boldsymbol{x}_A + \boldsymbol{u}_A - \boldsymbol{x}_{\mathtt{MP}})$ which does fulfill equation (9.4). The algorithmic procedure for computation of the penalty force and a potential relaxation of the attachment point is given in algorithm 13.

The algorithm is illustrated in figure 9.9. Time-step a.) is where the penetration of the material point at $\boldsymbol{x}_{\mathtt{MP}}$ is detected first. An attachment points $\boldsymbol{x}_A$ is initialized at the closest point projection $\boldsymbol{x}_C$, and a contact penalty force is computed. In the next time-step b.) the penetration depth is reduced due to the force. However, the trial penalty force, does not satisfy the friction cone, i.e., is outside the red area. The attachment point is relaxed to the point where the red friction cone meets the contact surface, and the final penalty force is computed.

9.4 Libra benchmark

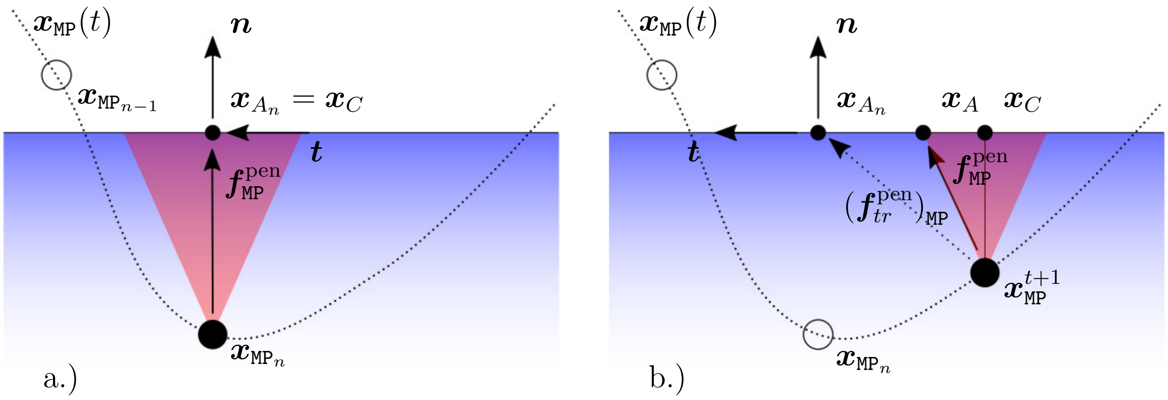
As a standard benchmark to the penalty contact, a virtual libra is considered. A cube-shaped elastic body is discretized by material points and set to rest on a rigid surface. With vanishing friction coefficient $\mu = 0$, this benchmark is used to investigate the non-penetration condition and the normal forces between the surface and the body. The cube, with an edge length of 1.0 m, has relatively low stiffness with Young's modulus $E = 1.0 \times 10^3\,\text{Pa}$ and Poisson's ratio $\nu = 0.45$. The total force on the rigid surface can be computed as

$$f_z = 1.0 \times 10^2\,\text{kg}\,\text{m}^{-3}\,{-10.0}\,\text{m}\,\text{s}^{-2}\,1\,\text{m}^3 = -1 \times 10^3\,\text{N} \tag{9.8}$$

with a material density of $\rho = 1.0 \times 10^2\,\text{kg}\,\text{m}^{-3}$ and consideration of gravity $\mathfrak{b} = (0.0, 0.0, -10.0)^T\text{m}\,\text{s}^{-2}$. The simulation is executed for $t_{\text{end}} = 3.0\,\text{s}$ with $\Delta t = 1.0 \times 10^{-3}\,\text{s}$.

```
Input:
  $\boldsymbol{x}_{\texttt{MP}}$ material point position
  $\boldsymbol{x}_C$ closest point projection
  $\boldsymbol{x}_A$ attachment point
Output:
  $\boldsymbol{f}_{\texttt{MP}}^{\text{pen}}$ contact penalty force
  $\boldsymbol{x}_A$ attachment point
/* compute trial penalty force                                */
$(\boldsymbol{f}_{tr}^{\text{pen}})_{\texttt{MP}} = \kappa_{\text{pen}}\,(\boldsymbol{x}_A - \boldsymbol{x}_{\texttt{MP}})$
/* check initial contact case                                 */
if $\boldsymbol{x}_A == \boldsymbol{x}_C$ then
    return $(\boldsymbol{f}_{tr}^{pen})_{MP}$, $\boldsymbol{x}_A$
end
/* compute surface vectors                                    */
$\boldsymbol{n} = (\boldsymbol{x}_C - \boldsymbol{x}_{\texttt{MP}})\,/\,(\|\boldsymbol{x}_C - \boldsymbol{x}_{\texttt{MP}}\|)$
$\boldsymbol{t} = (\boldsymbol{x}_A - \boldsymbol{x}_C)\,/\,(\|\boldsymbol{x}_A - \boldsymbol{x}_C\|)$
/* check friction cone                                        */
if $(\boldsymbol{f}_{tr}^{pen})_{MP} \cdot \boldsymbol{t} > \mu\,(\boldsymbol{f}_{tr}^{pen})_{MP} \cdot \boldsymbol{n}$ then
    /* relax attachment point                                 */
    $\hat{\xi} = -\dfrac{(\boldsymbol{x}_A - \boldsymbol{x}_{\texttt{MP}}) \cdot (\boldsymbol{t} - \mu\,\boldsymbol{n})}{(\boldsymbol{x}_C - \boldsymbol{x}_A) \cdot (\boldsymbol{t} - \mu\,\boldsymbol{n})}$
    $\boldsymbol{x}_A \leftarrow \boldsymbol{x}_A + \hat{\xi}\,(\boldsymbol{x}_C - \boldsymbol{x}_A)$
    $\boldsymbol{f}_{\texttt{MP}}^{\text{pen}} = \kappa_{\text{pen}}\,(\boldsymbol{x}_A - \boldsymbol{x}_{\texttt{MP}})$
end
return $\boldsymbol{f}_{MP}^{pen}$, $\boldsymbol{x}_A$
```

Algorithm 13: Computation of penalty contact force for a material point that penetrates a rigid body.



$\boldsymbol{x}_{\mathtt{MP}}(t)$
$\boldsymbol{n}$
$\boldsymbol{x}_{\mathtt{MP}_{n-1}}$
$\boldsymbol{x}_{A_n} = \boldsymbol{x}_C$
$\boldsymbol{t}$
$\boldsymbol{f}^{\text{pen}}_{\mathtt{MP}}$
$\boldsymbol{x}_{\mathtt{MP}_n}$
a.)
$\boldsymbol{x}_{\mathtt{MP}}(t)$
$\boldsymbol{n}$
$\boldsymbol{x}_{A_n}$
$\boldsymbol{x}_A$
$\boldsymbol{x}_C$
$\boldsymbol{t}$
$\boldsymbol{f}^{\text{pen}}_{\mathtt{MP}}$
$(\boldsymbol{f}^{\text{pen}}_{tr})_{\mathtt{MP}}$
$\boldsymbol{x}^{t+1}_{\mathtt{MP}}$
$\boldsymbol{x}_{\mathtt{MP}_n}$
b.)

Figure 9.9: Illustration of a contact situation at two subsequent time-steps. A material point following the path $\boldsymbol{x}_{\mathtt{MP}}(t)$ penetrates a rigid body, denoted by the surface with normal vector $\boldsymbol{n}$. The Coulomb friction cone for the penalty forces of the material point is depicted in red, respectively.

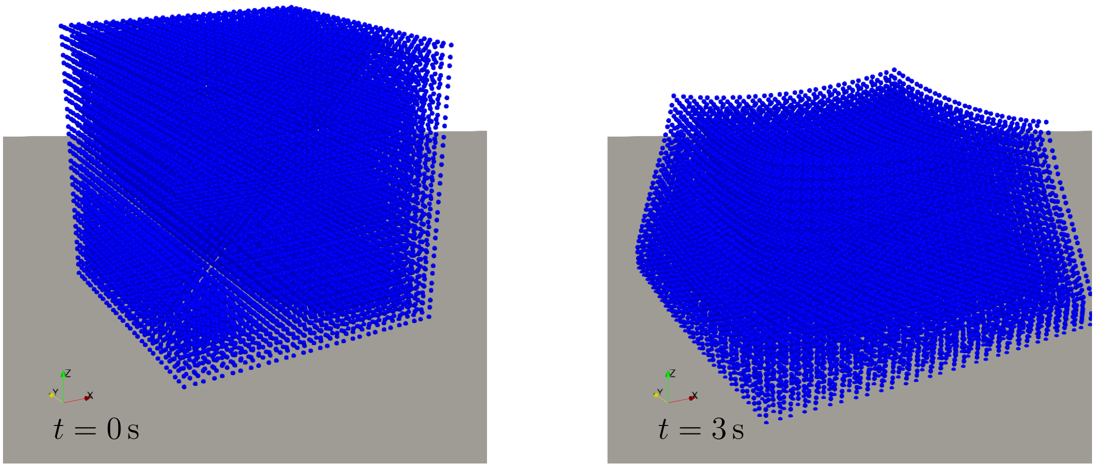


Figure 9.10: Libra problem, at the initial and final configuration. A soft elastic cube is set onto a rigid surface (gray) due to gravity. The cube is discretized by 27×10^3 material points.

Numerical damping $\alpha_d = 2.0 \times 10^1\,\mathrm{s}^{-1}$ dissipates energy from the system, as the cube is initially not exactly positioned on the surface, due to technical reasons. The set of $n \in [1, 2, 3, 4, 5, 6]$ is considered, to investigate the effects of varying penalty parameters $\kappa_{\text{pen}} = 10^n \mathrm{kg\,s}^{-2}$. The first and final states of the simulation are illustrated in figure 9.10. The cube is relatively soft and deforms on its self-weight. The results are given in figure 9.11. Smaller values of the penalty parameter, i.e., values $n = 1$ and $n = 2$, results in considerable penetration of $\approx 5\%$ and $\approx 1\%$. On the contrary $n = 6$ leads to an initial rebound of the body from the surface, and eventually, the simulation aborts. This study supports the general behavior that penalty parameters should be set as high as necessary but as low as possible. The total contact force the surface is subject to shows a similar pattern. Higher values of κ_{pen} lead to an unsteady force pattern or spurious higher force values, while all lower penalty values converge to the correct value with time. It can be concluded that an advisable range for the penalty parameter is the range of the stiffness of the contact bodies.

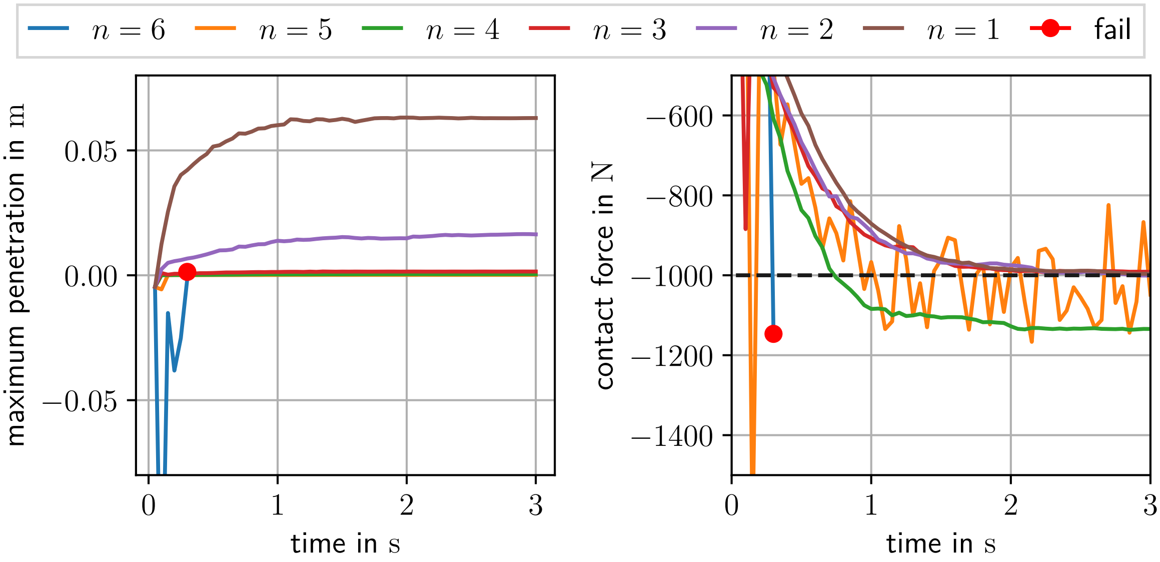


Figure 9.11: Libra results for $\kappa_{\text{pen}} = 10^n \text{kg}\,\text{s}^{-2}$ Maximum penetration of any material point into the surface over time (left) and total contact force amplitude acting on the surface.

9.5 Inclined plate benchmark

For benchmarking the frictional Coulomb contact implementation, a classical setup is considered. A compact body is set on a plate, which rotates to an angle. The body may stick to the plate or fall off of it, based on its friction coefficient relative to the plate.

The contact mechanics from earlier are applied to convex bodies. A convex body is represented by a triangulated surface mesh. The penalty force is computed for the current triangle of the surface mesh with the material point in contact. For a better understanding, the triangulation of the plate is chosen unnecessarily dense and illustrated in figure 9.16. The plate is assumed a rigid body with prescribed motion. It features a thickness of 0.1 m which is not relevant to this problem. While various friction coefficients are used $\mu \in [0.1, 0.2, 0.4, 1.0, 1.8]$, a constant penalty parameter of $\kappa_{\text{pen}} = 10^3 \text{kg}\,\text{s}^{-2}$ is chosen.

The cube with a uniform edge length of 0.5 m is made from a hyperelastic material with Young's modulus $E = 1 \times 10^4$ Pa, Poisson's ratio $\nu = 0.3$, and a material density of $\rho = 100\,\text{kg}\,\text{m}^{-3}$. Gravity is considered by $\mathfrak{b} = (0.0, 0.0, -10.0)^T \text{m}\,\text{s}^{-2}$. Its numerical discretization features 1000 material points. The CBG is fixed, with a uniform grid size of 0.2 m. In order to observe the relative movement of the cube on the surface of the plate, a control point A is defined in the center of the bottom face of the cube. The simulation is carried out for 7.0 s with a constant time-step size of $\Delta t = 1.0 \times 10^{-3}$ s, with slight numerical damping of $\alpha_d = 2.0 \times 10^1 \,\text{s}^{-1}$. During the dynamic analysis, the plate rotates around the $x-$axis, as shown in figure 9.12. The rotation angle is depicted with $\gamma \in [10°, 45°, 60°]$ and is reached within 2.0 s. Other than its rotation, the plate remains at its position during the analysis.

For an idealized version of this problem, we can compute an approximation, of whether the contact should result in sliding or stick. To prevent a sliding contact, the estimate is given with

$$\mu \geq \tan(\gamma). \tag{9.9}$$

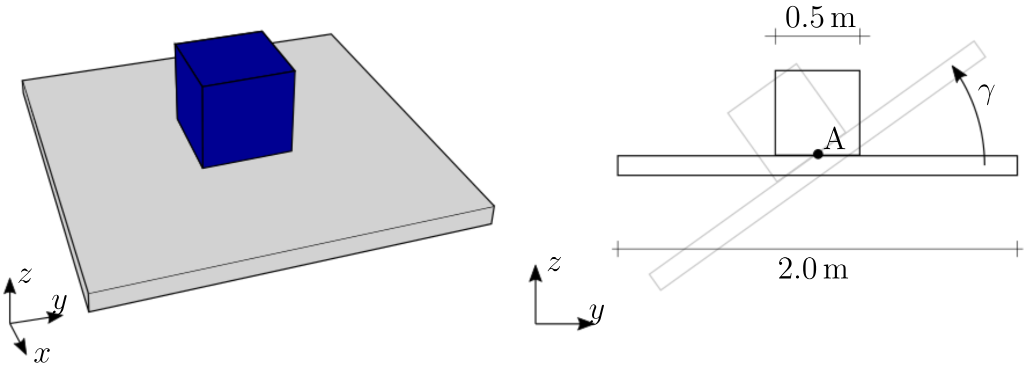


Figure 9.12: Visualization of the inclined plate benchmark problem in 3D (left). The plate rotates around the x-axis about a final angle of γ as shown by the 2d cut on the right. The control point A, defined on the bottom face of the cube is observed.

The first analysis is conducted with an angle of $\gamma = 10°$. Following equation (9.9), only the lowest considered friction would lead to a sliding contact condition. In figure 9.13, results for this case are shown. It can be seen that the vertical position of the control point at $\boldsymbol{x}^A$ diverges for $\mu = 0.1$ from all other graphs after 10° is reached. While the cube rests on the inclined plate for all higher values, just as expected, the lowest friction graph describes a linear slope, describing the sliding of the cube on the plate at a constant velocity. Additionally, in figure 9.13, an analysis of the contact force components is made. The total force on the rigid plate is projected to the components relative to its upper surface, and the ratio of tangential and normal components is given. As intended, the graph for $\mu = 0.1$ maximizes to this value under sliding conditions. For the other cases where stick contact is observed, the graph maximizes to $\tan(\gamma)$, which is covered by the estimate in equation (9.9).

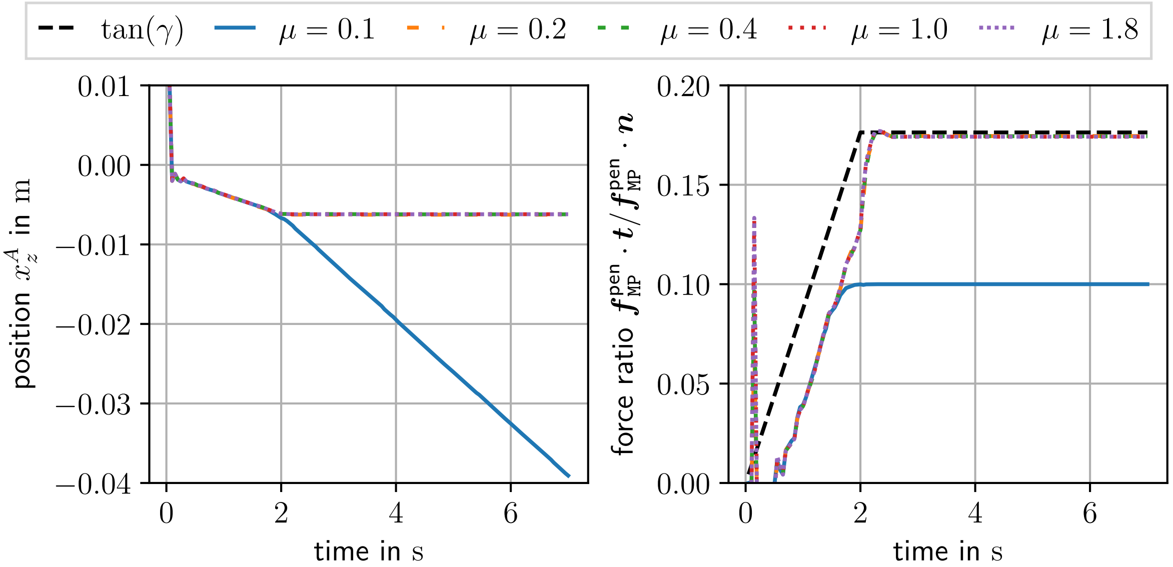


Figure 9.13: Vertical movement of the control point A at the bottom of the cube (left) and ratio of tangential and normal contact forces on the plate (right), for a final angle of $\gamma = 10°$. The case $\mu = 0.1$ shows sliding contact conditions, for all other friction coefficients the cube sticks to its initial position on the plate.

In the second case, the plate is inclined up to 45°, which should result in stick conditions for the two cases $\mu = 1.0$ and $\mu = 1.8$. The position graphs in figure 9.14 show exactly that behavior. Nevertheless, two interesting aspects can be observed. The slope of the position graph decreases, with increasing friction coefficient. This behavior indicates different sliding velocities. Another observation is that the stick cases do not show a straight graph, as observed for stick cases under 10° inclination. The reason for this can be observed in the visualization of the simulation. The contact condition is a stick condition, but the cube begins to roll on the plate. The rolling can be explained by the angular momentum which is introduced to the cube due to the dynamic inclination, and by considering the center of mass of the cube. Figure 9.16 shows snapshots of a simulation under $\gamma = 60°$, which are the same modes of rolling. The motion plots for point A in this case are given in figure 9.15, respectively. Following the same logic, all sliding cases show higher sliding velocities. Although, according to equation (9.9), only $\mu = 1.8$ should show a stick contact behavior for that angle, the rolling mode can also be observed for $\mu = 1.0$. Again, this can be explained by the dynamic effects, and the simplification made for the estimation.

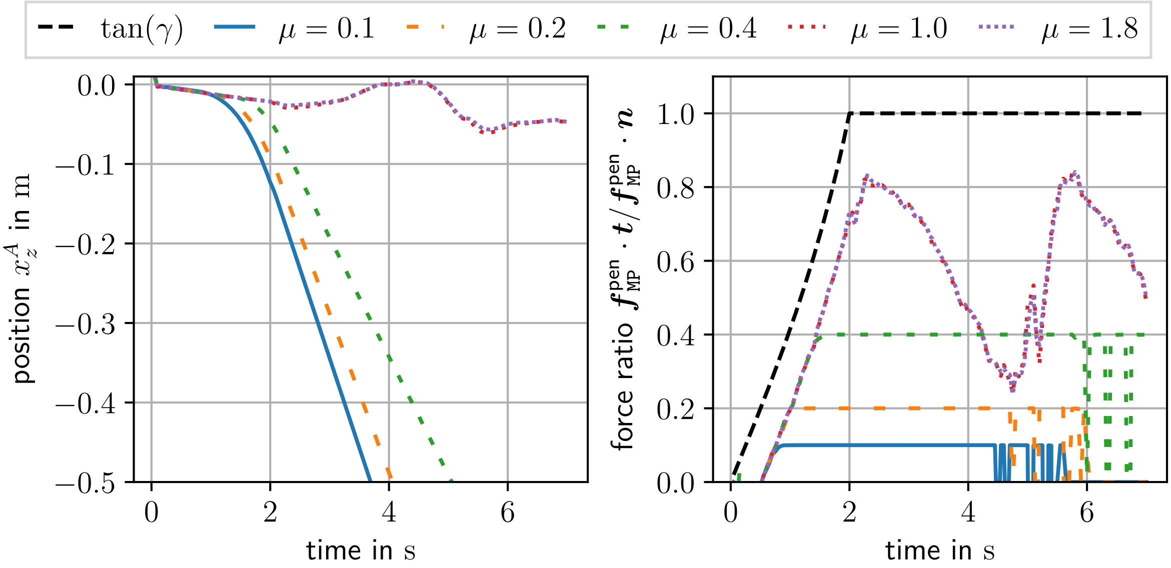


Figure 9.14: Vertical movement of the control point A at the bottom of the cube (left) and ratio of the tangential and normal contact forces on the plate (right), for a final angle of $\gamma = 45°$. A rolling mode of the cube can be observed for $\mu = 1.0$ and $\mu = 1.8$, for all other coefficients the cube slides with different velocities.

It can be seen that for all sliding cases, the ratio saturates to the value of the friction coefficient, comparing the force ratios depicted in figures 9.14 an 9.15. However, it never exceeds this, proving the correct implementation of Coulomb's friction law. Interestingly, the rolling cases force ratios do not saturate in the 45° case. For a final plate inclination of 60°, only $\mu = 1.0$ variant saturates at certain points. An abrupt change in the force ratios graphs in the later phase of the simulation indicates the cube reaching the end of the plate. A short oscillation phase depicts the cubes contact with its edge. After the oscillation phase, the ratio goes to zero. A zero ratio means no contact force is present anymore, i.e. the cube has fallen off from the plate. This event happens earlier for lower friction cases, as their sliding speed is relatively higher.

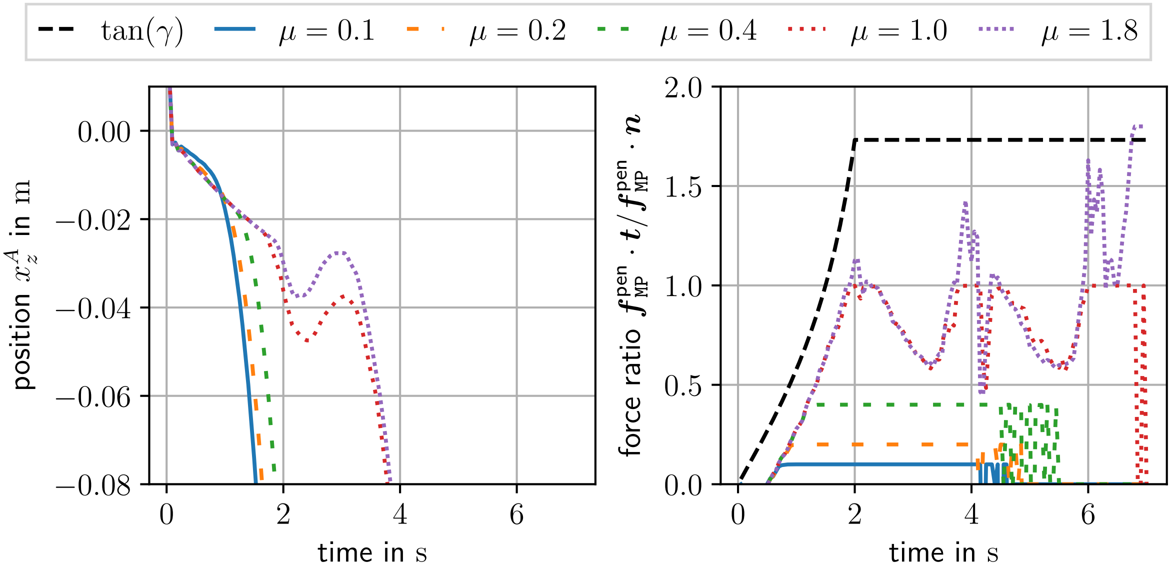
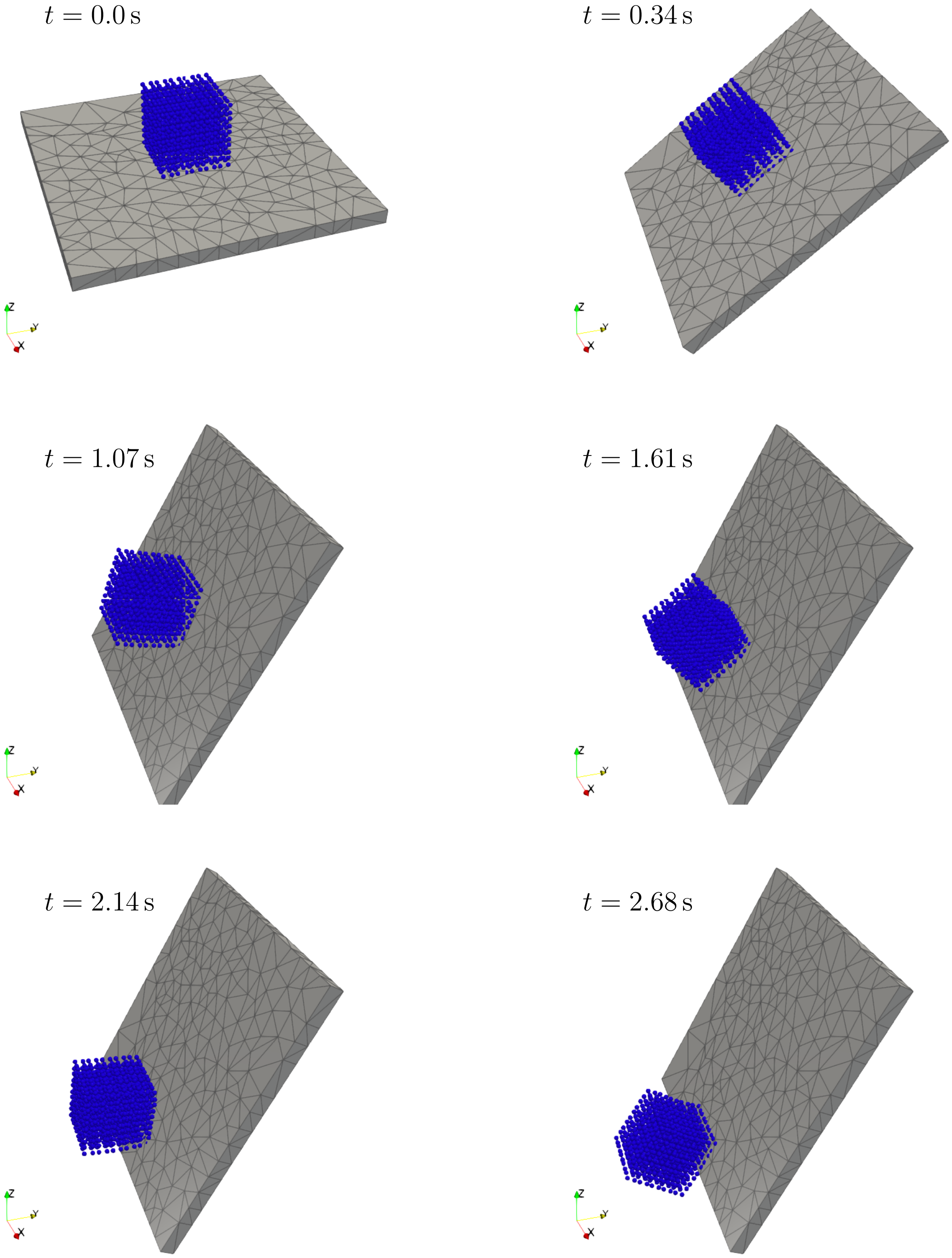


Figure 9.15: Vertical movement of the control point A at the bottom of the cube (left) and ratio of tangential and normal contact forces on the plate (right), for a final angle of $\gamma = 60°$. Phenomenologically the same behavior as for $\gamma = 45°$ can be observed. The force ratio falls off a lot earlier than in the previous case, indicating the cube falling off from the plate.



$t = 0.0\,\mathrm{s}$

$t = 0.34\,\mathrm{s}$

$t = 1.07\,\mathrm{s}$

$t = 1.61\,\mathrm{s}$

$t = 2.14\,\mathrm{s}$

$t = 2.68\,\mathrm{s}$

Figure 9.16: Visualizations of the inclined plane simulation for $\mu = 1.8$ and $\gamma = 60°$. The friction coefficient result in a stick condition. Due to dynamic effects the cube enters a rolling mode until it falls off from the plate.

10 MPM Variants

The material point method offers a variety of modifications in various aspects. Using alternative mapping operations between a CBG and the material points is the most common one. Some of the most cited literature on MPM is located in this field, resulting in sub-variants of the MPM. In Bardenhagen and Kober [11], the authors introduced the generalized interpolation material point (GIMP) method. The base idea is to treat a material point with a finite extend rather than being a singular point. As a consequence, several aspects are improved, such as smoothness, convergence rates, cell crossing noise, and extension instability. The latter phenomenon is analyzed in this chapter. Following the GIMP ansatz, the shape functions used for mapping operations are generalized by

$$\mathbb{N}^{I}_{\mathtt{MP}} = \frac{1}{\mathsf{V}_{\mathtt{MP}}} \int_{\mathcal{B}_\chi} \chi_{\mathtt{MP}}(\boldsymbol{x} - \boldsymbol{x}_{\mathtt{MP}})\, S(\boldsymbol{x})\; \mathrm{d}\boldsymbol{x} \tag{10.1}$$

and

$$\mathbb{B}^{I}_{\mathtt{MP}} = \frac{1}{\mathsf{V}_{\mathtt{MP}}} \int_{\mathcal{B}_\chi} \chi_{\mathtt{MP}}(\boldsymbol{x} - \boldsymbol{x}_{\mathtt{MP}})\, \nabla S(\boldsymbol{x})\; \mathrm{d}\boldsymbol{x}, \tag{10.2}$$

representing an averaging scheme over the virtual domain $\mathcal{B}_\chi$ of the material points. Hereby, $S(\boldsymbol{x})$ denotes an interpolation function over this domain, and $\chi_{\mathtt{MP}}(\boldsymbol{x} - \boldsymbol{x}_{\mathtt{MP}})$ is a characteristic particle function. This representation, includes the standard MPM by introducing the characteristic particle function as the delta dirac function

$$\chi_{\mathtt{MP}}(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{x} = \boldsymbol{x}_{MP} \\ 0 & \text{else} \end{cases} . \tag{10.3}$$

This case was used in this thesis exclusively up to this point by using the linear Lagrange grid basis functions (compare appendix section E) directly $\mathbb{N}^{I}_{\mathtt{MP}} = N(\boldsymbol{x}_{\mathtt{MP}})$ and $\mathbb{B}^{\mathtt{I}}_{\mathtt{MP}} = \nabla N^{I}(\boldsymbol{x}_{\mathtt{MP}})$. Alternative forms the characteristic particle function and the interpolation function $S(\boldsymbol{x})$ are presented, e.g., in Bardenhagen and Kober [11] and Steffen et al. [136], to name just a view. A different sight of the same idea was taken by Sadeghirad et al. [119], where the extend of the material point is described literally by a finite domain. The main motivation was to apply the deformation a material point accumulates over time to its associated volume. This way, the material point domain deforms, which reflects in the mapping operations. It was identified that using parallelograms for these domains without updating their shapes covers the so-called uGIMP variant. Applying only first-order deformations in principle directions (neglecting shear deformation) leads to the cpGIMP variant. The convected particle domain interpolation (CPDI) method is defined by applying all first-order deformation to the parallelogram. Implementation-wise, these variants follow the same algorithm as the standard MPM. Additionally, each material point carries two base vectors to define the parallelogram (or parallelepiped in 3D) and updates them according to the deformation gradient at the end of a time-step. Effectively, the shape functions $\mathbb{N}$ are no longer just the grid basis functions but obtained via the generic expression

$$\begin{aligned} \mathbb{N}^{I}(\boldsymbol{x}_{\mathtt{MP}}) &= \frac{1}{\mathsf{V}_{\mathtt{MP}}} \sum_{j} \left(\int_{\mathcal{B}_{\mathtt{MP}}} Q^{j}_{\mathtt{MP}}(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} \right) N^{I}(\boldsymbol{x}^{j}_{\mathtt{MP}}), \\ \mathbb{B}^{I}(\boldsymbol{x}_{\mathtt{MP}}) &= \frac{1}{\mathsf{V}_{\mathtt{MP}}} \sum_{j} \left(\int_{\mathcal{B}_{\mathtt{MP}}} \operatorname{grad} Q^{j}_{\mathtt{MP}}(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} \right) N^{I}(\boldsymbol{x}^{j}_{\mathtt{MP}}). \end{aligned} \tag{10.4}$$

which are a special form of the GIMP equations (10.2). The here introduced set of shape functions $Q^j_{\mathtt{MP}}$ are defined on the finite material point domain for each of its vertices j. Notice that this way, a material point maps al l nodes of cells that enclose one of its vertices. However, the solution of the CBG is still mapped to the actual material point. The idea is extended further in Sadeghirad et al. [120], where CPDI2 updates the vertex positions based on the velocity field from the CBG. Hence, the domain is not restricted to provide parallel edges or surfaces. In Sadeghirad et al. [120], the authors also discussed advantages and disadvantages for CPDI and CPDI2 variants alike. An extension of the CPDI update strategy, but for particle domains of flexible geometric shapes is presented in Nguyen et al. [105]. An illustration of two-dimensional MPM variants with a focus on their domain is given in figure 10.1.

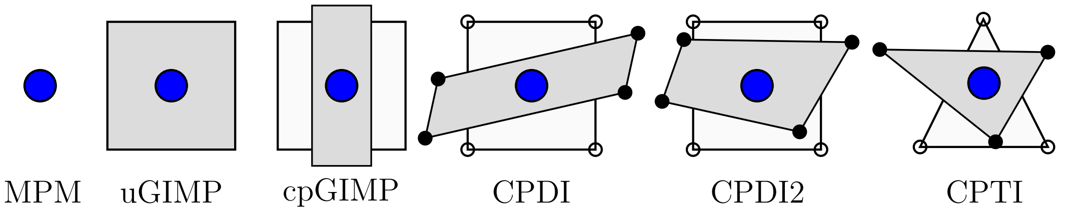


Figure 10.1: Illustration of MPM variants. In standard MPM, the material points associated volume is represented by a singularity. GIMP variants use implicit functional descriptions of the particles domain. In CPDI variants, the material points volume is represented by geometric shapes, defined by vertices. For some variants, the associated domain deforms according to the material points deformation state.

As a general note on CPDI, it is important to differentiate between CPDI1 and CPDI2. In this thesis, CPDI1 denotes any geometrical shape (including non-parallel shapes) but using the first-order deformation update from the actual material point. In a CPDI2 variant instead, the vertices of the domain are free to move with the velocity field of the CBG, resulting in unconstrained deformations.

Generally, a CPDI simulation requires considerable more memory (to store the vertex positions) and simulation time due to the higher number of mapping operations. This CPDI approach eliminates cell crossing noise, which usually occurs as a material point translates into the next computational cell. With CPDI, the gradient fields are continuous across cell boundaries; hence cell crossing noise is avoided.

The greatest advantage in using CPDI can be seen at tensile deformation. Here the standard MPM has the disadvantage that space between material points gets large. If two material points are separated by one computational cell, they do not share any nodes where both contribute. As a consequence they move independently, although representing the same body. This material discontinuity might be intentional, such as in cutting simulation, but is often unintended. In these cases, the use of CPDI reduces the initial material point density dramatically as material continuity is described by the vertex locations. The material points stay connected, even if separated by multiple cells if tensile deformation is large enough.

10.1 CPDI in ELSE

In the generally three-dimensional simulation code `ELSE`, two variants of CPDI are implemented. Both follow a CPDI1 update scheme. Besides a cpGIMP variant, which is associated here as a CPDI category, a CPDI scheme for hexahedral cells is available. Such a geometric cell was first used in Maassen et al. [94].

At initialization, each material point carries 8 vertices j, defining a hexahedral domain. In the `ELSE` implementation, the volume of the domain represents the material points associated volume. As a natural consequence, it is possible to use hexahedral meshes as an input to the MPM simulation. Consequently, a more accurate description of geometry, at least from a post-processing point of view is obtained.

The integral in equation (10.4) of the local shape function $Q^{\mathtt{j}}_{\mathtt{MP}}$ over the material points domain are computed numerically. They are material point-wise operations. The initial positions of the vertices $\boldsymbol{X}^{\mathtt{j}}$ are updated with the material points deformation gradient, as shown in figure 10.2. The gradient operator on the particle domain is computed by means of the chain rule as in the standard finite element procedure. It is worth mentioning that this approach is computationally more expensive as most CPDI variants provided in the literature, using analytically integrated expressions for the computation of the mapping functions. However, this approach is more flexible, and the overhead is neglect able with parallelization.

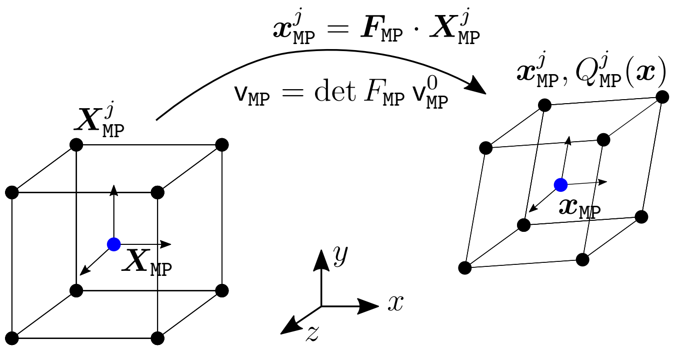

$\boldsymbol{x}^j_{\mathtt{MP}} = \boldsymbol{F}_{\mathtt{MP}} \cdot \boldsymbol{X}^j_{\mathtt{MP}}$
$\mathsf{v}_{\mathtt{MP}} = \det F_{\mathtt{MP}}\, \mathsf{v}^0_{\mathtt{MP}}$
$\boldsymbol{x}^j_{\mathtt{MP}}, Q^j_{\mathtt{MP}}(\boldsymbol{x})$
$\boldsymbol{X}^j_{\mathtt{MP}}$
$\boldsymbol{X}_{\mathtt{MP}}$
$\boldsymbol{x}_{\mathtt{MP}}$
y
x
z

Figure 10.2: CPDI1: 8-noded hexahedral particle domain deforms with the particles deformation gradient.

10.2 Truss benchmark

As previously discussed, the extension instability in standard MPM is very challenging in tensile deformation. In this boundary value problem, a truss is subject to a huge gravitational load. Hence, it exceeds large tensile deformation. The problem was introduced in Sadeghirad et al. [119] as a plane strain version to show the advantages of the CPDI variant. The exact problem is considered as a validation step to the implemented cpGIMP and CPDI variant in `ELSE`. Additional discretizations are considered, however, for a more detailed view on the problem. Sadeghirad et al. [119] considered a second version of the truss, including a larger load. This variant is adopted here as well but in a three-dimensional version.

The truss consists of a hyperelastic material with parameters given in figure 10.3. The gravitational load is introduced by $\mathbb{b} = (0.0, 0.0, g)^T \mathrm{m\,s^{-2}}$, with $g = 1 \times 10^3$ for the plane strain, and $g = 3.5 \times 10^3$ in the case of the full three-dimensional analysis. For the first two-dimensional case, the z-direction is assumed to provide a zero strain condition. In both cases, the bottom face of the truss is subject to rolling boundary conditions, i.e., $v_y = 0$. Notice further that a control point A is considered in the center of the top surface of the truss. The spatial origin is located in the center of the bottom plate, respectively. The CBG uses the spatial origin as its origin to make sure that the CBG always uses the symmetry of the problem as edges for the computational cells. This arrangement is motivated in order to recover the results from Sadeghirad et al. [119].

l	$= 1.0$ m
E	$= 1.0 \times 10^6\,\mathrm{N\,m^{-2}}$
ν	$= 0.3$
ρ	$= 1050\,\mathrm{kg\,m^{-3}}$
Δt	$= 6.0 \times 10^{-5}\,\mathrm{s}$
t_{end}	$= 2.4 \times 10^{-1}\,\mathrm{s}$

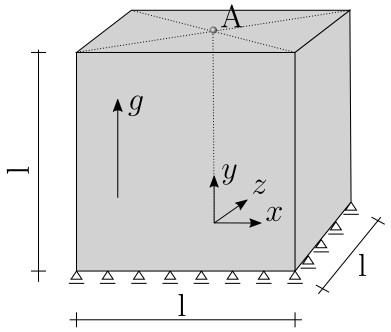


Figure 10.3: Simulation parameters (left) and illustration (right) of the truss benchmark problem.

In Sadeghirad et al. [119], the authors stated a uniform grid spacing of dx = 0.5 m, which is adopted here. The truss in this example is always discretized by material points with equal distance to each other using an equal divnision of the geometry. The parameter n is used to denote the number of material points per edge.

As a first investigation, we observe the problem of the extension instability, see figure 10.4. Using a relatively coarse discretization of $n = 6$ resulting in `NoMP` $= 6 \times 6 = 36$, the spurious separation of an upper part of the truss can be observed. Figure 10.4 shows a time snapshot of the simulation at $t = 8.4 \times 10^{-2}\,\mathrm{s}$ where the situation occurs that material points are separated by a whole cell of the CBG, causing this phenomenon.

In the next analysis, the same discretization pattern is applied for cpGIMP and CPDI variants. The initial configurations (compare figures 10.5 and 10.6) are equal to those used in Sadeghirad et al. [119]. For visualization purposes, the domain is shown, rather than the actual material points that lie in the center of their domains, respectively. Both figures show the same positions in time. It can easily be seen that for cpGIMP (see figure 10.5), the originally square domains are just deformed in $x-$ and $y-$direction, while the domains in CPDI (see figure 10.6) also exceed shear deformation. In both cases, the elongated domains prevent spurious discontinuities. It is noteworthy that the simulation in Sadeghirad et al. [119] showed extension instability for the presented discretization in the cpGIMP variant. The higher stability might be caused to implementation details, e.g., for build-in mapping corrections in `ELSE`. These corrections prevent the use of a mass cut-off value for near zero nodal masses as described in algorithm 10.

Both the cpGIMP and CPDI variants show the same deformation pattern, given their limitations. For comparison, an MPM simulation is performed, which does not show the extension instability due to a more dense initial discretization. Figure 10.7 shows snap-

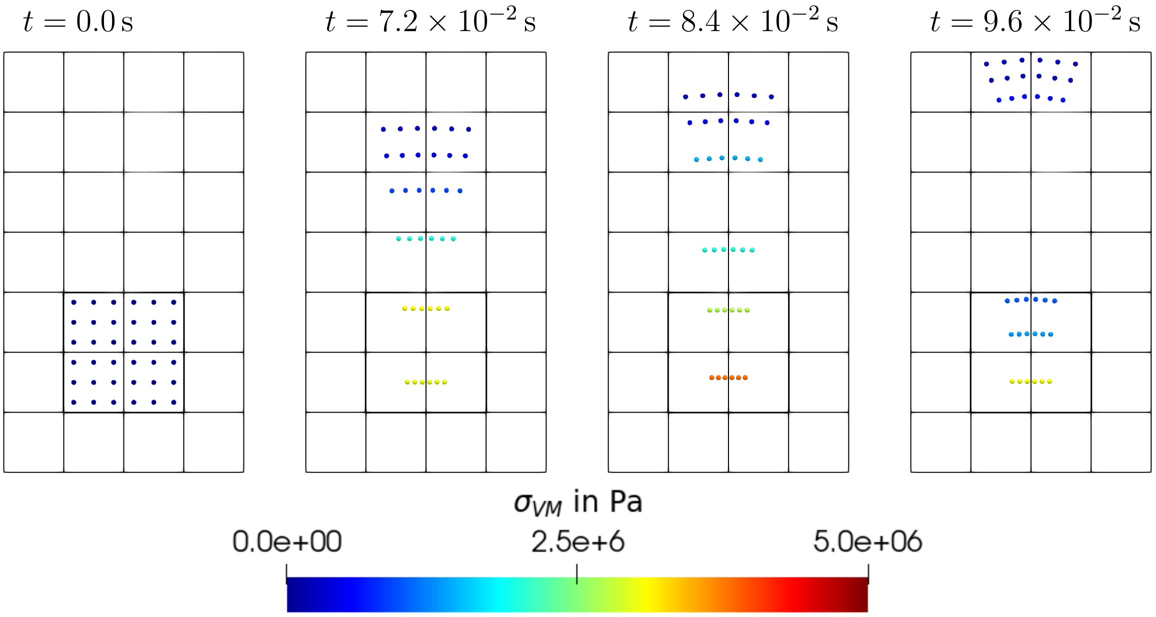


Figure 10.4: Visualization of the extension instability for the truss benchmark, using standard MPM with dx = 0.5 m and $n = 6$.

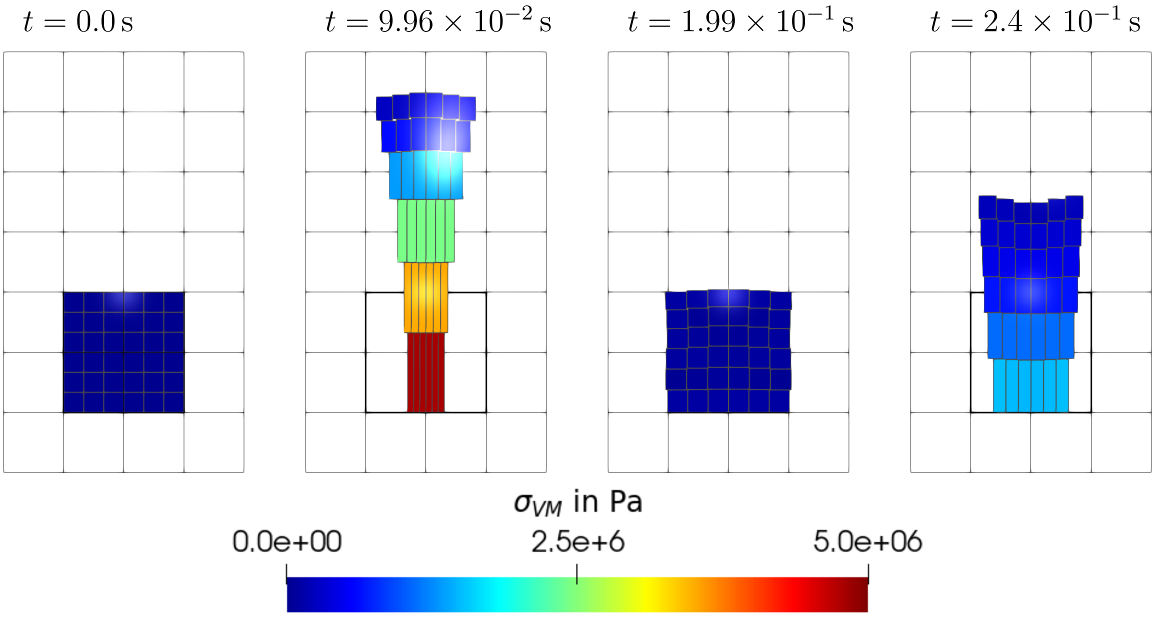


Figure 10.5: ELSE simulation for the truss benchmark, proposed in Sadeghirad et al. [119], using a cpGIMP variant. The ELSE simulation do not suffer from extension instability for the given discretization $n = 6$, as experienced by the primary source. The uniform grid spacing is fixed to dx = 0.5 m.

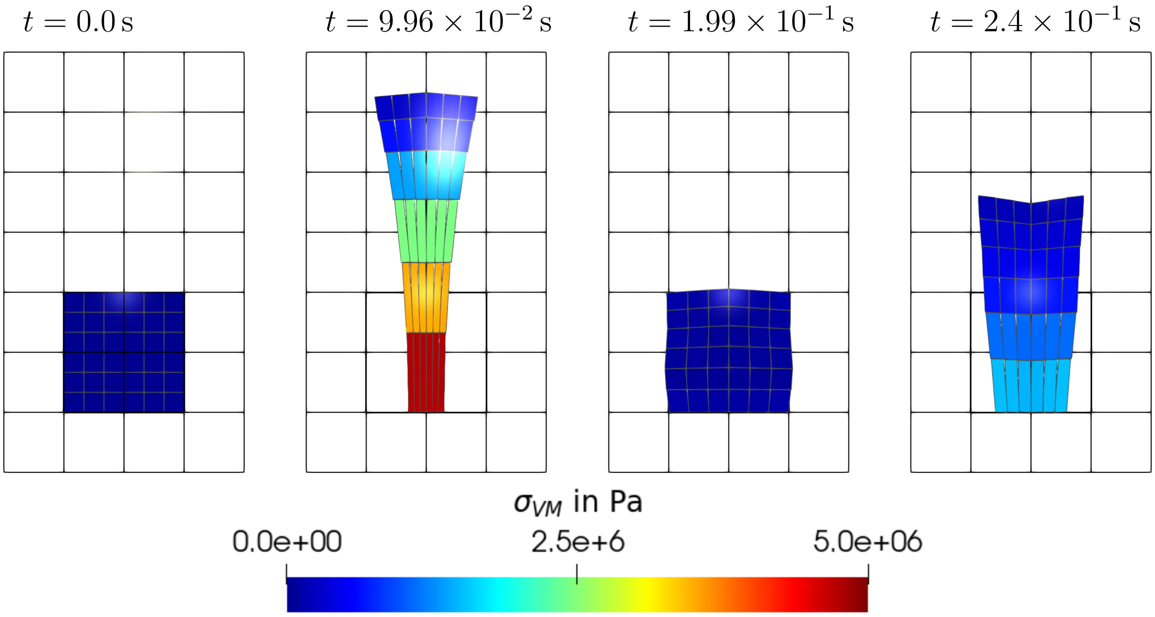


Figure 10.6: ELSE simulation for the truss benchmark, proposed in SADEGHIRAD ET AL. [119], using a CPDI variant with $n = 6$ and $\mathrm{dx} = 0.5\,\mathrm{m}$.

shots of this simulation. It can be seen that the results agree overall but show different deformations, especially after the first elongation.

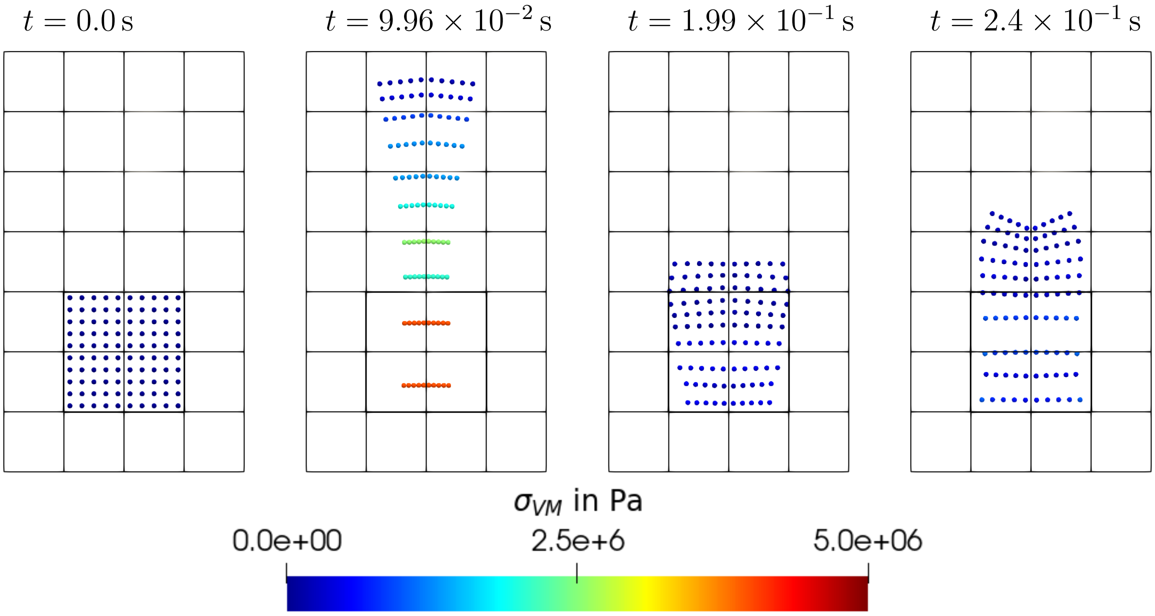


Figure 10.7: MPM simulation of the truss benchmark with just the right amount of initial material points at $n = 10$ to prevent extension instability.

In order to find a better solution for the truss problem, a convergency study is conducted. The results for dense discretization are visualized in figure 10.8. In this study, the standard MPM and the CPDI variant are used for comparison. Besides the number of material points, the grid size impact is analyzed as well. The CPDI discretization taken for comparison uses fewer material points than the MPM variant as it features a better convergence rate towards the final results. The deformation of the truss for a more dense

CBG is considerably different. It can be concluded that for this example, the sensitivity of the simulation with respect to dx is not neglect-able.

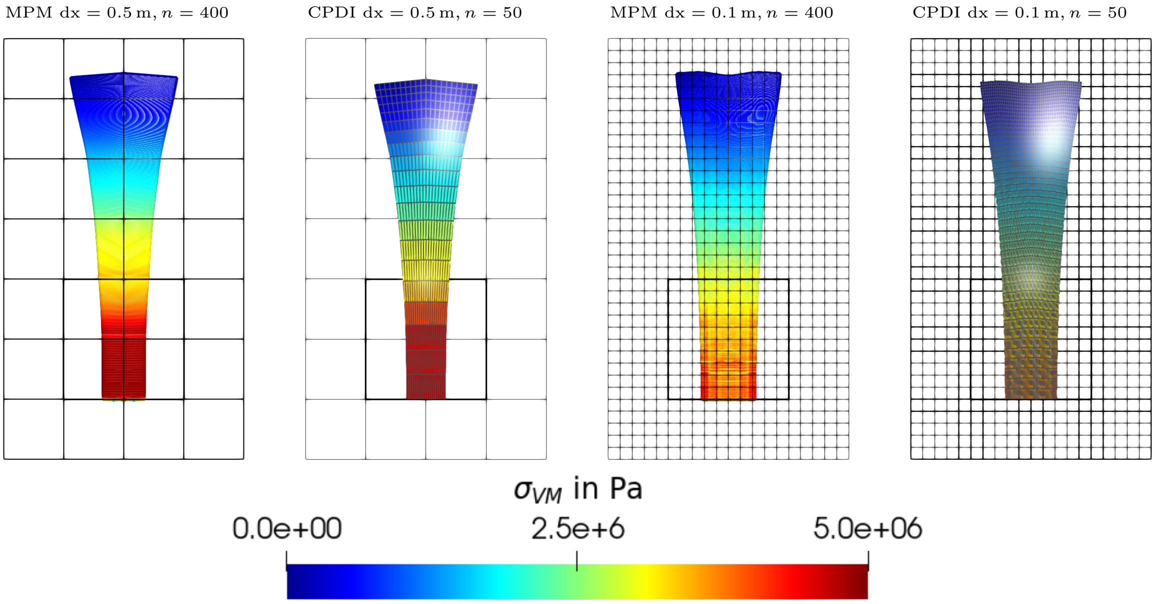


Figure 10.8: Comparison of truss shapes at $t = 1.99 \times 10^{-1}$ s for very fine discretization with two different CBG sizes.

Figure 10.9 shows the displacement of control point A over time during the simulation. For comparison, all visualized simulations are given with the results from Sadeghirad et al. [119], respectively.

It can be seen that at the same discretization density, the exact solution from Sadeghirad et al. [119] is recovered. Hence, the present CPDI implementation is validated. Further, the cpGIMP implementation not only works for this problem but also meets the same result. The MPM variant that uses more material points in the discretization to prevent the extension instability follows broadly the same pattern but shows a significant offset to the other solutions. As mentioned before, standard MPM requires a more dense discretization to converge to the model solution. For comparison, the dense discretizations at the smaller grid size dx $= 0.1$ m are given to show the influence. The first displacement amplitude shows a plateau rather than a pike.

For the second part of the analysis, we consider $g = 3.5 \times 10^3$, resulting in extreme deformations, up to 800%. In contrast to Sadeghirad et al. [119], the simulations are carried out in 3D to show the capabilities of the `ELSE` implementation. For the initial discretizations, the parameter n is also used in the z-direction, resulting in `NoMP`$= n^3$. Figure 10.10 provides snapshots of the simulation. Even at these large deformations in three dimensions, no extension instability occurs.

For finer discretization, however, a different phenomenon can be observed. While figure 10.11 shows a working discretization for the whole simulation time, figure 10.12 shows snapshots of a fine discretization with `NoMP`$= 50 \times 50 \times 50 = 125000$ CPDI material points and dx$= 0.1$ m. This set of parameters worked before in the plane strain analysis but now exposes a break-off behavior from its support, shown in the last snapshot in figure 10.12. The cause of this break can be observed already within the snapshot at time $t = 7.44 \times 10^{-2}$ s. A closeup of the truss's support and an enlarged view of the supports

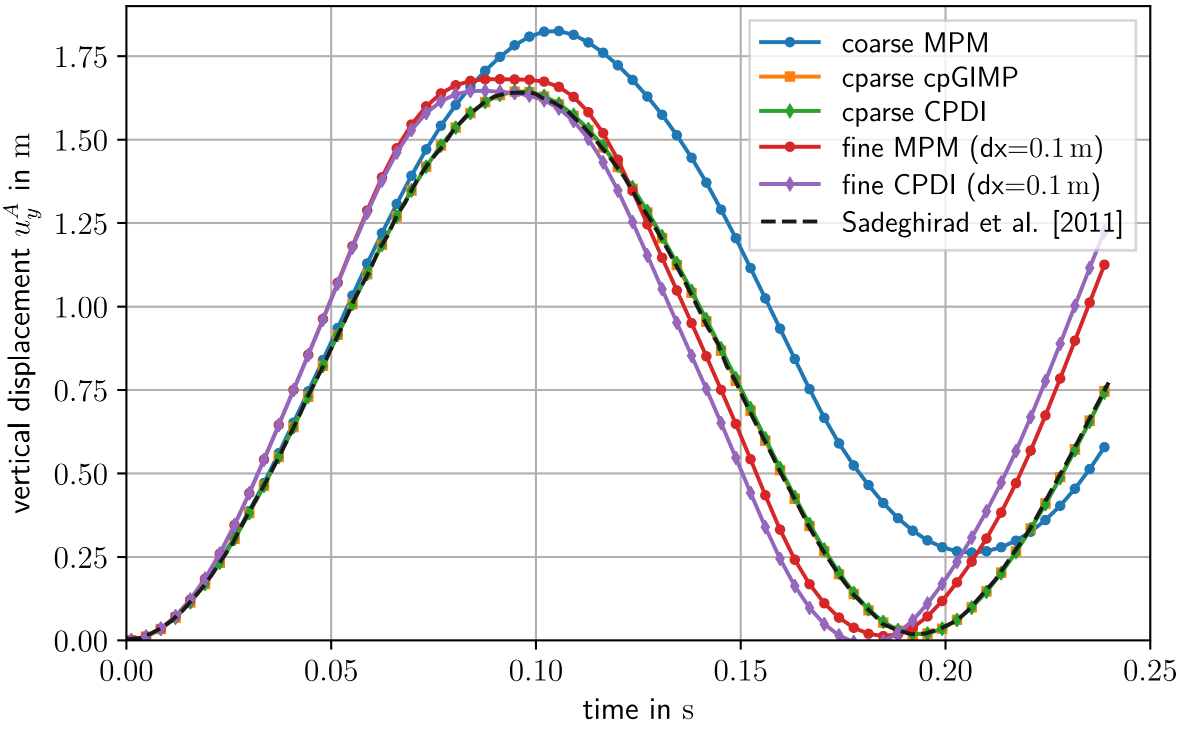
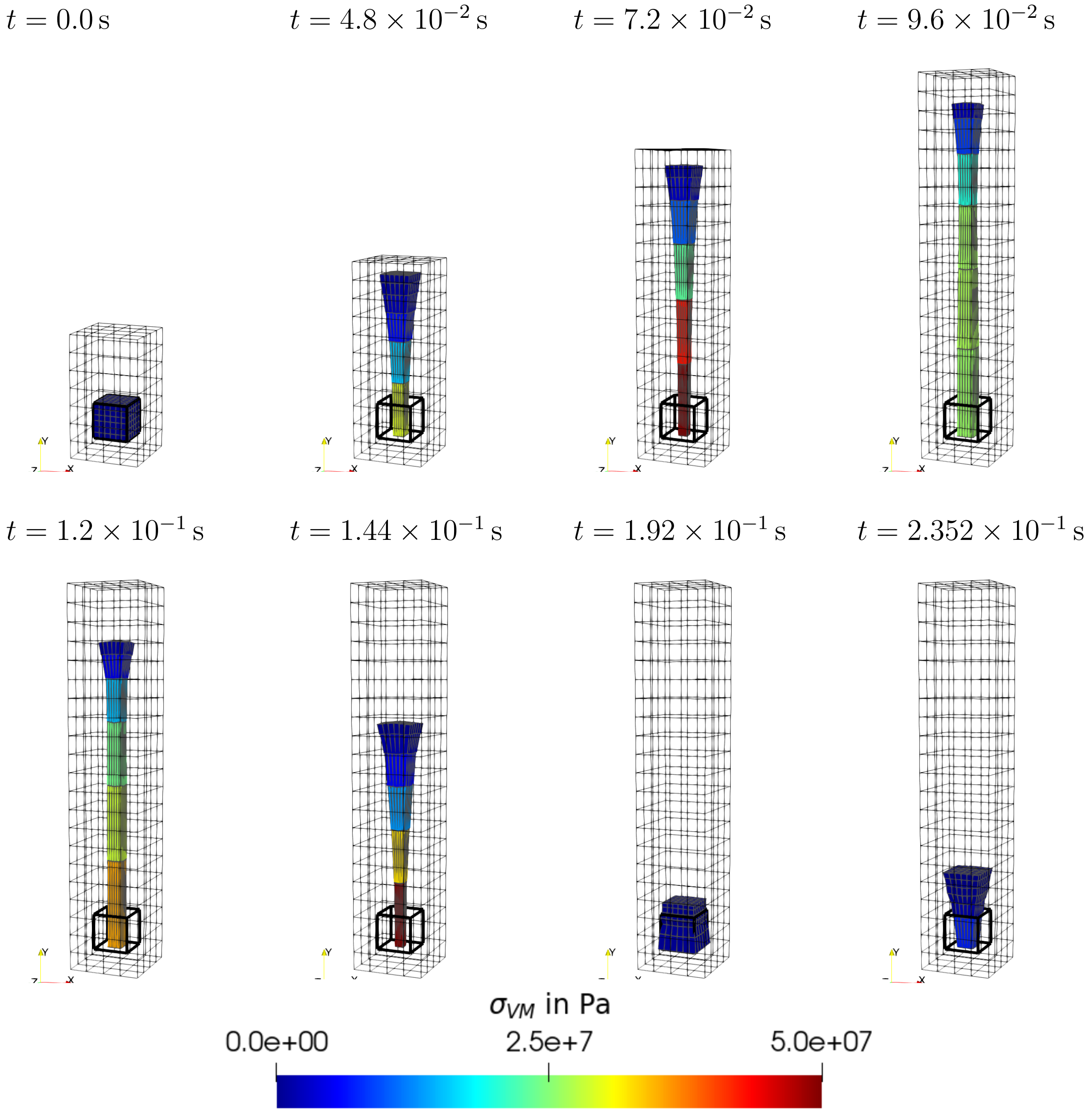


Figure 10.9: Vertical displacement of control point A over time for various discretizations using standard MPM and CPDI on the plane strain truss problem.

corners are given in figure 10.13. The boundary conditions are most effective if vertices of the material points near the bottom are close to the geometric support line of $y = 0$. This generally is the case, compare, e.g., figure 10.6. In 3D, however, the resolved shear deformations applied to the material point's domain are larger, leading to higher distances of the domains vertices due to the update of equation (fig:CPDI1Q8). This phenomenon exaggerates and propagates to the middle of the support, starting from the very corner of the truss, as shown in figure 10.13. A stable simulation without break off requires an enlargement of the uniform grid size to prevent this strong localization.

Figure 10.14 shows the vertical displacement of point A for the three-dimensional analysis. The results of Sadeghirad et al. [119] are included as well, although these were obtained under plane strain conditions. Again, using a similar discretization density than these authors lead to almost the same response. Notice that with an initial dimension of 1.0 m per edge, the truss deforms up to 800% in this problem. Hence, it represents an extreme case. The simulation could be successfully carried out with `NoMP`$= 20 \times 20 \times 20 = 8000$ CPDI domains but at a CBG spacing of dx$= 0.2$ m. For smaller grid spacings, the break off can be observed at different time-steps.

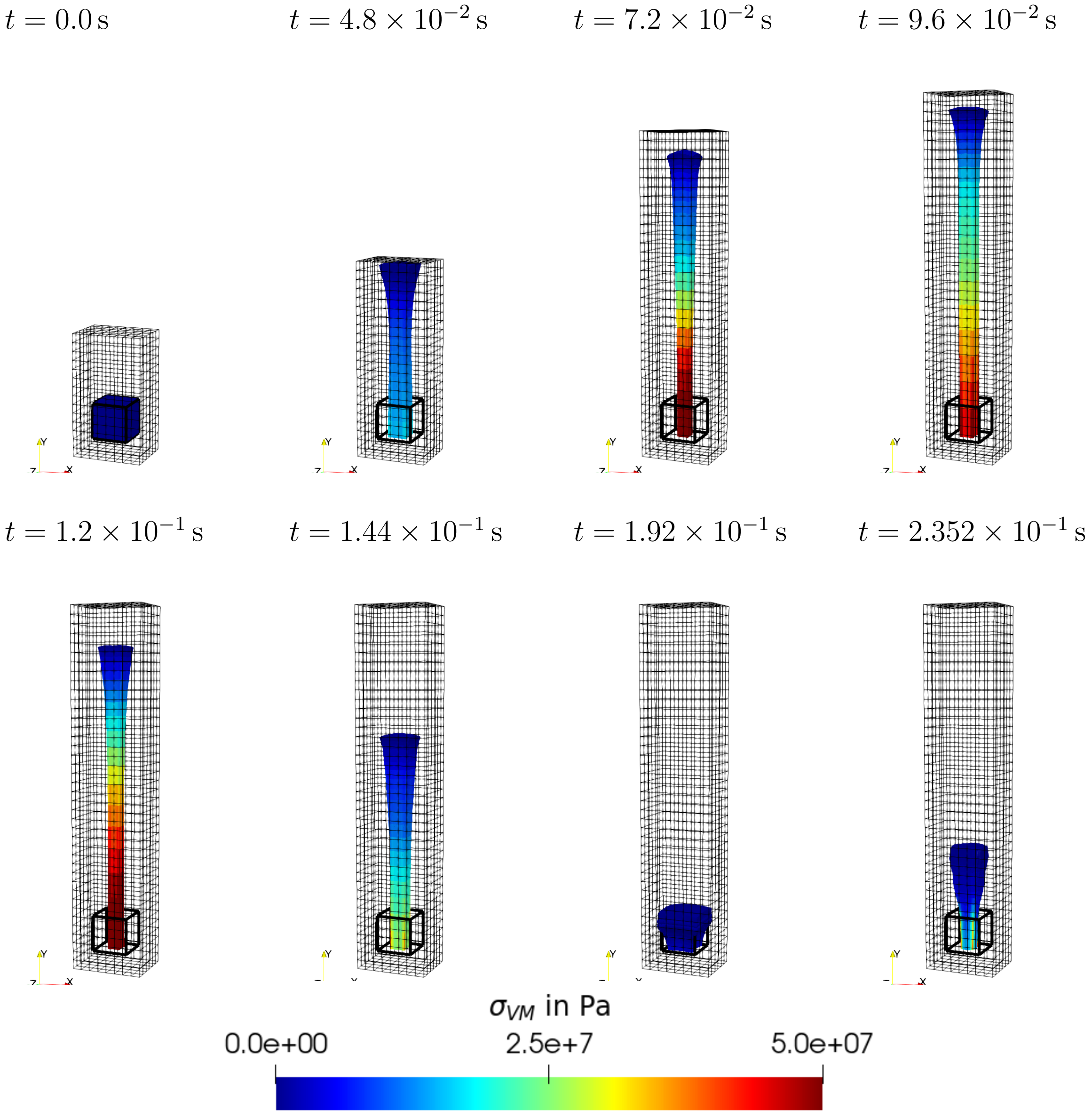
The truss benchmark analysis validates the CPDI implementation in `ELSE`. The results of Sadeghirad et al. [119] could be replicated exactly using equal or similar simulation parameters. The implemented cpGIMP variant performs even more reliable. An extension to a three-dimensional analysis showcases the benefits of the CPDI variant in that considerably fewer material points can be used to prevent extension instability. However, the results expose a tendency of the MPM towards spurious localization problems. In CPDI, a localized deformation has directly reflected the geometry of the material points



$t = 0.0\,\text{s}$ $t = 4.8 \times 10^{-2}\,\text{s}$ $t = 7.2 \times 10^{-2}\,\text{s}$ $t = 9.6 \times 10^{-2}\,\text{s}$

$t = 1.2 \times 10^{-1}\,\text{s}$ $t = 1.44 \times 10^{-1}\,\text{s}$ $t = 1.92 \times 10^{-1}\,\text{s}$ $t = 2.352 \times 10^{-1}\,\text{s}$

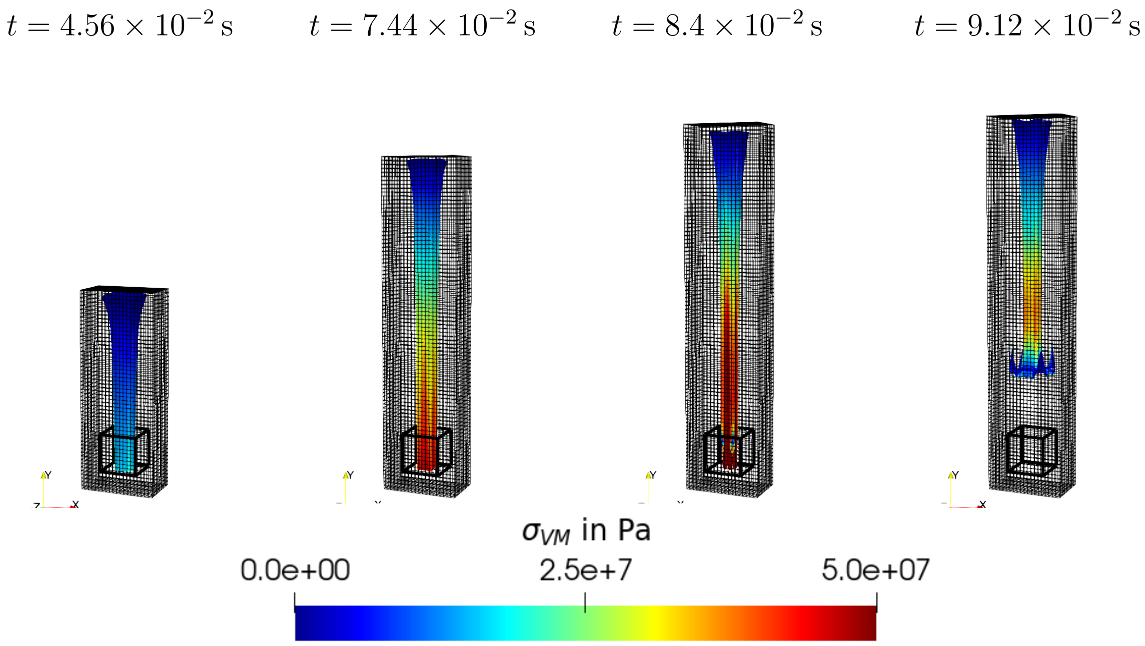
Figure 10.10: Deformation of the three dimensional truss, subject to $g = 3.5 \times 10^3$ at an initial discretization of `NoMP`$= 6 \times 6 \times 6 = 216$ CPDI material points and dx$= 0.5\,\text{m}$.



$t = 0.0\,\mathrm{s}$ $t = 4.8 \times 10^{-2}\,\mathrm{s}$ $t = 7.2 \times 10^{-2}\,\mathrm{s}$ $t = 9.6 \times 10^{-2}\,\mathrm{s}$

$t = 1.2 \times 10^{-1}\,\mathrm{s}$ $t = 1.44 \times 10^{-1}\,\mathrm{s}$ $t = 1.92 \times 10^{-1}\,\mathrm{s}$ $t = 2.352 \times 10^{-1}\,\mathrm{s}$

Figure 10.11: Deformation of the three dimensional truss, subject to $g = 3.5 \times 10^3$ at an initial discretization of `NoMP`$= 20 \times 20 \times 20 = 8000$ CPDI material points and dx$= 0.2\,\mathrm{m}$.



$t = 4.56 \times 10^{-2}\,\mathrm{s}$ $t = 7.44 \times 10^{-2}\,\mathrm{s}$ $t = 8.4 \times 10^{-2}\,\mathrm{s}$ $t = 9.12 \times 10^{-2}\,\mathrm{s}$

Figure 10.12: Deformation of the three dimensional truss, subject to $g = 3.5 \times 10^3$ at an initial discretization of `NoMP`$= 50 \times 50 \times 50 = 125000$ CPDI material points and dx$= 0.1\,\mathrm{m}$.

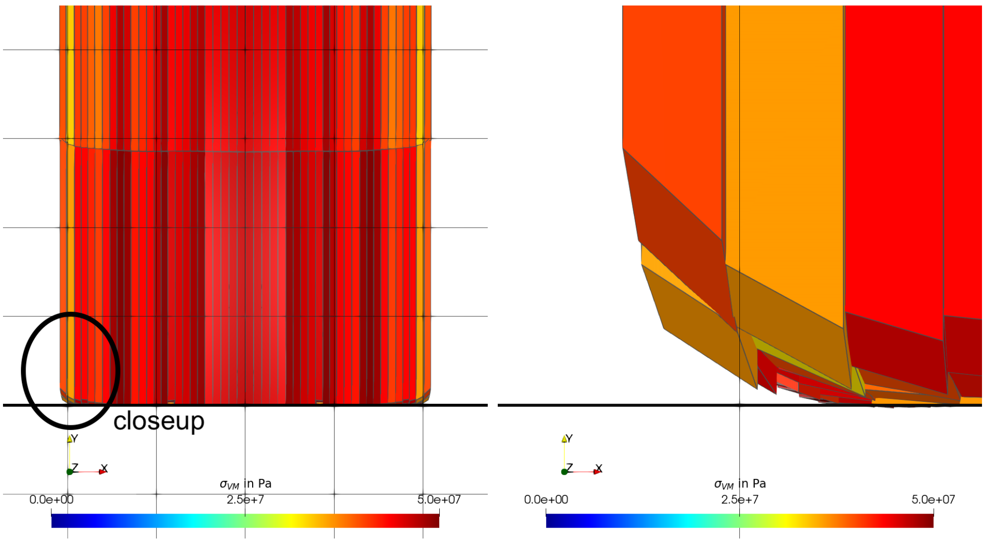


Figure 10.13: Observations of the support in the three dimensional truss analysis. The shown discretization feature `NoMP`$= 50 \times 50 \times 50 = 125000$ CPDI material points and a uniform grid spacing of dx$= 0.1\,\mathrm{m}$. Shown is the bottom of the truss at $t = 7.44 \times 10^{-2}\,\mathrm{s}$ in total (left) and a closeup perspective (right). One can identify a starting break-off of the truss from the support, driven by shear deformed material point domains.

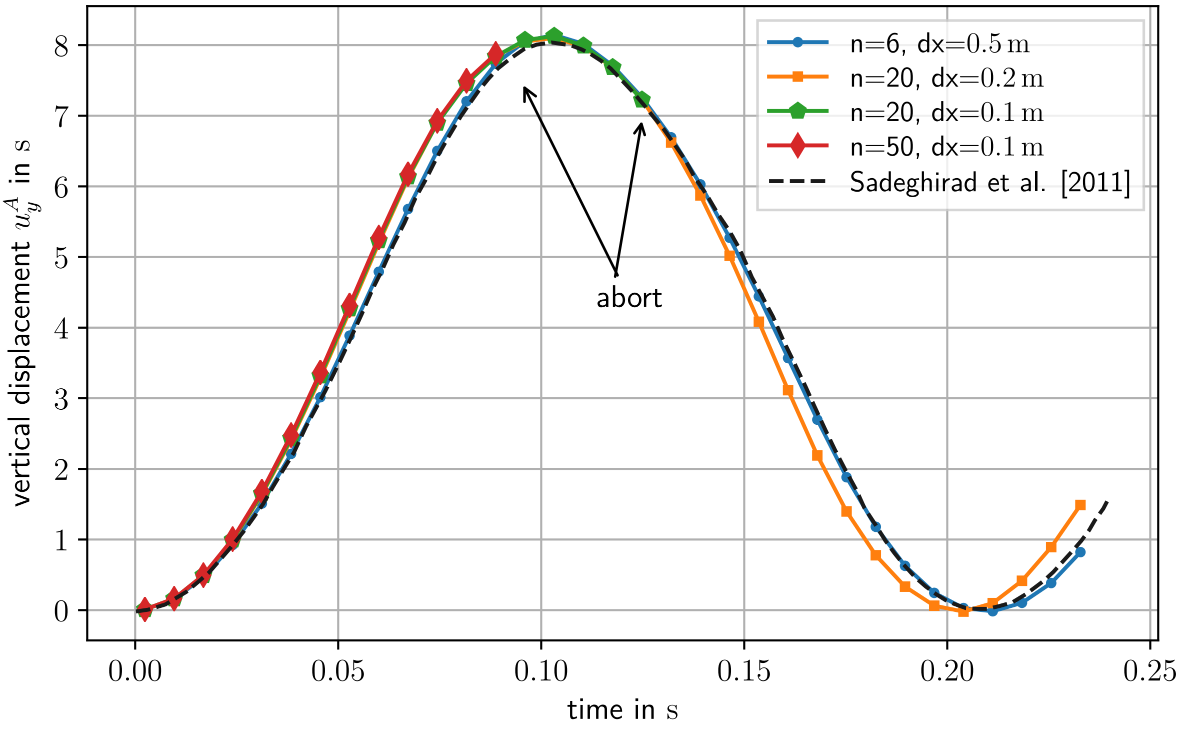


Figure 10.14: Vertical displacement of control point A over time for various CPDI discretizations of the three dimensional truss problem.

domain, which exaggerates these effects. A countermeasure to this problem was found by using larger CBG spacings, which might reduce the accuracy of the overall simulation. Special attention needs to be taken in the design of simulation parameters with these considerations in mind.

11 The Split-Hopkinson-Pressure-Bar experiment for material fitting

11.1 42CrMo4 steel in ”Virtual Machining” project

The Johnson-Cook material, discussed in chapter 4.3.8, is fitted to a 42CrMo4 steel, in view of dynamic simulation with metallic materials. 42CrMo4 is a low alloyed steel. It can be heat treated and is used in many engineering applications due to its high strength and durability. When used in cold forming processes, possible use cases are automotive drivetrains and gears. Also, in machinery, where parts are subject to high loads, 42CrMo4 represents a popular, cost-efficient choice offering high resistance. However, despite its advantages, its high strength and low thermal conductivity make it a challenging material for machining processes, compare, e.g., Xu et al. [159].

Consequently, investigation efforts are taken to research on the cutting behavior of 42CrMo4 to develop cutting profiles for better and more predictive machining, see for example, Bergmann et al. [12]. The microstructure changes, especially on the cutting surface, are known to have strong influence on the part's durability and resistance to wear. An analysis of this subsurface evolution on the considered 42CrMo4 steel can be found, e.g., in Kimm et al. [75]. For upcoming analysis in this thesis the focus lies on a variant of the steel that was used in the scope of the ”UA-Ruhr Professur *Virtual Machining*” and has a chemical composition of 0.429% C, 0.225% Si, 0.757% Mn, 0.014% P, 1.101% Cr, and 0.188% Mo.

The ” *Virtual Machining*” project is funded by the ”Stiftung Mercator” and the ”Mercator Research Center Ruhr” (PE-216-0024). In this project, an interdisciplinary approach was chosen to research machining process optimizations and develop numerical models. The work group is composed of institutes from three universities of the ”UA-Ruhr”, namely the ”Lehrstuhl Werkstofftechnik” at ”Ruhr-Universität Bochum”, the ”Insitut für Mechanik” at the ”Universität Duisburg-Essen” and the ”Lehrstuhl XIV Virtual Machining” at ”TU Dortmund University” where the scientific leadership is held by Prof. Dr.-Ing. Petra Wiederkehr. This thesis was written in the scope of this project, respectively.

For the analysis in this thesis, the ”QT660” variant from Kimm et al. [75] is chosen. In this state, the steel was austenitized at 850 °C and subsequently quenched and tempered at 660 °C. All temperature treatments were conducted in an inert atmosphere to prevent oxidation. The ”QT660” was also subject in Maassen et al. [94] before. The material parameters for a Johnson-Cook model are hand fitted, using simulations of experiments. Steady state tensile tests were conducted at the ”Lehrstuhl Werkstofftechnik”. The macroscopic data gathered from the tensile test are used to determine elastic modulus and parameters $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{N}$. A simulation with the JC-material subroutine was carried out on finite elements, without the influence of temperature $\gamma_{\mathrm{TQ}} = 0$ or plastic rate $\mathcal{C} = 0$. A comparison of numerical and experimental tensile data is given in figure 11.1.

The results in figure 11.1 represent a compromise to the overall fitting, described in the following. Further, the numerical analysis does not follow the strength degeneration of the experiments due to the limits of the JC-material model.

The challenge in dynamic analysis at high strain rates is the rate dependency in the plastic regime. In order to quantify this behavior, the Split-Hopkinson-Pressure-Bar (SHPB)

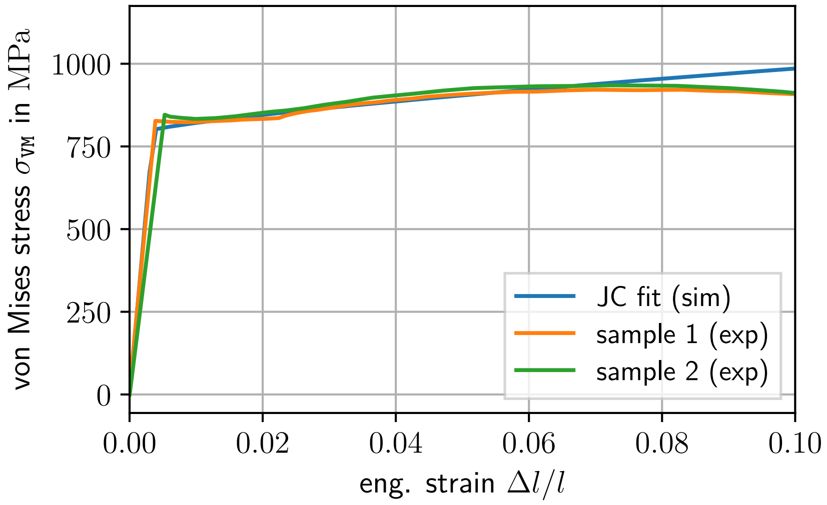


Figure 11.1: Quasi-static tensile data of 42CrMo4 from experiments and FEM simulations on a JC-material, at room temperature.

elastic modulus	E	$2.242 \times 10^{11}\,\mathrm{N\,m^{-2}}$
material density	ρ_0	$7850\,\mathrm{kg\,m^{-3}}$
Poisson's ratio	ν	0.3
initial yield strength	$\mathcal{A}$	$801 \times 10^{6}\,\mathrm{N\,m^{-2}}$
strain hardening modulus	$\mathcal{B}$	$1531 \times 10^{6}\,\mathrm{N\,m^{-2}}$
work hardening modulus	$\mathcal{N}$	0.81
plastic rate dependency parameter	$\mathcal{C}$	0.036
temperature dependency parameter	$\mathcal{M}$	1.48
reference plastic rate	$\dot{\varepsilon}_0^{pl}$	$5.6 \times 10^{-5}\,\mathrm{s^{-1}}$
reference temperature	θ_0	293.15 K
melting temperature	θ_m	1793.15 K
specific heat capacity	c_θ	$460\,\mathrm{J\,kg^{-1}\,K^{-1}}$
heat expansion coefficient	α_σ	$1.27 \times 10^{-5}\,\mathrm{K^{-1}}$

Table 11.1: Johnson-Cook material parameter for heat treated steel 42CrMo4. The parameters were fitted to quasi-static tensile data and SHPB experiments.

experiment can be used. As proposed in Maassen et al. [94], the MPM is qualified to model this highly dynamic experiment. The analysis of the SHPB is described in more detail in the next paragraph. Using the data from SHPB experiments, conducted at the "Lehrstuhl XIV Virtual Machining", the rate dependency parameter $\mathcal{C}$ and temperature dependency $\mathcal{M}$ were determined for the present JC-model. After initial separate fitting, all parameters were adjusted to provide an optimal agreement to all experimental data available. An overview of the used parameters is given in table 11.1. The missing model parameters, such as density or melting temperature, are taken from the literature or provided by the steel supplier.

11.2 MPM model of the SHPB

The Split-Hopkinson-Pressure-Bar experiment is used as a material test. Its main purpose is to provide material information in high strain rate regimes. Usually, well-known quasi-static ($\varepsilon \approx 1 \times 10^{-5}\,\mathrm{s}^{-1}$ to $1\,\mathrm{s}^{-1}$) tensile or upsetting tests are used for the determination of material behavior, and detect, e.g., the plastic yield strength in metallic materials. As long as the information is used in the context of slow processes, they are usually sufficient to predict process results.

For highly dynamic processes, the influence of rate dependency is a lot stronger. Also, inertia must not be neglected, thus, different techniques are applied. Extreme testing setups are considered in case of processes that involve impacts or penetrations ($\varepsilon \approx 1 \times 10^{5}\,\mathrm{s}^{-1}$ to $1 \times 10^{10}\,\mathrm{s}^{-1}$), e.g.,in armor analysis. Popular tests are the Taylor impact experiment or plate penetration test, see, e.g. Johnson and Holmquist [72], Zhang et al. [162], or Manes et al. [97].

The range of strain rates between $\varepsilon \approx 1\,\mathrm{s}^{-1}$ to $1 \times 10^{5}\,\mathrm{s}^{-1}$ is very important in many engineering applications. They occur, for example, during production in hot- and cold-forming, in failure situations such as car crashes or subtractive manufacturing. A popular choice to gather the material information in this range is the Split-Hopkinson-Pressure-Bar experiment, which is also subject to the analysis in this chapter. For an overview and review of different techniques for analyzing material behavior at high deformation, the interested reader is referred, e.g., to Field et al. [42].

The overall idea of the SHPB experiment is to load the specimen rapidly with tension or compression. This loading is achieved by placing the specimen between two relatively stiff and long bars. A strong connection must be established in the case of tensile loads, see, Al-Mousawi et al. [4]. However, tensile loads are not considered in this thesis. The load is now rapidly introduced at the free end of the Incident Bar (IB). Usually, gas cannons that accelerate a striker bar to impact the IB are used to archive the demanded load speeds. By design, the IB must not deform plastically under that load but rather transmits the resulting stress waves towards the contact surface to the specimen. While the waves travel through the IB, they are subject to dispersion effects. For an optimal test result, the wave forms a well-defined plateau. The wave is received by a strain gauge in a distance to the contact surface. As the wave reaches the specimen, it partly propagates and partly reflects back into the IB. This reflection is measured as well. Due to this load, the specimen deforms elasto-plastically, leading to a dissipation of energy. The stress wave is weakened but transmits into the Transmitter Bar (TB) at the end of the specimen. The transmitted part is now received by a second strain gauge attached to the TB. The main concept relies on the amount and profile of the dissipated energy, expressed by the measured difference in the strain gauges signals, to quantify the plastic behavior of the specimen under the present loading conditions.

Besides this main principle, the SHPB is built upon several assumptions, restrictions and variations. The basic mechanical assumption of a one-dimensional stress state along the axis of the bars and the specimen is the most strict one. In Al-Mousawi et al. [4] and Shin and Kim [127] an overview of the fundamental mechanical equations relevant in an SHPB are given. Further, the effects of the contact situations are neglected. Specifically, the effect of friction between the specimen and the bars, Hartley et al. [53] analyzed the effects on the resulting measurements. They saw considerable deviations and propose

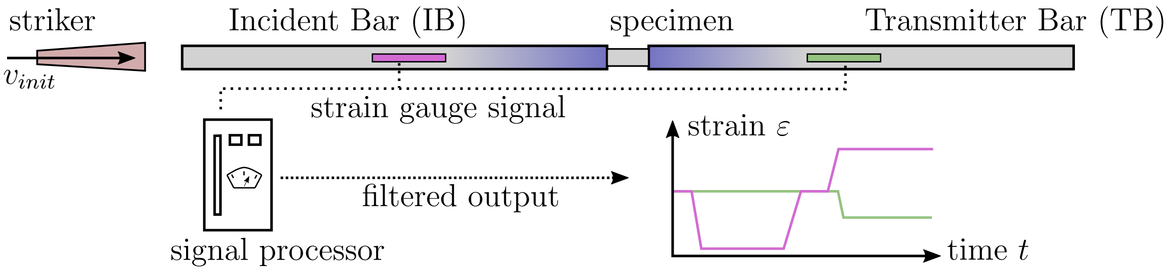


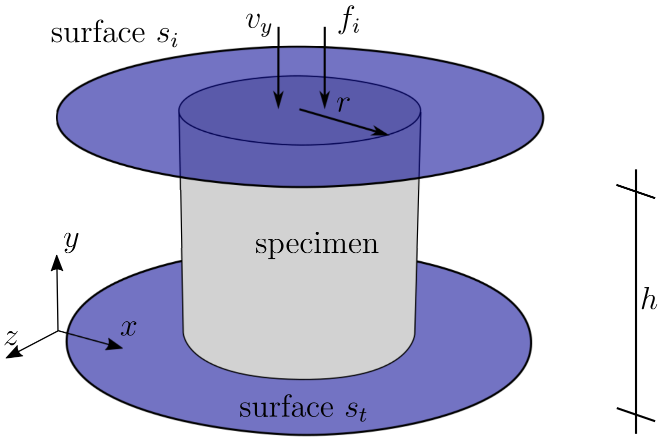
Figure 11.2: Schematic sketch of a SHPB setup.

special care in specimen preparation to minimize them. The post-processing of the raw signals from the strain gauges introduces further noise to the experiment. How strong these weaknesses impact the quality of the results depend on the specific dimensions of the bars, the placements of the gauges along, the gauge's signal quality and the executed wave speeds. To date, there are no standards defined for an SHPB. Consequently, SHPB obtained data are subject to considerable variations for similar or same materials, as shown in Kariem et al. [73]. Simulations of the process are usually conducted to fit the parameters of a specific constitutive material law. Ideally, the same numerical setup is used for fitting and evaluation of the processes to predict. For finite elements, this was done, for example, in Chandrasekaran et al. [22] and Manes et al. [96]. However, Afdahl et al. [2] used simulations before even manufacturing an SHPB test setup to evaluate the performance of the design in prior.

During the project ”Virtual Machining” an SHPB setup was used for performing tests on 42CrMo4 material. Pre-heated samples were also used to investigate the effect of temperature dependence, using an induction unit as shown in Zabel et al. [161].

The numerical MPM model of the SHPB plays an important role in the calibration process towards further dynamic problems using 42CrMo4. As described before, it is used for the parameter fitting on the applied JC-material model. Further, the numerical tools discussed up to now are applied to a real application. The calibration of the MPM involves suitable spatial and temporal discretizations and choices for simulation parameters such as the penalty contact parameter. An in-depth analysis of these aspects of the SPHB experiment was already done in Maassen et al. [94]. Here we build upon the findings of this work, respectively. However, the simulations in this chapter use the more elaborated JC-material and present different spatial discretizations. The same set of experimental data is chosen, featuring a pre-heated sample under a strain rate of $\varepsilon \approx 2661.9\,\mathrm{s}^{-1}$. The obtained strain signals were filtered using a low-pass filter based on Savitzky and Golay [123]. The gauges were placed in the middle of both IB and TB. Each bar is made from steel at an initial diameter of $14 \times 10^{-3}\,\mathrm{m}$ and measures $1.5\,\mathrm{m}$ in length. In combination with their known density of $9850\,\mathrm{kg\,m^{-3}}$ and elastic modulus of $210.0\,\mathrm{GPa}$, it is possible to compute the absolute force in the bar over time. In contrast to the more detailed analysis in Maassen et al. [94], just the recovered force signal of the IB is used to compare with numerical contact forces for the sake of a clear presentation.

The MPM model of the SHPB discretizes the actual specimen by material points and approximates the ends of IB and TB as rigid surfaces s_i and s_t. An illustration of the geometrical setup is given in figure 11.3. The idealization reduces the experiment to the relevant part to eliminate noises. With respect to figure 11.2, just the specimen and the contact surfaces are included in the model. The considered material is the 42CrMo4, for which material parameters in table 11.1 were used. There are no boundary conditions on the grid. However, the incident surface s_i follows a prescribed motion v_y. This velocity is computed from the experimental signal and leads to the same compression over time, as described in MAASSEN ET AL. [94]. The transmitter surface s_t does not move for the considered simulation time. Thus the load is introduced by the contact of the surfaces with the specimen leading to approximately 40% compression. Following the SHPB's first impact wave requires a simulation time of t_{end}.



r	$= 2.0 \times 10^{-3}\,\mathrm{m}$
h	$= 4.0 \times 10^{-3}\,\mathrm{m}$
v_y	$= 10.6476\,\mathrm{m\,s^{-1}}$
μ	$= 0.1$
κ_{pen}	$= 1.0 \times 10^{8}\,\mathrm{kg\,s^{-2}}$
Δt	$= 5.0 \times 10^{-9}\,\mathrm{s}$
t_{end}	$= 200.0 \times 10^{-6}\,\mathrm{s}$
θ_{init}	$= 400.0\,^{\circ}\mathrm{C}$
γ_{TQ}	$= 0.85$

Figure 11.3: Visualisation of the SHPB problem (left) and simulation parameters (right).

Suitable contact parameters were taken from MAASSEN ET AL. [94]. All simulations are conducted using a constant time step size, as given in figure 11.3.

The present analysis features three standard MPM discretizations and three CPDI variants. In contrast to the discretizations used in MAASSEN ET AL. [94], the CPDI meshes in this analysis are regular. This scheme enables an argument on the stability and quality of the simulations with respect to the used variant and discretization pattern. An overview of the set of considered discretizations is given in table 11.2.

	dx/ dy / dz	NoMP	CPDI vertices	MP layers in height
MPM-B1	1.0×10^{-3}	6320		20
MPM-B2	6.0×10^{-4}	50240		40
MPM-B3	4.0×10^{-4}	170160		60
CPDI-B1	8.0×10^{-4}	1300	10400	20
CPDI-B2	6.0×10^{-4}	11132	89056	44
CPDI-B3	4.0×10^{-4}	36936	295488	54

Table 11.2: Listing of MPM and CPDI spatial discretizations used in the SHPB simulation.

Results obtained with the MPM variant are shown in figure 11.4. The graph compares the experimental force from the IB with the contact force f_i amplitude of the incident surface s_i. The graph is accompanied by deformation snapshots of the highest discretization MPM-B3. The morphology of the specimen during the deformation shows a clear round geometry throughout the simulation. As expected, due to the slight amount of friction, the specimen bulges outside. This behavior leads to the typical bulbous geometry of deformed SHPB samples. For the MPM-B2 and MPM-B3, very good agreement with the experimental force amplitude over the relevant time span of the experiment can be observed. A spike at the initialization of the stress wave into the specimen that can be seen in the experimental curve is not covered. However, this spike is most likely caused by not completely even surfaces and accelerated by imperfect signal capture intervals in the physical setup. The numerical model, however, does not show these imperfections. The force measured from the experiment drops after $t = 150.0 \times 10^{-6}\,\mathrm{s}$, caused by a separation of the specimen and the bars. Data gathered after that point is not considered for the fitting, respectively. Comparing the forces for the MPM discretizations shows that MPM-B2 and MPM-B3 do not show much difference. The MPM-B1 version, however, shows a completely odd force curve. At first, the load is not initiated at the same point in time, than for the other discretizations. The reason for this is that contact between the material point themselves and not with the implied geometry is considered. Consequently, the initial distance between material points and the surfaces is larger in this variant. The same effect is true for all other discretizations but does not show such a strong influence. Thus it can be seen that the contact on the MPM-B3 is initiated slightly earlier than on MPM-B2. Nevertheless, the force of MPM-B1 has a completely different slope over time, which gets clearer at an inspection of the deformation morphology of this discretization later on.

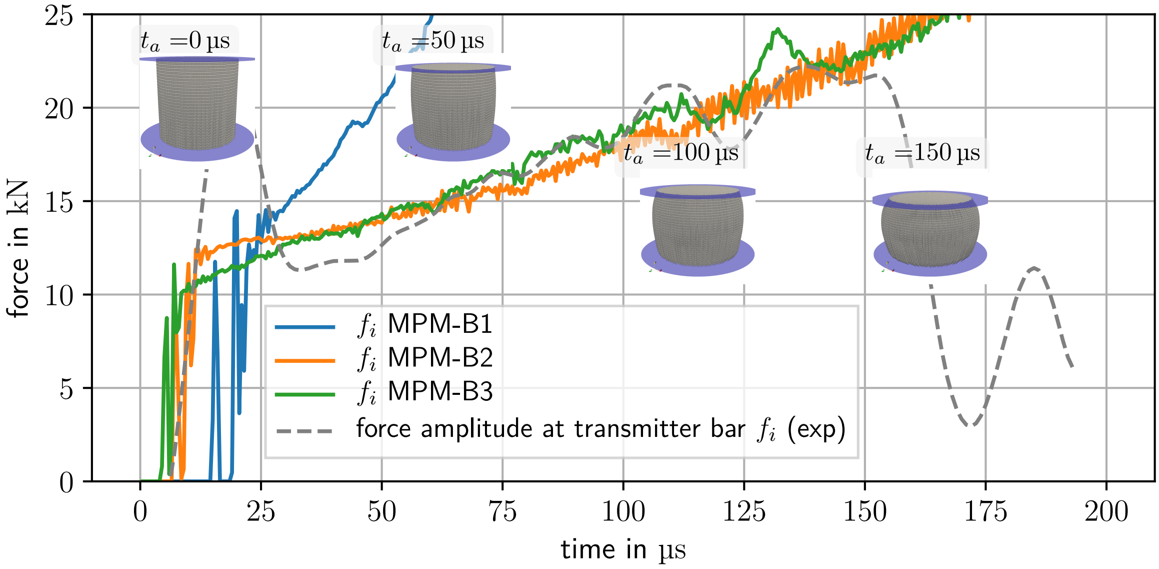


Figure 11.4: Contact force amplitudes on the incident surface s_i for the MPM discretizations over time. The experimental force is included for comparison. Current deformation snapshots of the MPM-B3 discretization are shown besides the graphs.

The results obtained by the CPDI variant are shown in the same fashion in figure 11.5. Overall, the same trends and observations can be made. The deformation pattern is as reasonable as with the MPM variant. Generally, the forces are slightly lower in amplitude and show more noise but closer around a mean value. The CPDI set also shows the already discussed offset in contact initialization due to their initial discretization. In contrast to the MPM-B1, the CPDI-B1 still holds the same slope for the force over time. It can even be the same, considering a shift along the time axis, making the contact initialization point agree with the other data. It can be concluded that the material parameters in table 11.1 represent a good fit for the 42CrMo4 steel, given that all simulations were conducted with the same material parameters and are in good agreement with the contact forces with the experimental data. The material even shows weak sensitivity to the chosen variants.

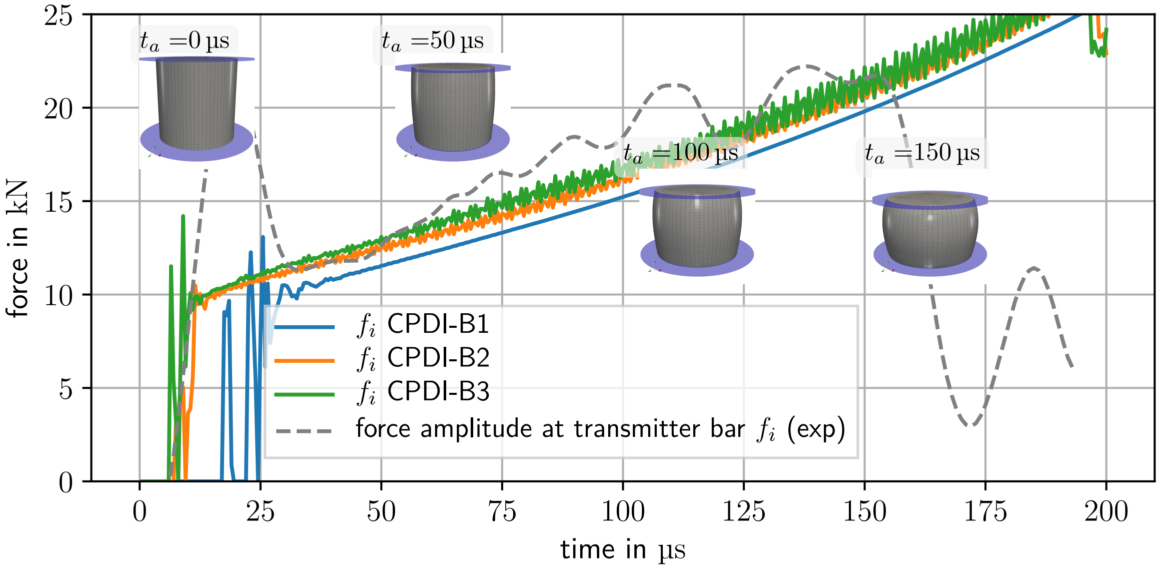


Figure 11.5: Contact force amplitudes on the incident surface s_i for the CPDI discretizations over time. The experimental force is included for comparison. Current deformation snapshots of the CPDI-B3 discretization are shown besides the graphs.

For a closer analysis of the simulation results, figures 11.6 and 11.7 show a view inside the specimen after $t = 150.0 \times 10^{-6}$ s. The inside view is realized by excluding the front-facing quarter of the specimen in post-processing, respectively. For each discretization, the temperature distribution and von Mises stress distribution are shown on the current deformed configuration. All temperature and stress plots are bounded to the same color scale for better comparison, respectively.

The general impression is that only minor differences can be found for the different variants regarding temperature and stress distribution. Notice that the visualized results differ from their actual total compression due to the slightly shifted time of contact initialization. This is more strongly visible on the coarsest discretizations.

Consequently, these coarsest discretizations MPM-B1 and CPDI-B1, show less deformation. The MPM-B1 deformation strongly deviates from all the others. Further a different stress and temperature state can be obeserved. Its shape at the shown time is non-symmetric with respect to its height. This behavior is reflected by the contact forces to the surfaces, compare figure 11.4. We can conclude that the discretization is too coarse to approximate the SPHB reasonably. Contrary, the CPDI-B1 produces better results, even though it features six times lesser CPDI domains than the number of material points in MPM-B1. However, its shape is also slightly non-symmetric. Stresses and temperature concentrate on the lower part of the sample. A possible explanation is that the CBG used for the simulations is not symmetric either. Although this is the case for all considered discretizations, it can be concluded that for larger grid spacings, the influence of the CBG geometry is much stronger. The CPDI scheme is less prone to these effects, also shown here in a compressive scenario.

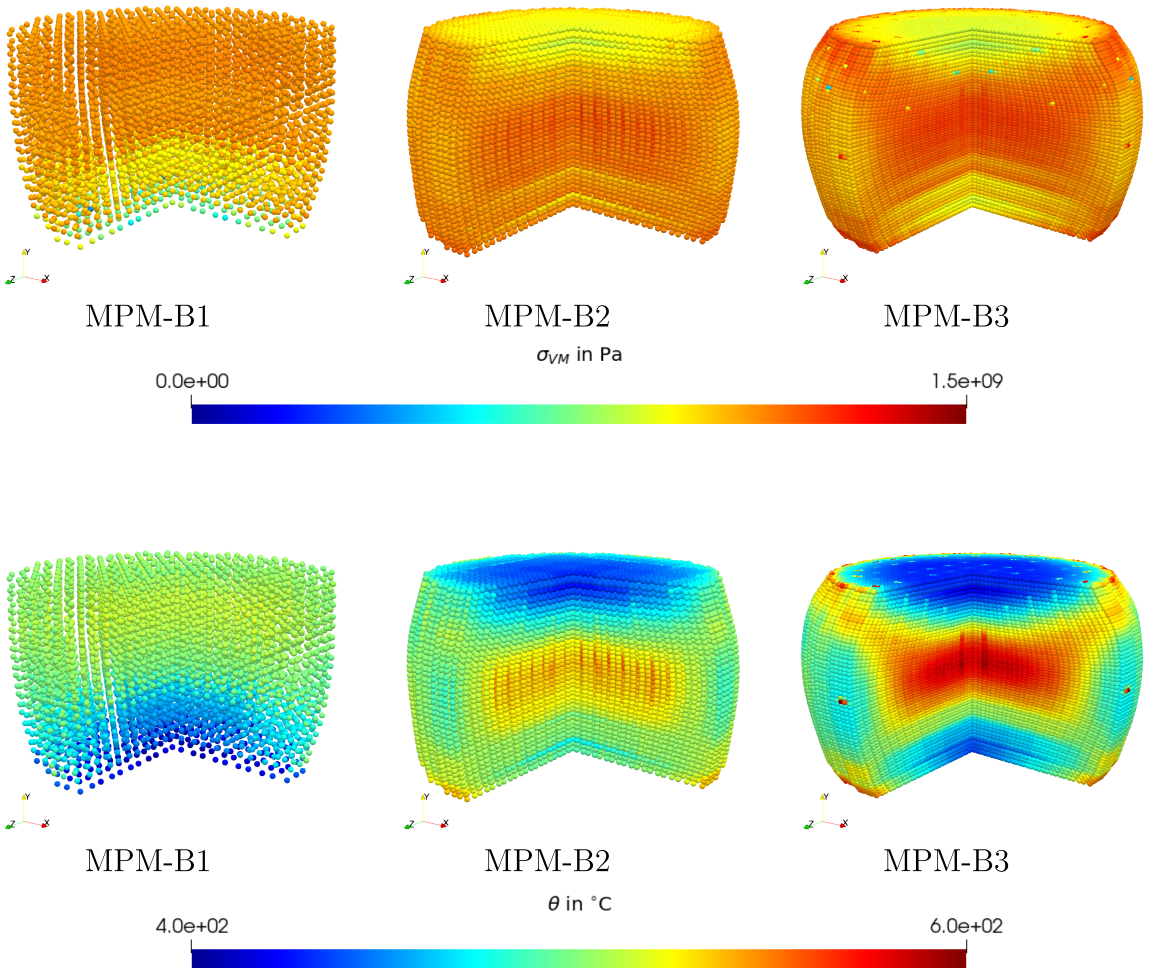


Figure 11.6: Visualizations of temperature and von Mises stress $\sigma_{\mathtt{VM}}$ in the current deformation on the MPM discretizations at $t = 150.0\,\mu\text{s}$. For an insight into the specimen a quarter of it in the front-view is cut from the visualization.

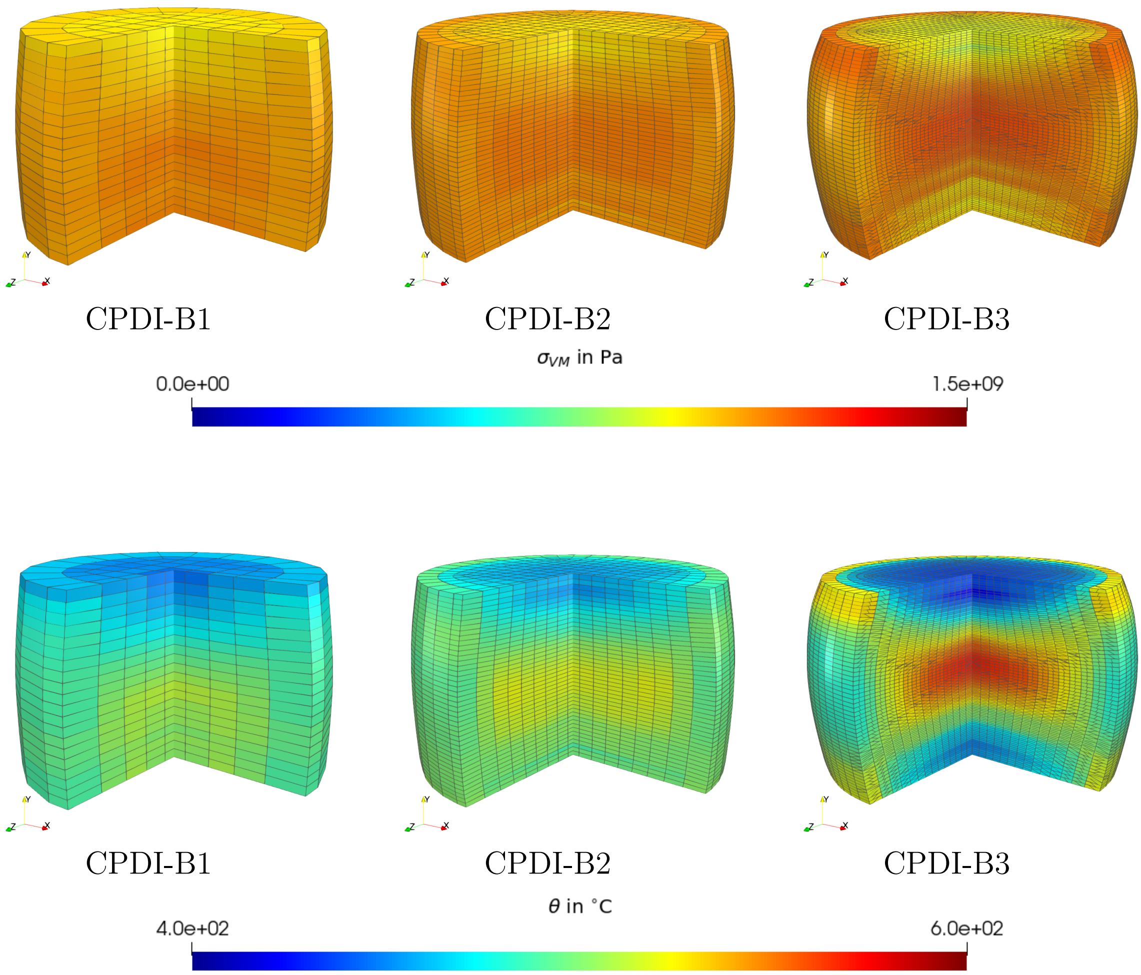


Figure 11.7: Visualizations of temperature and von Mises stress $\sigma_{\mathtt{VM}}$ in the current deformation on the CPDI discretizations at $t = 150.0\,\mu\text{s}$. For an insight into the specimen a quarter of it in the front-view is cut from the visualization.

For all finer discretizations the results are more realistic. Also, they do not show perfect symmetry, which is not expected due to the effects of inertia and considering the induced load to be directional. The stresses and temperature in a metal SHPB specimen should concentrate in the center and build up an X-pattern towards the edges of the top and bottom surfaces. This stress pattern was shown in many simulations, e.g., Manes et al. [96], and can be observed from investigations of the specimen in post-experimental analysis, compare, Kimm et al. [76]. It can be noticed that the MPM-B3 discretization shows some spikes in temperature and stresses on some material points. Also, the temperature localizes stronger on the edges of the contact surfaces. Additionally, the pike-temperature is generally higher, which can also be observed when comparing MPM-B2 and CPDI-B2.

In the analysis done in Maassen et al. [94], irregular meshes were considered for the specimen's CPDI discretization. It was also shown that a very dense discretization causes the analysis to abort early. In this case, a strong localization could be observed. With the presented regular meshes for the CPDI scheme, these failure modes are avoided. In fact, the CPDI-B3 variant is even denser than the critical one in the mentioned publication. Also, the results obtained are more consistent with respect to the deformation pattern. It can be concluded, that the MPM (or CPDI alike) shows a considerable dependency towards the initial discretization. In the SPHB model, a high degree of symmetry can be observed, and the simulation stability is increased if the discretization reflects this.

Comparing the CPDI and MPM variants the CPDI produced more smooth fields. Also, the macroscopic response regarding the contact forces is more smooth. On the contrary, a stronger localization can be observed in the MPM. Nevertheless, some of these localizations seem unphysical. When comparing the variants, the actual difference of `NoMP` needs to be considered. The MPM discretizations generally have way more material points, which the CPDI counterparts compensate with their advanced mapping capabilities. Due to these, higher stability is achieved as unphysical localizations are avoided. On the other hand, actual localizations, as observed in experiments along with the shear bands and the center of the specimen, are not visible in the presented discretizations. Therefore it is probably necessary to use very fine discretizations to resolve them, which would be very expensive in a CPDI scheme. As an advantage, it can be pointed out that the shape of the deformation is represented by CPDI, even at coarse discretization levels.

The MPM variants, excluding the MPM-B1, localize stronger than their CPDI companions. This is most likely related to their higher material point density at comparable computational costs. Yet, artificial localizations can be observed. At higher discretizations, the deformation is reasonable and well reconstructed even without a clear boundary definition. The displayed SHPB analysis show good agreement in macroscopic force measures, deformation patterns and stress/temperature distributions. Using regular meshes in CPDI's initial discretizations highly increases the numerical stability by which the quality of the results is also increased. The used JC-material shows a good fit for the considered 42CrMo steel, as the experimental key measures are reproduced well at almost any discretization. The choice of MPM variant should be taken in the context of the expected results. Generally, the CPDI scheme shows great advantages with respect to numerical stability and geometry approximation. Nevertheless, it shows weaknesses in approximating local phenomena due to its smooth representations. If it is expected that localizations or local discontinuities occur, an MPM scheme is advisable.

12 Metal cutting MPM simulation

Manufacturing processes where metal cutting occurs are of various types. Most commonly, turning, drilling and milling are to be mentioned. For a more detailed insight into the subject from a practical and theoretical perspective, the interested reader is referred to the textbook of Childs et al. [28]. High demand for numerical modeling of these processes is given to avoid expensive experiments for their wide range of applications. However, the highly dynamic conditions in these processes, the complex physical and constitutive phenomena, the extreme material deformation, and discontinuities that occur in chip formation are still challenging in numerical simulations. A comprehensive review of the state of modern machining simulations can be found in Rodríguez et al. [115].

Especially regarding the material discontinuities, which is challenging for mesh-based Lagrangian techniques such as FEM, the use of particle methods is motivated. In the following, two types of cutting problems are investigated. Both consider idealized conditions and are carried out in two-dimension only. Although these problems are far from more complex processes, such as milling, they represent typical benchmarks in the cutting simulation community.

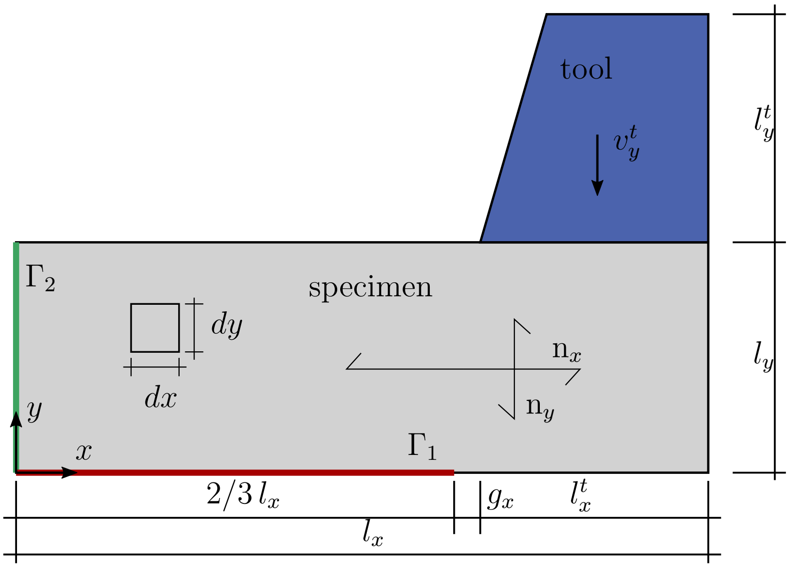
At first, a vertical cut is simulated, and in this problem, the grid-shift technique is introduced to enhance the simulation results. As a second example, an orthogonal cut is considered. Both cutting problems are analyzed with respect to numerical stability and parameter sensitivity.

12.1 A discussion of the MPM cutting simulations using `ELSE`

The cutting simulations using the MPM of this thesis are conducted using the `ELSE` code. As a constitutive routine, the Johnson-Cook material as described in 4.3.8 is used. An additional damage model, which is mandatory for modeling segregated chip formation, is not employed, basically for the lack of damage parameters on the considered materials. However, this approach is commonly used in the literature. Another restriction is the consideration of adiabatic conditions. At this state, the `ELSE` code does not support heat conductivity. Nevertheless, the cutting simulations use high cutting speeds. These lead to shorter simulation times and support the suitability of the adiabatic assumption. Further, contact conditions between the tool and the workpiece are known to have a considerable impact in metal cutting. The current contact mechanism implemented in `ELSE` does not include elaborate models but only the Coulomb type. For cutting simulations in general, it is also important to include the heat generated from frictional dissipation, use temperature dependent friction parameters and more elaborated models. The upcoming cutting simulations consider fast processes usually done using sharp tools and under the application of lubricants. As an approximation, the simulations consider no- or very low friction coefficients. Also, the tools do not feature a cutting blade radius but are modeled with an idealized sharp tip.

12.2 Vertical cutting analysis - introducing grid-shift technique

The vertical cutting process considered in the following describes cutting a strip from a metal sheet. Hence, a plane strain idealization is used to analyze the material behavior. A similar setup for vertical cutting was considered in Oñate et al. [109] and Sabel [118] in the context of PFEM. Aluminum is considered as a workpiece material. The parameters for the JC-constitutive model were taken from the literature and summarized in table 12.1. Further, the geometry and fixed parameters of the problem are given in figure 12.1. Material point discretizations and grid sizes were combined in sets to investigate the sensitivity of the simulation to these parameters. The sets are outlined in table 12.2. All simulations of the virtual cut use a constant time step size of $\Delta t = 5 \times 10^{-10}\,\mathrm{s}$, and $\Delta t = 1 \times 10^{-10}\,\mathrm{s}$ for any set with ny= 200, respectively.



l_x	$= 3 \times 10^{-3}\,\mathrm{m}$
g_x	$= 1 \times 10^{-5}\,\mathrm{m}$
l_x^t	$= 1 \times 10^{-3}\,\mathrm{m}$
l_y	$= 1 \times 10^{-3}\,\mathrm{m}$
l_y^t	$= 1 \times 10^{-3}\,\mathrm{m}$
v_y^t	$= -5 \times 10^{1}\,\mathrm{m\,s^{-1}}$

Figure 12.1: Vertical cutting problem illustration and geometrical simulation parameter.

The problem considers a part of the sheet which cantilevers from a rigid surface. The rigid surface is idealized by boundary conditions applied to boundary Γ_1 as depicted in red in figure 12.1. The full extend of the sheet metal is not considered. Boundary conditions along Γ_2 are used to approximate the transition to the neglected part. The specific type of boundary conditions is discussed later. A rigid tool is used for the actual cutting. It transitions downwards with a constant velocity v_y^t and leaves a small gap g_x between its tip and the support of the sheet.

Initially, the problem is simulated with standard MPM. The boundary conditions used at the support and virtual cut of the workpiece are grid boundary conditions. In particular, boundary Γ_1, we employ $v_x^I = 0\,\mathrm{m\,s^{-1}}\,\forall I$ with $x_x^I \leq 0\,\mathrm{m}$, and for Γ_2 we set $v_x^I = 0\,\mathrm{m\,s^{-1}} \wedge v_y^I = 0\,\mathrm{m}\,\forall I$ with $x_y^I \leq 0\,\mathrm{m}$. Notice that with the spatial origin being the geometrical origin of the CBG it is guaranteed that grid nodes are placed on the boundaries, respectively.

For the first set of three simulations, the discretizations of S0, S1, and S2 are used. The vertical force component of the contact force, measured at the rigid tool, is further taken as a macroscopic benchmark criterion. Figure 12.2 shows visualizations of the simulation for S0 until the simulation aborts. The final time of $t_{end} = 2.0 \times 10^{-5}\,\mathrm{s}$ denotes a relative

elastic modulus	E	$72 \times 10^{9}\,\mathrm{N\,m^{-2}}$
material density	ρ_0	$2800\,\mathrm{kg\,m^{-3}}$
Poisson's ratio	ν	0.33
initial yield strength	$\mathcal{A}$	$546 \times 10^{6}\,\mathrm{N\,m^{-2}}$
strain hardening modulus	$\mathcal{B}$	$678 \times 10^{6}\,\mathrm{N\,m^{-2}}$
work hardening modulus	$\mathcal{N}$	0.71
plastic rate dependency parameter	$\mathcal{C}$	0.024
temperature dependency parameter	$\mathcal{M}$	1.56
reference plastic rate	$\dot{\varepsilon}_0^{pl}$	$4 \times 10^{-2}\,\mathrm{s^{-1}}$
reference temperature	θ_0	$293.15\,\mathrm{K}$
melting temperature	θ_m	$874.15\,\mathrm{K}$
specific heat capacity	c_θ	$862\,\mathrm{J\,kg^{-1}\,K^{-1}}$
heat expansion coefficient	α_σ	$23.6 \times 10^{-6}\,\mathrm{K^{-1}}$

Table 12.1: Aluminum AL 7075 T6, taken from Hassanifard et al. [55].

	dx/ dy / dz	`NoMP`	nx	ny	$\tilde{d}_x/\tilde{d}_y$
S0	$1 \times 10^{-4}\,\mathrm{m}$	10800	180	60	
S1	$6.25 \times 10^{-5}\,\mathrm{m}$	30000	300	100	
S2	$5 \times 10^{-5}\,\mathrm{m}$	120000	600	200	
G0	$1 \times 10^{-4}\,\mathrm{m}$	10800	180	60	dx/2
G1	$6.25 \times 10^{-5}\,\mathrm{m}$	30000	300	100	dx/2
G2	$5 \times 10^{-5}\,\mathrm{m}$	120000	600	200	dx/2
G0 100	$1 \times 10^{-4}\,\mathrm{m}$	30000	300	100	dx/2
G0 200	$1 \times 10^{-4}\,\mathrm{m}$	120000	600	200	dx/2
G1 60	$6.25 \times 10^{-5}\,\mathrm{m}$	10800	180	60	dx/2
G1 200	$6.25 \times 10^{-5}\,\mathrm{m}$	120000	600	200	dx/2

Table 12.2: Listing of MPM discretizations used in the vertical cutting simulation. The sets starting with a "G" are used in the context of a grid-shift with a maximum shift of the CBG's origin by $\tilde{d}_x/\tilde{d}_y$.

feed of 100% of the tool.

However, certain numerical instabilities can be observed as the simulations abort too early. The explicit analysis has no elaborate abort criterion but experiencing an overflow in the mapping routines. In simple words, the code aborts as a material point travel an extreme amount which is usually the case as unphysical stress concentrations lead to extreme accelerations.

Observing the workpiece morphology in the cutting plane shown in figure 12.2 indicates problems of the MPM analysis. An unphysical void is formed just below the tip of the tool. Besides this obvious issue, the shown von Mises stress distribution gives a hint to another phenomenon. In this rather coarse S0 discretization, the stress pattern shows a clear mesh dependency. The stress distribution shows gradients exactly at the cell-boundaries of the used CBG, especially on the left side of the cut. This pattern is, of course, expected to a certain degree, as stresses are updated using gradients provided by the grid. These gradients, in return, are constant per cell of the CBG in the present implementation. While this phenomenon might be easy to compensate with using a smaller mesh size, hence using a finer resolution, a more problematic situation can be observed on

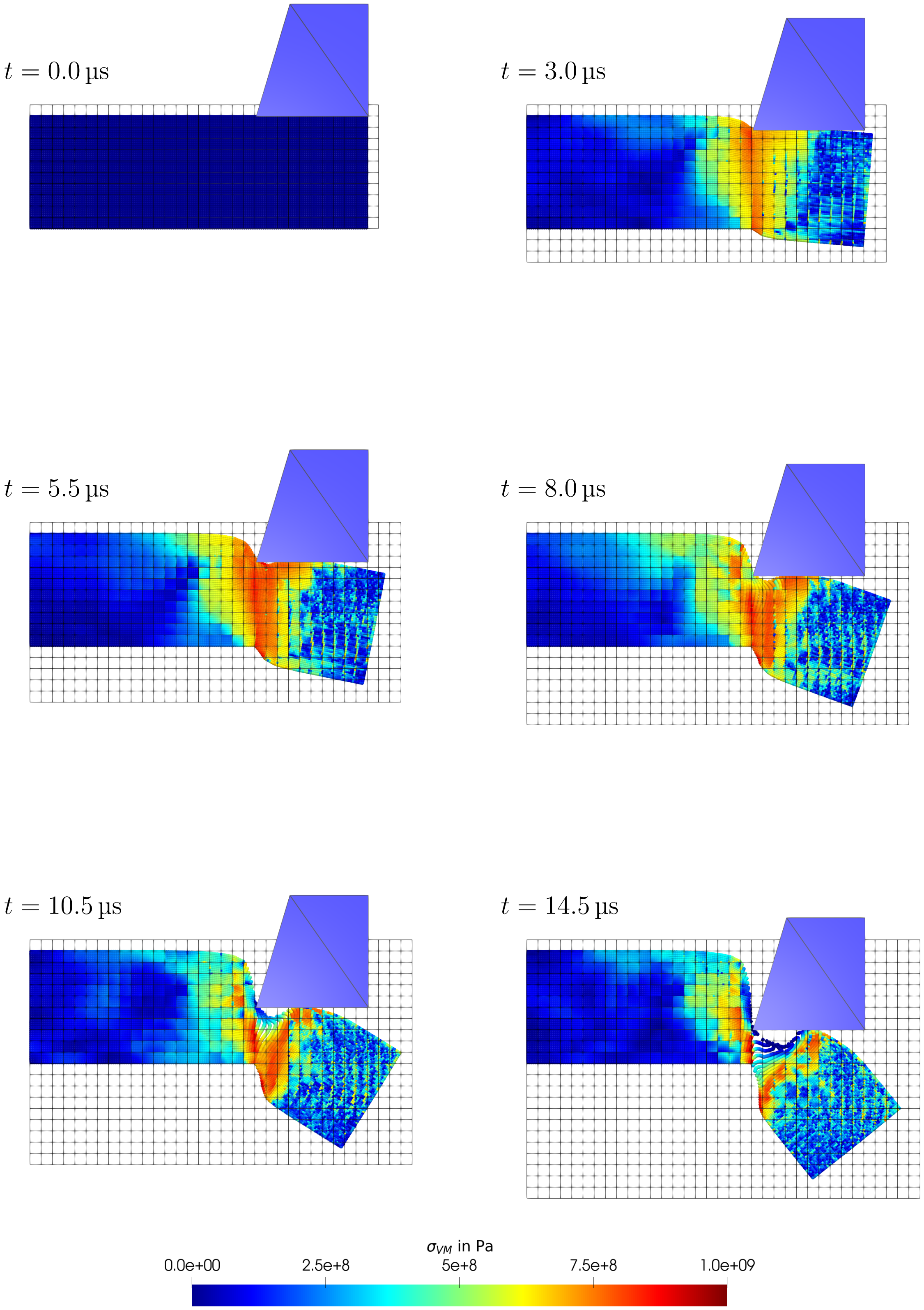


Figure 12.2: Time series of simulation snapshots for S0 discretization. The CBG geometry is reflected by the von Mises stress distribution, indicating a mesh dependency. Further, an unphysical void underneath the tool can be observed in later stages of the simulation.

the right-hand side of the cut. At a closer inspection, stress concentrations can be found at material points in cell corners and along edges. It is natural to assume stress waves inside the material propagating away from the location of the cut. These waves, however, cannot be appropriately resolved by the present discretization.

The discussed problems are still present at finer discretizations. Figure 12.6 shows, inter alia, results from the S2 set. It can directly be seen that the stress distribution still reflects the geometry of the CBG to an extreme degree, the void is forming underneath the tool, and that the stress waves on the right hand side of the cut are poorly resolved and show stress concentrations at corners and along edges of cells.

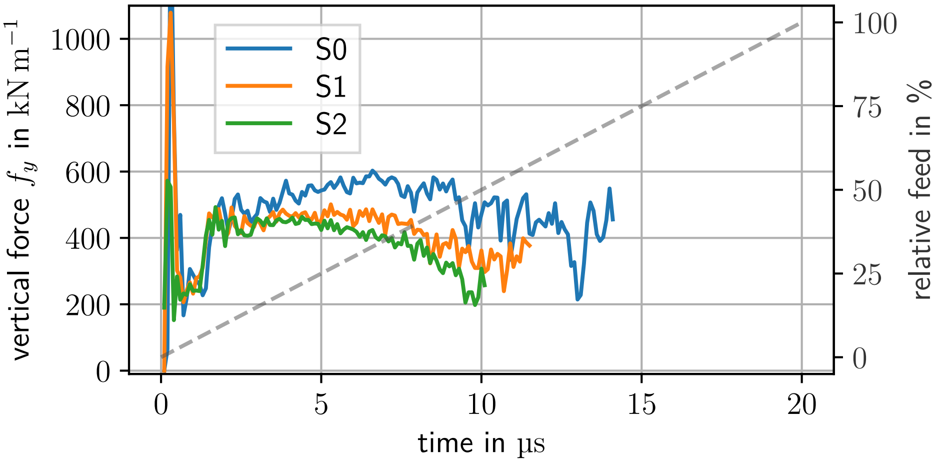


Figure 12.3: Vertical forces at the rigid tool for standard MPM discretizations at the vertical cutting problem. The simulations show some convergency in the force with finer discretizations but abort very early.

Using finer discretization as, e.g., S2, increases the simulation time, and leads to an earlier abortion, as shown in figure 12.3. It can directly be observed that the achieved relative feed of the tool decreases with higher discretizations. Jet, the S0 variant fails at a relative feed of $\approx$ 70%, which is also not satisfactory. Nevertheless, a convergence of the given vertical force over time can be observed. The results indicate that on the given scheme, a severe and computationally expensive resolution must be considered to achieve reliable results. On the other hand, this would decrease numerical stability, and hence it is hard to achieve completion on the given problem.

From the literature, these problems are known and usually addressed in using alternative MPM variants. Prevalent in the field of cutting simulations are GIMP methods, which introduce higher-order shape functions on the CBG. The main drawback of these approaches is the weaker performance. The computation of the mapping operators is more complex to implement and demands more computational effort at runtime. Also, the assembly of the system that depends on those mappings is more expensive. This problem is described, e.g., in Buzzi et al. [17]. To date, there is no better performance possible than using linear shape functions on regular cartesian-based CBGs.

Based on the knowledge that the grid geometry is reflected in material point's solution over time, the idea of changing the grid during the simulation is pursued in the following. This concept is generally covered in MPM theory, as, the arbitrary mesh can be used. For each individual time-step, the solutions are equally valid. As a result of this approach, the grid pattern does not accumulate on the material points from time-step to time-step. Thus, a change of the grid spacings, a rotation of the cartesian base of the grid and the translation of the grid's origin are possible while preserving the advantages of a regular cartesian grid layout. In the following, the third option is chosen exclusively. A graphical interpretation and implementation details are given in Appendix E. Usually, using different CBGs inside a single MPM simulation is not discussed in the literature. Nevertheless, some numerical codes optimize CBG's during the simulation following similar approaches as in `ELSE` code. However, in the following the grid-shift technique is used on purpose. The origin of the grid is translated in space, using a displacement vector based on random numbers, scaled by $\tilde{d}_x/\tilde{d}_y$. Notice that this displacement is re-computed for each time-step. Hence the CBG translates in each time-step as well.

By applying the grid-shift technique to the present vertical cutting problem, additional sets for the discretization are considered and given in table 12.2. Additionally, as the grid shifts, it is no longer guaranteed that grid boundary conditions work. Hence, the boundary conditions are applied directly on the material points in the vicinity of Γ_1 or Γ_2 in the form of spring-like forces, to compensate for it. This approach allows for the enforcement of, e.g., preservation of the initial position of a material point $\boldsymbol{X}_{\mathtt{MP}}$. The force is computed for each affected material point using

$$f^{BC}_{\mathtt{MP}} = \kappa^{BC} \; (x_{\mathtt{MP}})_i - (X_{\mathtt{MP}})_i \, , \tag{12.1}$$

and in each spatial direction i individually. Hereby, κ^{BC} in $\mathrm{kg\,s^{-2}}$ is an artificial stiffness parameter that has to be set problem-specific. The forces are assembled on the CBG analogue to a contact force, compare equation (9.2).

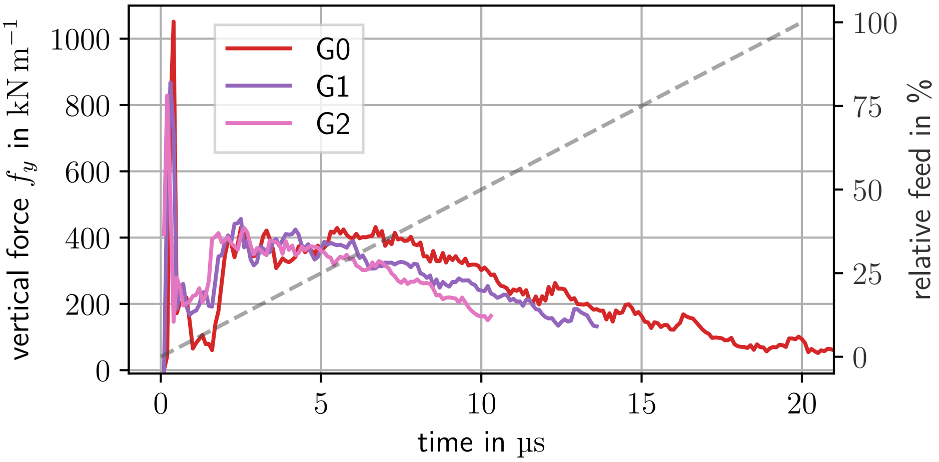


Figure 12.4: Vertical cutting forces over time, for three levels of discretization density using the grid-shift technique. The stability of the simulation enhances, G0 is able to perform the analysis beyond a relative feed of 100%. The results are much closer to each other than the same discretizations without grid-shift.

For the currently investigated vertical cutting problem, the spring forces, as mentioned in equation (12.1), are applied to Γ_1 for each material point with $(X_{\mathtt{MP}})_2 \leq 4 \times 10^{-5}\,\mathrm{m} \vee (X_{\mathtt{MP}})_1 \leq 2/3\, l_x$ in x and y-direction and to Γ_2 for each material point that meets $(X_{\mathtt{MP}})_1 \leq 4 \times 10^{-5}\,\mathrm{m}$ in y-direction only. In each case, the artificial stiffness is set to $\kappa^{BC} = 1 \times 10^7\,\mathrm{kg\,s^{-2}}$.

Performing the simulations with this modified setup allows for analysis beyond a relative feed of 100%. Visualizations of selected time steps can be found in figure 12.7. Notice that the different treatment of boundary conditions shows a clear influence on the simulation. However, this aspect is not investigated in detail in this example. Nevertheless, the results appear physically realistic. In contrast to the previous simulations, no void is forming underneath the tool. Also, the stress distribution within the workpiece is more smooth. Particularly in the first three stages illustrated in figure 12.7, the stress fields do not reflect the grid geometry as seen before. It is, however, noticeable that some stress concentrations are visible on the corners of the left hand side of the workpiece. Further in the analysis, the stresses localize in the cutting region, and stress waves can be observed as they propagate through the material. The stress waves appear in a good resolution, and no influence of the geometry of the used CBG can be observed.

Figure 12.6 shows a snapshot of the vertical cutting simulation at the dense discretizations S2 and G2 in direct comparison. Notice that the discretizations are equal, except for the application of the grid-shift technique and the alternative boundary conditions. The stress distribution on S2 shows the same mesh-dependency as observed in G0, even though its a higher resolution. This is even better visible in the temperature, which builds due to the plastic work. It can be assumed that the localization along the cutting edge is significantly driven by the geometry of the mesh. The grid-shift variant shows no signs of such mesh-dependency. The finer resolution of G2 in contrast to G0 resolves higher frequency stress waves, which is to be expected. The curved shape of the localization band that forms in the direction of the cut is more realistic, given the analysis in the literature, compare, e.g., Oñate et al. [109].

The cutting force curves obtained using grid-shift of the same discretization levels as investigated without it are displayed in figure 12.4. As mentioned before, the simulation stability is better using grid-shift, leading to higher values in the final relative feed. Another observation is that the curves are a lot more close to each other, indicating a reduced sensitivity with respect to the discretization density than without the grid-shift approach.

The analysis is carried out for varying numbers of material points at fixed grid spacings, in order to investigate the sensitivities of a grid-shift simulation with respect to the discretization further. The cutting force curves are given in figure 12.5. It can be concluded that 10800 material points can be already considered to produce a reliable result in the simulation. Also, the sensitivity with respect to grid spacing is very weak. Given that larger grid spacing speeds up the simulation and technically allows for larger time steps, this is a great advantage. Nevertheless, the trend that more dense discretizations destabilize the simulation remains.

To get a better idea of the impact of denser material point discretizations, figure 12.8 shows von Mises and temperature plots at time $t = 1.25 \times 10^{-5}\,\mathrm{s}$ of the simulation, for `NoMP` = 10800 (G0), `NoMP` = 30000 (G0 100), and `NoMP` = 120000 (G0 200) at the same grid spacing. Almost no difference can be seen, even in the direct comparison of these visualization. Is is slightly noticeable that the temperature, and hence plastic strain localization slightly more diffuse when using more material points.

In summary, the MPM implementation is capable of simulating the vertical cutting problem to a satisfying degree. By introducing the grid-shift technique, the simulations show better stability and complete physical behavior when compared to the literature. Mesh dependency of the results is avoided using a grid-shift, and even stress waves are resolved.

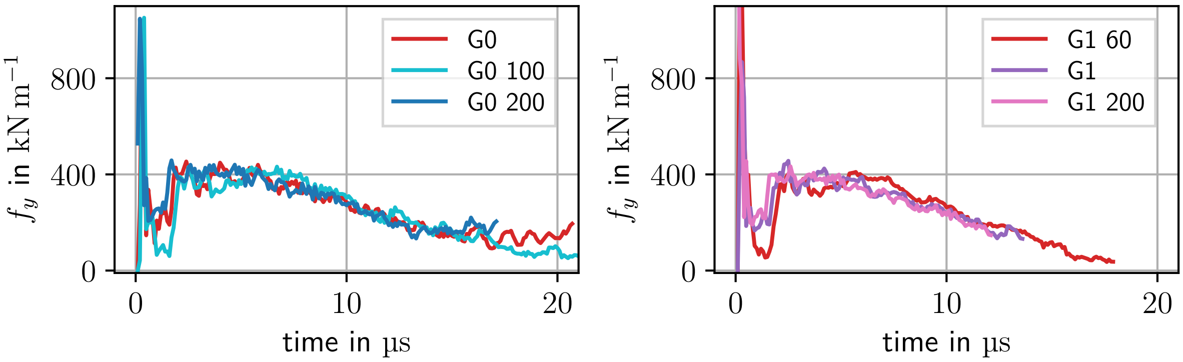


Figure 12.5: Vertical cutting force f_y over time. Comparison of G0 and G1 grid spacings with different number of material points. The grid-shift simulation show almost no sensitivity with respect to material point density.

Further, it was shown that the sensitivity of the simulation with respect to discretization density is weaker with the grid-shift ansatz. With the grid-shift technique relying on simple mapping operators, i.e., linear shape functions, the simulation quality can be increased dramatically without further computational expense compromises.

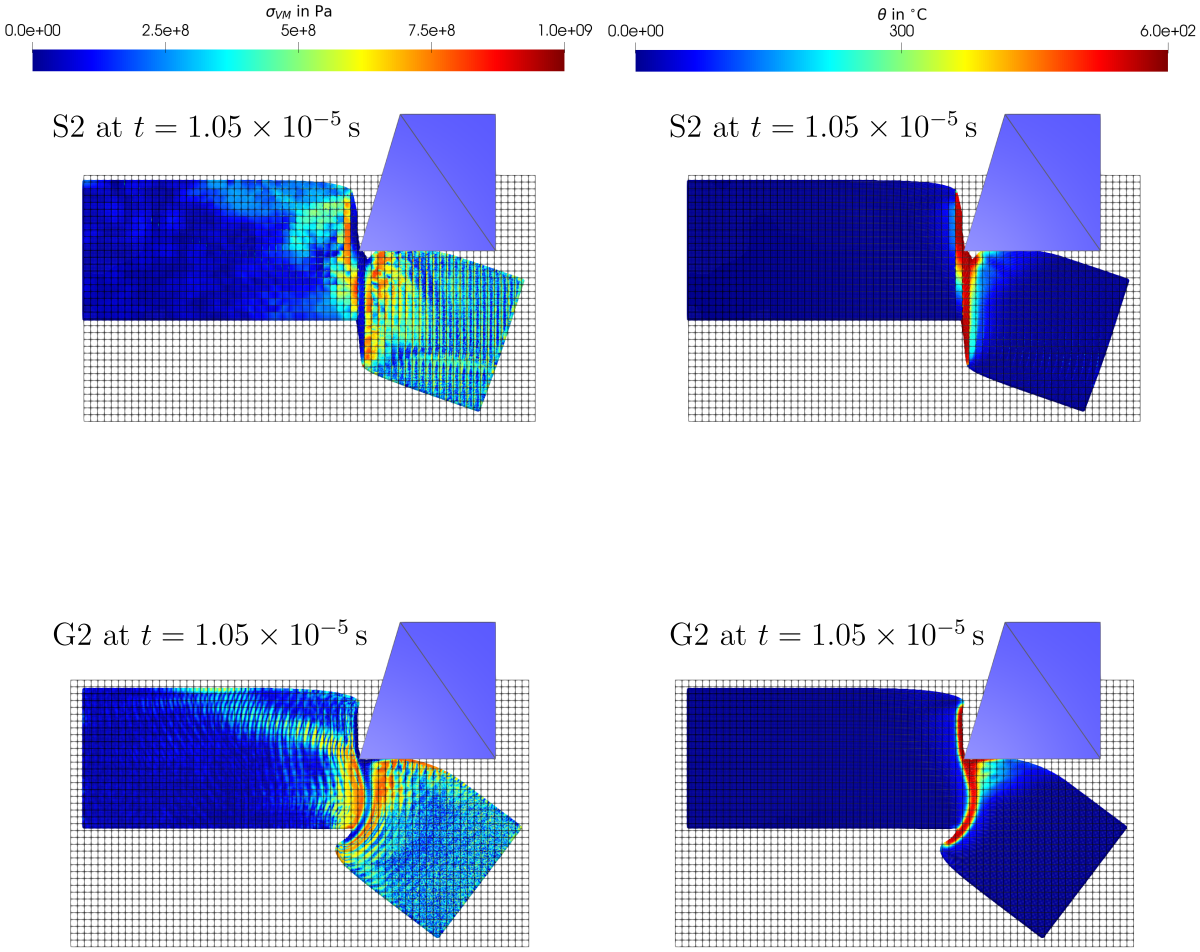


Figure 12.6: Direct comparison of fine discretizations S2 (top) and G2 (bottom), in stress and temperature distribution at the same time-step. The grid-shift simulation allows for a more physical localization. Further, in the grid-shift technique drastically reduces mesh-dependency of the results with respect to the CBG.

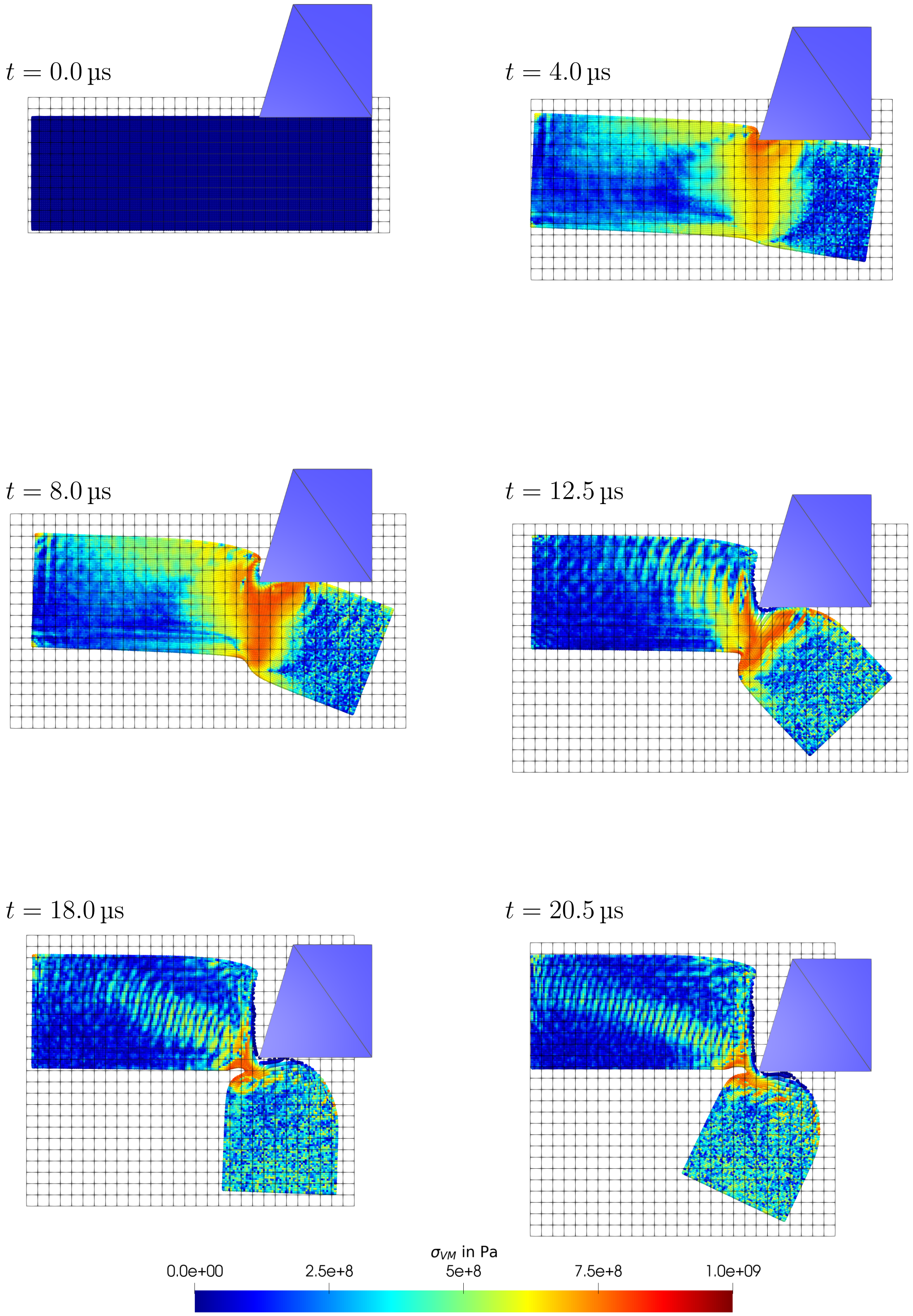
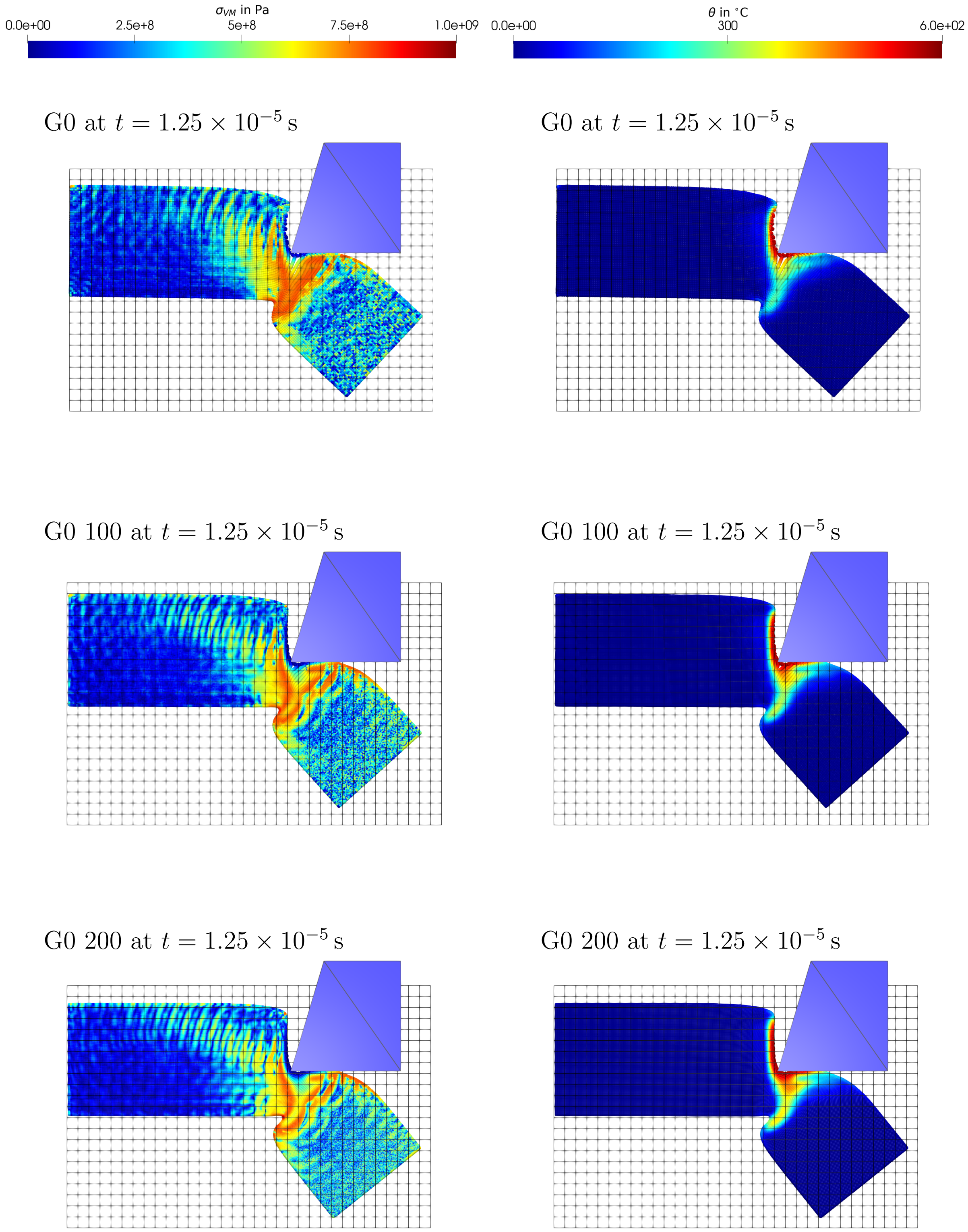


Figure 12.7: Vertical cutting simulation on the G0 discretization using the grid-shift technique. The displayed CBG changes for each time step. The simulation does not show the void below the tool, as seen without the grid-shift, compare figure 12.2. The resulting stress fields are more smooth and a remarkable resolution of stress waves propagating through the material is achieved.



G0 at $t = 1.25 \times 10^{-5}$ s

G0 at $t = 1.25 \times 10^{-5}$ s

G0 100 at $t = 1.25 \times 10^{-5}$ s

G0 100 at $t = 1.25 \times 10^{-5}$ s

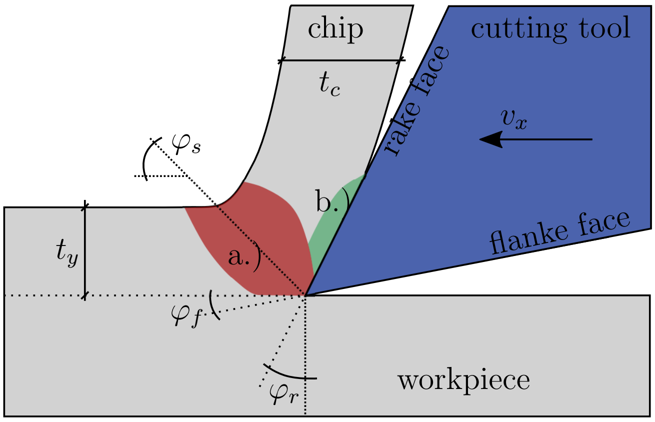
G0 200 at $t = 1.25 \times 10^{-5}$ s

G0 200 at $t = 1.25 \times 10^{-5}$ s

Figure 12.8: Visualizations of the vertical cut problem at $t = 1.25 \times 10^{-5}$ s. Shown are von Mises stress and temperature plots for G0 (top), G0 100 (middle), and G0 200 (bottom). Only minor differences are visible. The grid-shift technique reduces the sensitivity of MPM analysis with respect to discretization density. Further, no mesh-dependency can be identified.

12.3 Orthogonal cutting analysis

Metal cutting has been studied for a long time, by mechanics and metallurgists alike. The orthogonal cut represents the idealized cutting condition, which can be analyzed in a two-dimensional representation, illustrated in figure 12.9. However, its special conditions, orthogonal cutting covers the main aspect of metal cutting, such as chip formation, tool-workpiece interaction, cutting speeds and feeds, the influence of the tool geometry and temperature. It is extremely complicated or impossible to investigate any of these aspects individually in experiments. The temperature during cutting, for example, can not be identified in situ at the positions of interest, which are the primary and secondary shear zones.



φ_f	flanke angle
φ_r	rake angle
φ_s	shear angle
v_x	cutting velocity
t_y	cutting depth
t_c	chip thickness
a.)	primary shear zone (red)
b.)	secondary shear zone (green)

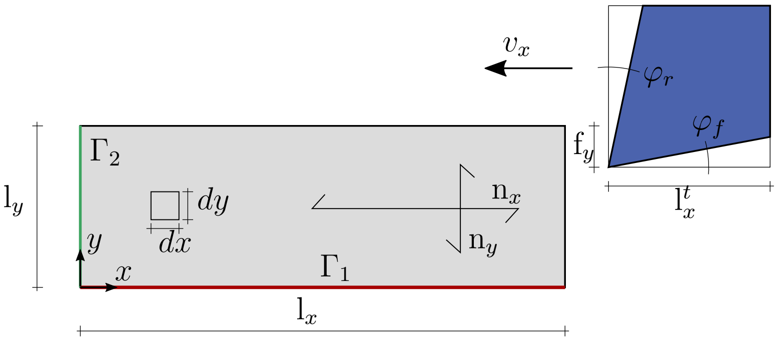
Figure 12.9: Orthogonal cutting geometric scheme and theoretical overview, compare, e.g., Childs et al. [28] or Rodríguez et al. [115].

These limitations stress the importance of numerical orthogonal cutting analysis. With the previously mentioned challenges of cutting simulations regarding available numerical techniques, many investigations of this problem can be found in the literature.

Two-dimensional finite element simulations can be found, for example, in Chen et al. [25] and Cheng and Mahnken [27], Gurusamy and Rao [49]. Here, the authors used a separation layer where the chip mesh separates from the remaining mesh covering the workpiece. In orthogonal cutting, the spatial location of this separation can be well estimated and hence this approach leads to reasonable results. However, proper separation criteria need to be applied. Additionally, mesh dependency can be observed regarding the separation and also for the formation of localizations, as shown in Hortig and Svendsen [60]. The Arbitrary Lagrangian Eulerian (ALE)-FEM can also be used. It has the advantage that the location of an occurring material discontinuity has not to be specifically implemented. ALE analysis of orthogonal cutting was done, e.g., in Pantalé et al. [111] and Wöste et al. [155]. Particle methods were used in orthogonal cutting simulation as well. Islam et al. [67] used the SPH on this problem, as well as Afrasiabi et al. [3], who used Graphics Processing Units (GPUs) to improve performance. A PFEM model for orthogonal cutting, even considering a 3D setup, can be found in Carbonell et al. [19]. Segregated chip formation based on ductile fracture criteria was analyzed in Huang et al. [63] using the Optimal Transportation Meshfree (OTM) method.

Quite early in its development, the standard MPM was used for orthogonal cutting simulations by Więckowski [154] and later by Nairn [103]. With the availability of GIMP, this MPM variant was used in Ambati et al. [7] and Gu et al. [48], with the latter one even considering thermal conductivity.

In the following, an orthogonal cutting setup is considered for simulation with the MPM implementation in `ELSE` and using the grid-shift technique. As workpiece material, the 42CrMo4 steel with parameters from table 11.1 is applied again. Similar to the vertical cutting analysis before, a rigid tool is assumed and the simulation is subject to plane strain conditions. A graphical illustration of the problem is given in 12.10, alongside some geometrical parameters.



l_x	$= 6 \times 10^{-3}\,\mathrm{m}$
l_y	$= 1 \times 10^{-3}\,\mathrm{m}$
φ_r	$= 7.5^\circ$
φ_f	$= 5.0^\circ$
l^t_x	$= 1 \times 10^{-3}\,\mathrm{m}$
l^t_y	$= 1 \times 10^{-3}\,\mathrm{m}$
v^t_x	$= -5 \times 10^{1}\,\mathrm{m\,s^{-1}}$

Figure 12.10: Orthogonal cutting problem illustration and geometrical simulation parameter.

Furthermore, the workpiece is discretized by ny material points in the y-direction and nx= 3ny material points in the x-direction. The material points located near a boundary are subject to treatment in the fashion of equation (12.1) to account for the boundary treatment. At Γ_1 for each material point with $(X_{\mathtt{MP}})_2 \leq 8 \times 10^{-5}\,\mathrm{m}$, the spring forces are applied in the x, and the y-direction, and for Γ_2, each material point that meets $(X_{\mathtt{MP}})_1 \leq 8 \times 10^{-5}\,\mathrm{m}$ in the y-direction only. In each case, the artificial stiffness is set to $\kappa^{BC} = 1 \times 10^{8}\,\mathrm{kg\,s^{-2}}$. The penalty parameter for the contact between material points and the tool is set to $\kappa_{\text{pen}} = 1 \times 10^{13}\,\mathrm{kg\,s^{-2}}$. While in the following different sets of parameters are investigated, the time step size is kept constant at $\Delta t = 2 \times 10^{-10}\,\mathrm{s}$.

As in the previous analyses, the sensitivity of the simulation to the discretization is investigated first. For this, the feed of the tool is kept to $\mathrm{f}_y = 0.3 \times 10^{-3}\,\mathrm{m}$. The considered spatial discretizations alter the number of material points for ny= 80, ny= 100, and ny= 150 resulting in a total `NoMP`= 19200, `NoMP`= 30000 and `NoMP`= 67500. We investigate the permutations of ny with three uniform grid spacings, i.e., dx=dy=$2 \times 10^{-4}\,\mathrm{m}$, dx=dy=$1 \times 10^{-4}\,\mathrm{m}$, and dx=dy=$8 \times 10^{-5}\,\mathrm{m}$. With a length of the workpiece $l_x = 6 \times 10^{-3}\,\mathrm{m}$, the simulation is intended to proceed until 80 micros which require 4×10^{5} time-steps. Snapshots of the cutting simulation at ny=80 and dx=dy=$1 \times 10^{-4}\,\mathrm{m}$ are displayed in figure 12.11. Before the results are studied in more detail, it is reported that the morphological influence of spatial discretization is almost not visible to the naked eye. The displayed snapshot series in figure 12.11 is representative. During the simulation, the tool moves to the left, cutting into the workpiece. In the contact region of the tool's tip and the workpiece, a chip is forming and evades the cutting zone along the tool's rake face. Further in the simulation, the formed chip curls counter-clockwise. The simulation intentionally terminates before self contact of the chip and the workpiece.

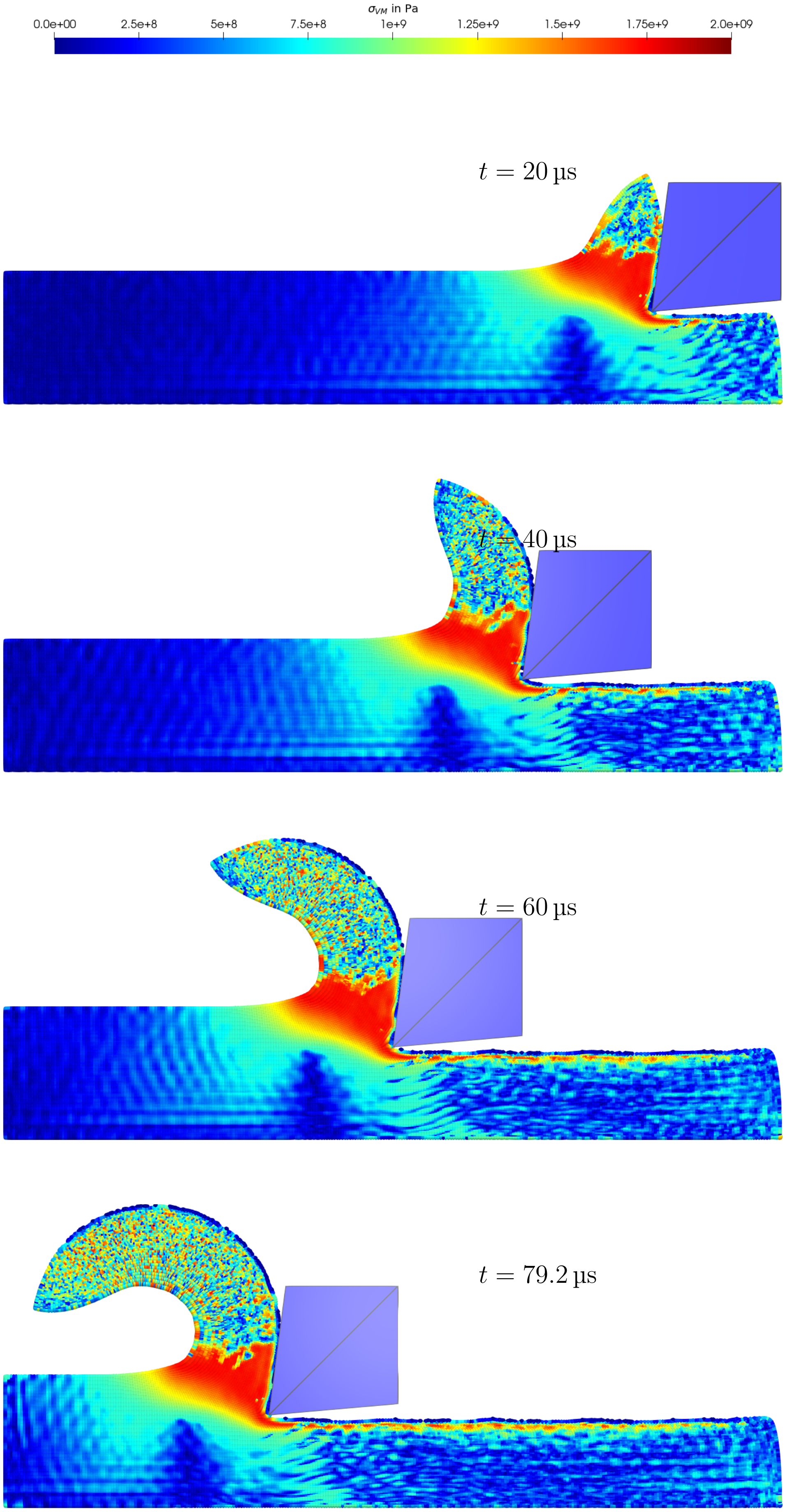


Figure 12.11: Time series of simulation snapshots in orthogonal cutting for dx=1×10^{-4} m and ny=100.

For a more analytical view, the cutting forces are investigated as well. In this example, and with the calibrated values for, e.g., the penalty stiffness at the contact, the recorded forces are very noisy. This phenomenon is known from experiments, where the signals are filtered for the sake of better analysis. Although noises might have different reasons, the following graphs show mainly filtered force curves. Nevertheless, the raw data are depicted as well to give an idea of the filter effects. The conclusions from the upcoming results are equally valid for the raw signals. They are just better to see on the filtered curves. Applied is a Savitzky-Golay, Least-Squares filter at a window size of 31 samples and using first-order approximation, compare, Savitzky and Golay [123]. The same technique was also used in the processing of experimental results on the SPHB example, as explained in Maassen et al. [94]. Figure 12.12 shows the cutting forces in the x and the y-direction for all considered ny, on the largest considered grid spacing dx=2×10^{-4} m.

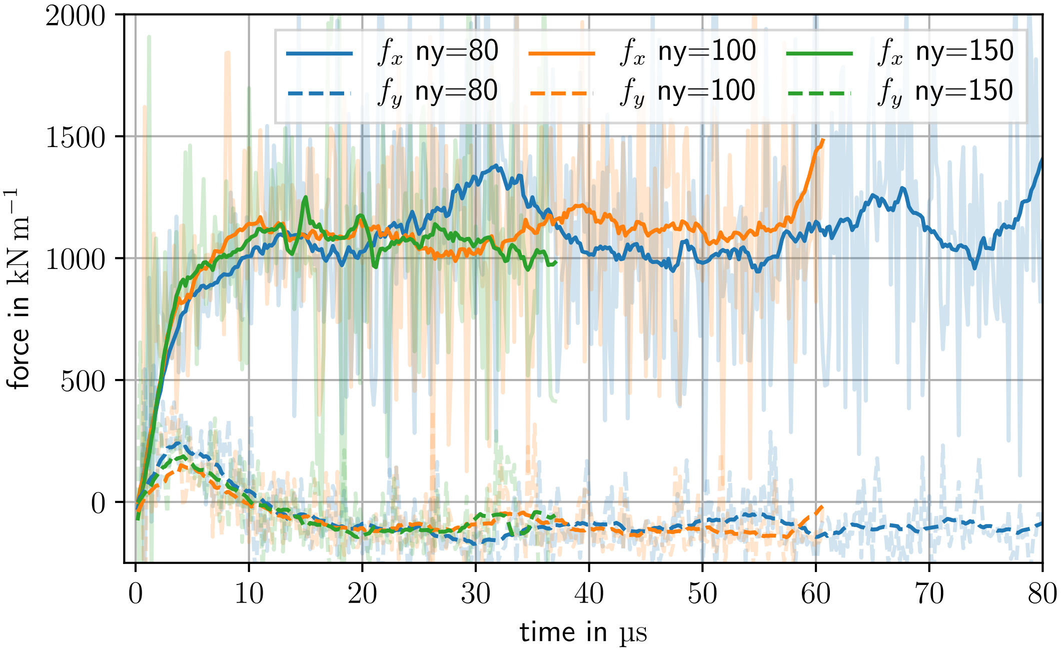


Figure 12.12: Orthogonal cutting forces f_x and f_y over time. Original raw data are shown with filtered data for dx=2×10^{-4} m.

The presented cutting forces present typical features known from physical and numerical analysis of orthogonal cutting. Both shown components reach an almost steady-state level after a certain period of time. For the considered setup, this period can be taken at approximately 15-20 µs. This impression is also backed by observations from the stress field in the simulations in the region of the cut. The vertical force component f_y goes to the negative regime, which is physical for the considered tool geometry. The horizontal component f_x is the major property for analyzing orthogonal cutting experiments and simulations. In the case of figure 12.12, the steady-state level lies above 1 MN m^{-1}. The considered discretization densities for material points produce almost identical force curves. Especially the horizontal components are very close and generally less subject to fluctuations. A possible explanation for the fluctuations observed in the force signals, especially before filtering, are stress waves that occur in the workpiece, following resonance frequencies. Naturally, these are weaker on the horizontal component, as the general force level is also low. The main difference between the investigated discretization densities is the duration until the simulation destabilizes. Again, the higher the material point density is, the earlier the simulation aborts. With respect to the force curves, however, there is no clear advantage in using more material points visible.

The effect of varying grid spacing can be investigated reviewing figures 12.13 and 12.14.

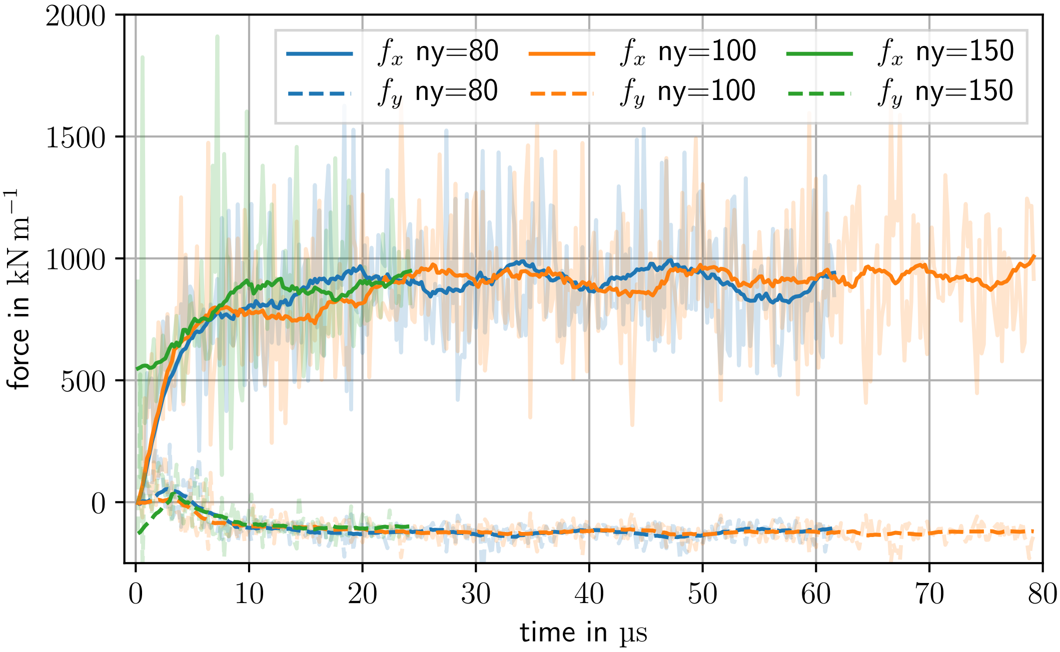


Figure 12.13: Orthogonal cutting forces f_x and f_y over time. Original raw data are shown with filtered data for dx=1×10^{-4} m.

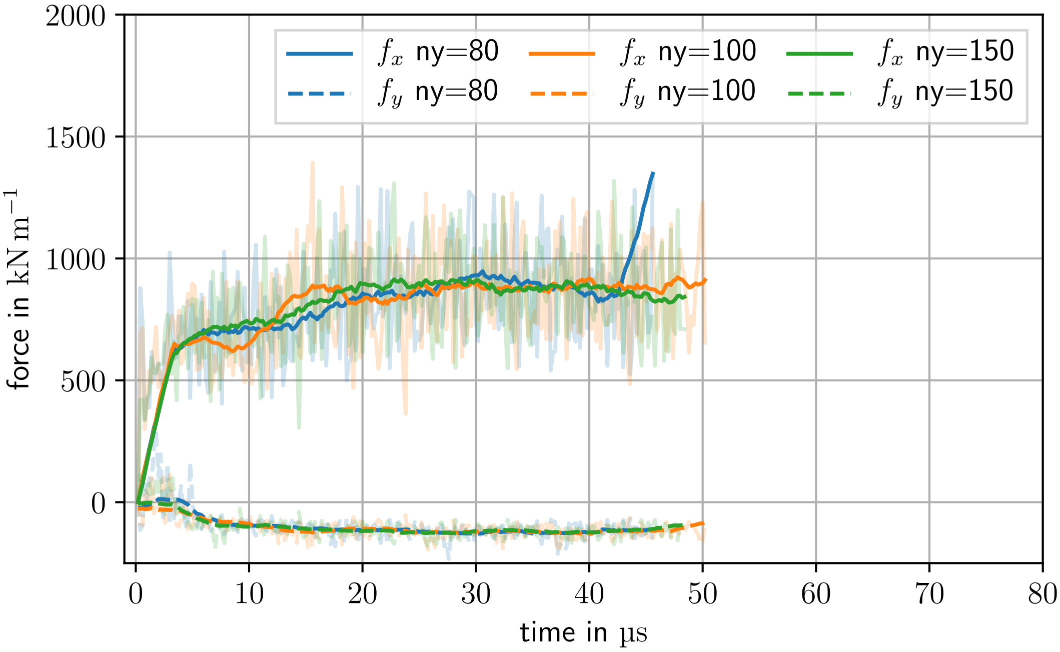


Figure 12.14: Orthogonal cutting forces f_x and f_y over time. Original raw data are shown with filtered data for dx=8×10^{-5} m.

In general, the same observations as made in figure 12.12 can be identified in these simulation results. For all considered discretization, the number of material points does not significantly impact on the measured force amplitudes. With smaller grid spacing, the force amplitudes show convergence towards approximately $9 \times 10^3\,\mathrm{kN\,m^{-1}}$. Also, the noise level on the raw data reduces at finer CBGs. This observation supports the theory that stress waves are a major reason for these noises in the numerical simulations, as they are better resolved on a finer CBG. Unfortunately, the results also show that smaller grid spacings generally lead to an early abortion of the simulation.

Concluding that the discretization sensitivity on this grid-shift orthogonal cutting analysis is low, phenomenological aspects are investigated. The question is whether the typical mechanisms of orthogonal cutting can be reproduced by the simulation. Figure 12.15 presents close-ups from the region of chip formation for the representative simulation of figure 12.11. Analyzing the von Mises stress distribution, shows that the primary shear zone is well developed. Also, the secondary shear zone can be identified along the contact surface between the forming chip and the tool. The outer layers of material points along this contact surface, however, show a stress gradient. This gradient lead to almost no stresses for material points in the contact area. The reason is that these material points also exhibit the largest plastic deformation, and consequently, strong thermal softening.

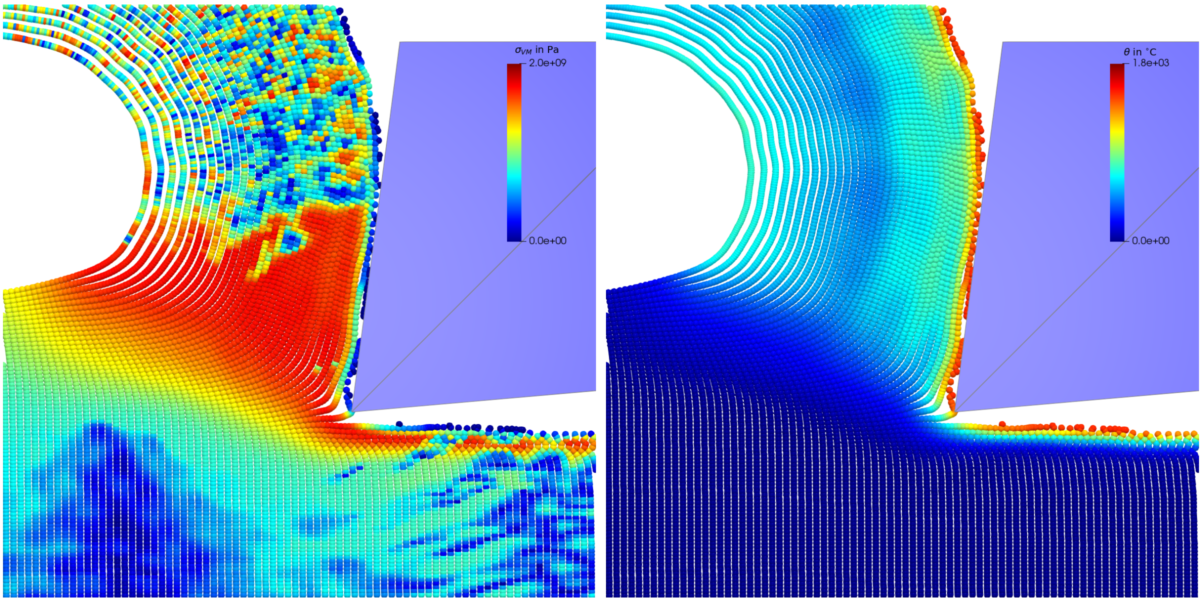


Figure 12.15: Close-up on simulation results in the region of chip formation. Displayed are the von Mises stress distribution (left) and the temperature distribution (right) of the workpiece, for ny= 100 and dx=1×10^{-4} m at $t = 79.2\,\mu\mathrm{s}$. In the stress plot, the primary and secondary shear zones (see figure 12.9) can be identified well.

The stress distribution within the chip is very chaotic. In total, the stresses within the chip are lower than in the chip formation zone, which is expected. Also, generally lower stress levels are visible in the workpiece underneath the primary shear zone. The concentrated stresses in the cutting area now propagate and affect larger reasons, leading to a reduced intensity. The propagation takes the form of dynamic stress waves, which are well resolved in the simulation. Another observation can be made in the region of the newly developed surface, where the tooltip has already been. Here, stress concentrations appear in an almost regular pattern near the surface. Again at the surface itself, the stresses are almost vanishing for the same reason as in the contact area.

The argument of huge plastic deformations and hence thermal softening can be further backed by reviewing the temperature distribution. With the considered adiabatic conditions, the temperature increase is only driven by plastic deformation. The higher temperatures, up to almost melting level, can be found where contact with the tool is taken or took place. Without any considered damage model, forming the material discontinuity at the cut induces these extreme plastic deformations. The accumulated temperature stagnates at the material points, which is expected due to the model assumption of no heat exchange. Within the chip, it can be observed that the temperatures are larger towards its boundaries and lowest in the middle of it. Of course, this pattern reflects the plastic deformation, which also causes the curling of the chip. Nevertheless, the temperatures are generally noticeably higher on the right-hand side of the chip, as displayed in figure 12.15, as the deformations there are higher, introduced by the secondary shear zone.

Furthermore, simulations are performed with varying feed depths of 0.1 mm, 0.3 mm, and 0.4 mm. The cutting forces for these are shown in figure 12.16. It is obvious that a larger cutting depth also causes higher cutting forces. For the considered set, the differences in magnitude are reasonable, given that the magnitude differences and feed differences are both about a factor of two. This trend can be confirmed in both horizontal force f_x and vertical force f_y.

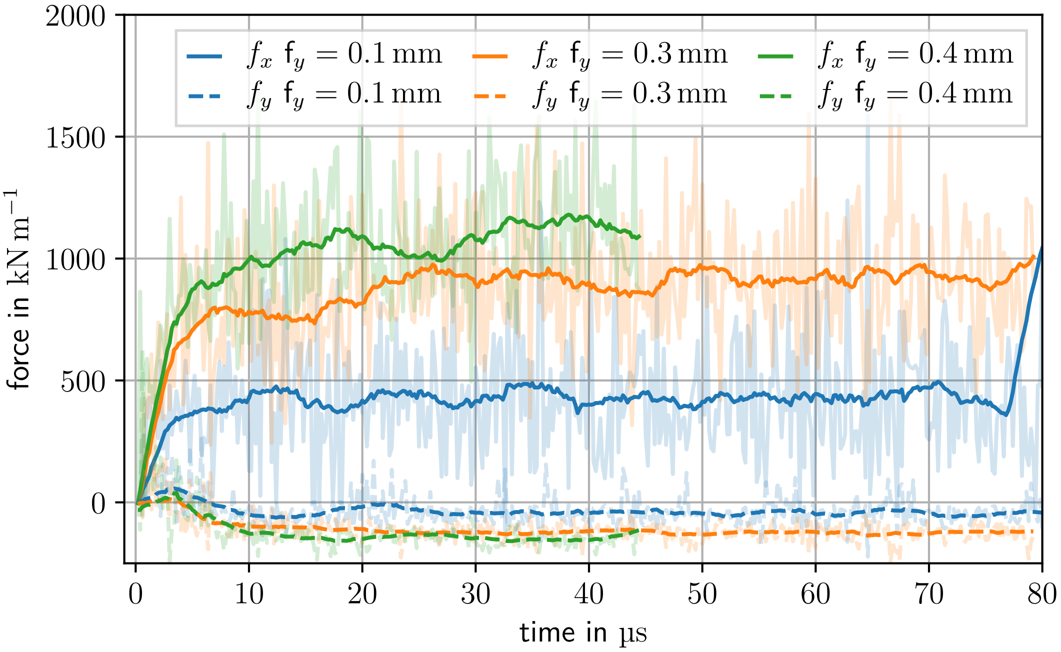


Figure 12.16: Orthogonal cutting forces f_x and f_y over time for varying feeds. Original raw data are shown with filtered data for varying feed f_y at dx=1×10^{-4} m.

As an interesting aspect, the shear angle φ_s should not be affected by the cutting depth f_y as it is mainly influence by the geometry of the tool, i.e., the rake angle φ_r. Figure 12.17 shows snapshots of the aforementioned simulations, again with a close-up to the chip forming region. Each figure is annotated with the cutting depth and an angle of 25°, which is approximately the shear angle throughout the simulations. The general pattern in the also displayed von Mises stress distribution is also preserved. The remaining stresses in the smallest chip are slightly higher, which might be due to its weaker resolution in material points compared to the other analyses. A noticeable difference between the chips is the curling angle. As the snapshots show exactly the same scale, the radius of the circle formed by the chip is larger with an increasing feed.

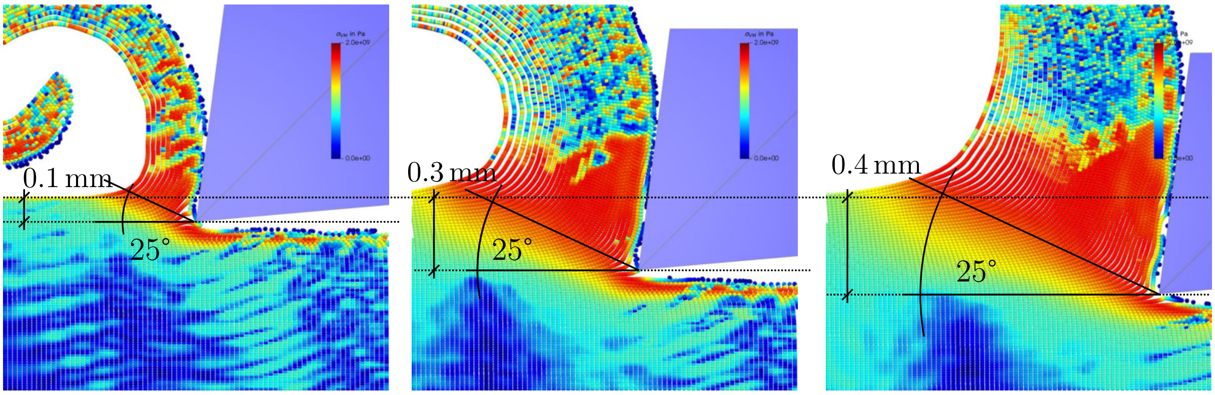


Figure 12.17: Close-ups of orthogonal cutting simulation with varying feed f_y. The discretization consists of ny= 100 at dx=1×10^{-4} m. The shear angle φ_s remains constant for all simulations.

A noticeable factor in machining is the contact conditions between tools and workpieces. More specifically, the friction conditions are known to have Hughe impacts on the temperature evolution in the cutting region, the wear of the tool, and also on the macroscopic cutting forces. Figure 12.18 shows the resulting force curves for the initial simulation but using different friction coefficients.

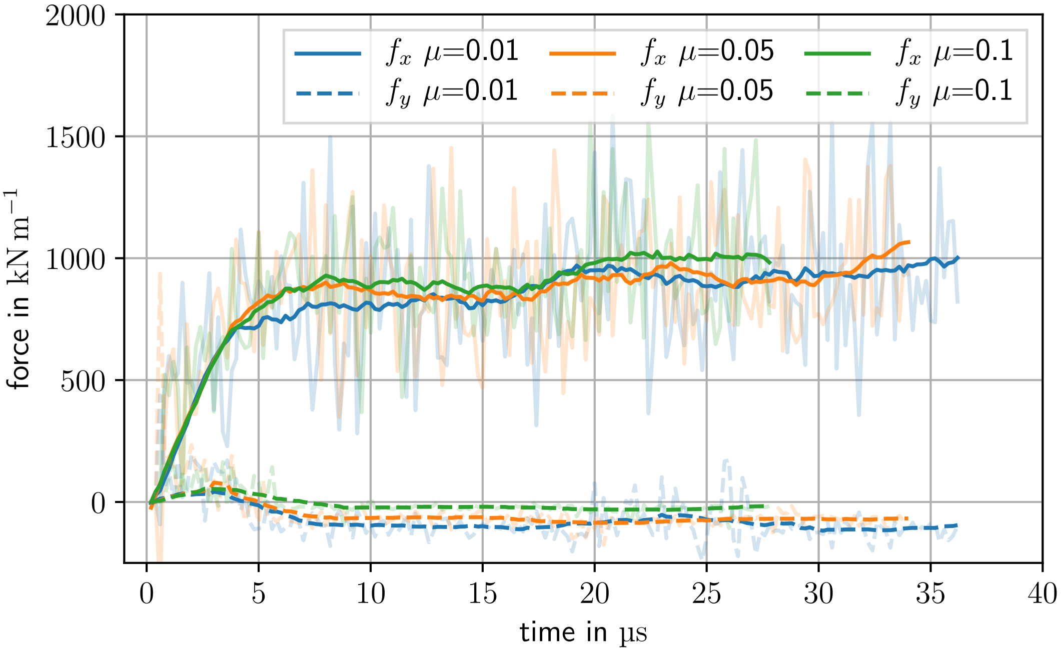


Figure 12.18: Orthogonal cutting forces f_x and f_y over time for varying friction coefficients between the tool and the workpiece. Original raw data are shown with filtered data for varying friction coefficients at dx=1×10^{-4} m.

It can be noticed that the horizontal cutting forces are not affected too much. Yet, the vertical force component shows a clear trend of larger forces. During the up-movement of the forming chip, it exchanges now tangential forces with the tool at the rake face. As the chip needs to move upwards, it drags the tool with it. This effect is. of course, larger with higher friction coefficients, making the simulation results reasonable.

As the last analysis on this problem, we investigate the deterministic character of the results produced with the implemented grid-shift technique. The implementation of the grid-shift technique used in this thesis relies on random numbers for the shift of the grid's origin in each time-step, as described in the appendix section E. Hence, the simulations include a stochastic element, leading to slightly different results on repeated analysis with the same parameters. To analyze this effect, an orthogonal cutting simulation was repeated five times, and the horizontal forces curves of these samples are compared. Raw, and filtered data are shown in figure 12.19. Considering that the data feature a certain noise level, it can be concluded that the five simulations lead to the same cutting forces and are equally stable. A comparison of the morphology of the workpieces also shows no major differences.

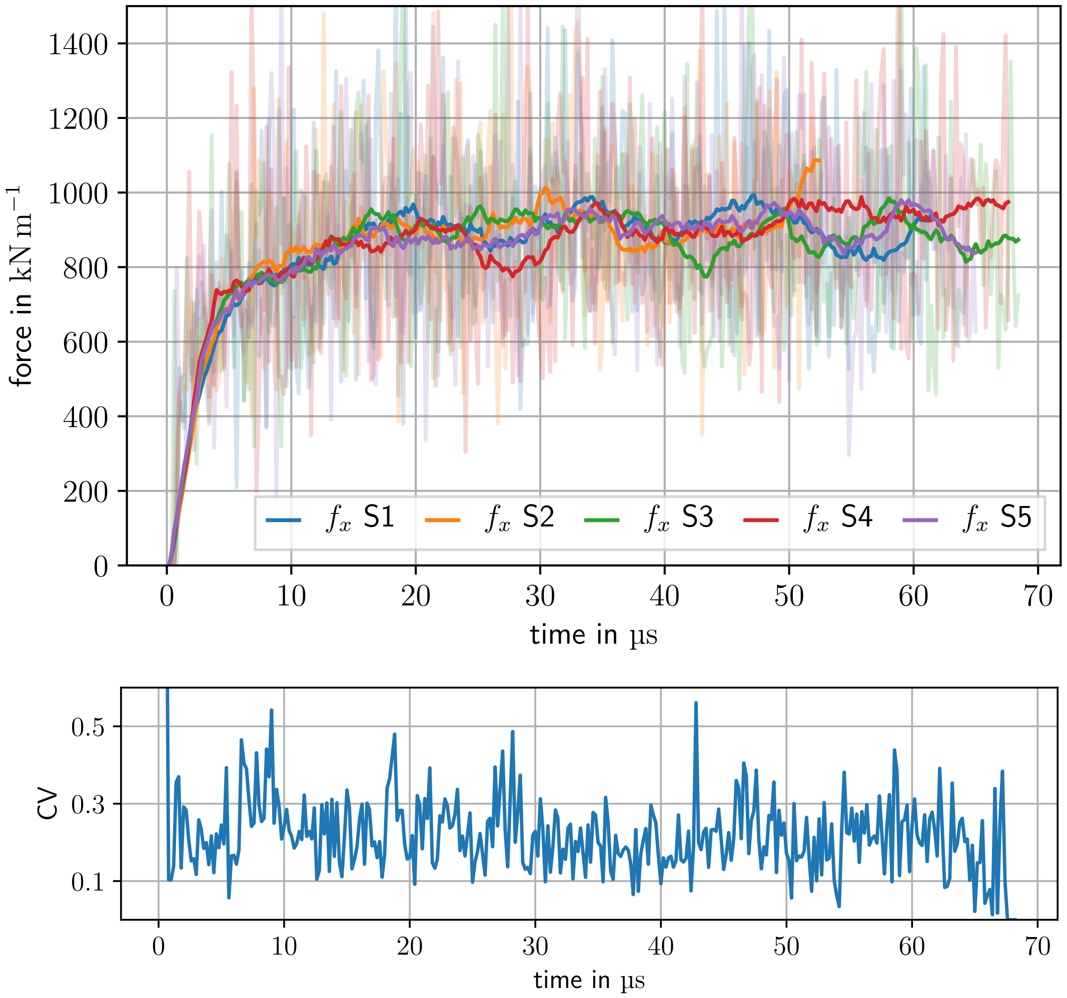


Figure 12.19: Orthogonal cutting force f_x over time for a repeated analysis with ny=80. Original raw data are shown with filtered data for dx=1×10^{-4} m and the coefficient of variation for the 5 samples of raw data.

The coefficient of variation (CV) is considered the mathematical way to quantify the differences between the sampled force curves. It describes the standard deviation of all samples from their mean value at each time step. Such an analysis would actually require a ten times larger set of samples. However, the computed CV can be taken as a hint towards the trend of the evolution of the CV using more samples. The CV for this set is shown in figure 12.19 as well. While it is generally not straightforward to interpret the CV, values below 0.3 are considered acceptable to low for many engineering applications. These data imply that grid-shift using a random number pattern over multiple time steps produces reliable results, with variations in an acceptable range.

13 Conlcusion and Outlook

In this thesis, an in-depth introduction to the Material Point Method (MPM) was given. After the introduction of the required continuum mechanical background, the utilized material laws and time discretizations were presented. These materials include a (JC) Johnson-Cook thermo-visco elasto-plastic matrerial for state of the art metal simulations. The balance of momentum was discretized by means of a Galerkin weak-form and discretized with the MPM. An implementation of the MPM into the Eulerian Lagrangian Simulation Environment (ELSE) code was discussed, and some software design aspects were outlined.

The present ELSE implementation was carefully benchmarked using standard examples and extensions of those. The implementation relies on an explicit solution scheme, using a Leapfrog time integration algorithm. It was shown that this scheme also allows for path-independent steady-state simulations of a Cook's membrane problem. The MPM results were matched with those from the literature using the Finite Element Method (FEM). Included were also contact mechanics with a focus on standard MPM contact and contact between MPM and rigid bodies. The MPM to rigid body contact establishes a Coulomb frictional contact based on a penalty force and passed an inclined plane test under dynamic conditions. Additionally, a Convected Particle Domain Interpolation (CPDI) technique was implemented and applied in both 2D and 3D. The 3D case exposed a critical failure criterion for CPDI schemes where boundary conditions are exposed on the Computational Background Grid (CBG).

Using the ELSE code, dynamic metal processes were investigated successfully. The simulation of a Split-Hopkinson-Pressure-Bar (SHPB) experiment was shown and could improved stability compared to the literature could be achieved by applying structured and hence more uniform discretization schemes on CPDI and MPM simulations. Further, the JC-material was fitted based on this example and in relation to experimental results for a 42CrMo4 steel.

A special focus of this work lies in the simulation of metal cutting processes. The complexity and challenges of this application push the established simulation techniques such as FEM to their limits, as outlined in chapter 12. The MPM was applied to these problems for its clear advantages of handling extreme deformations and material discontinuities as they occur in a cut.

The MPM was first used to simulate vertical cutting in a plane strain setup and considering an aluminum material. Motivated by the analysis, a novel grid-shift technique was introduced. The concept is to use a CBG with the same properties, i.e., grid-spacing, during the simulation but shift its origin slightly in each time step. It was shown that this increases the simulation's stability, reduces sensitivities with respect to discretization, and allows for a better resolution of dynamic features, such as stress waves, then compared to standard MPM. These benefits are achieved without additional computational costs and still use the standard and most effective tri-linear shape function on the CBG.

With orthogonal cutting representing the major benchmark problem for cutting simulations to date, the grid-shift technique was applied to this application. The benefits of grid-shift seen in the vertical cutting analysis could be confirmed on this problem as well. Further, the results obtained from the MPM on orthogonal cutting are precise and resolve all expected features. An analysis of discretization influences showed that a rather coarse discretization already produces good results. The influence of cutting depth towards the angle of the primary shear zone on the formation of the chip was studied and showed the required consistency over varying feeds. Also, the influence of friction showed physical behavior.

As an overall conclusion, the results of this thesis underline the strong points of the MPM as a reliable tool in numerical structural mechanics. Although some problems have been identified and discussed, the method produces reliable results under extreme conditions. Also the computational implementation is fairly straightforward, and the algorithms allow for performant simulations.

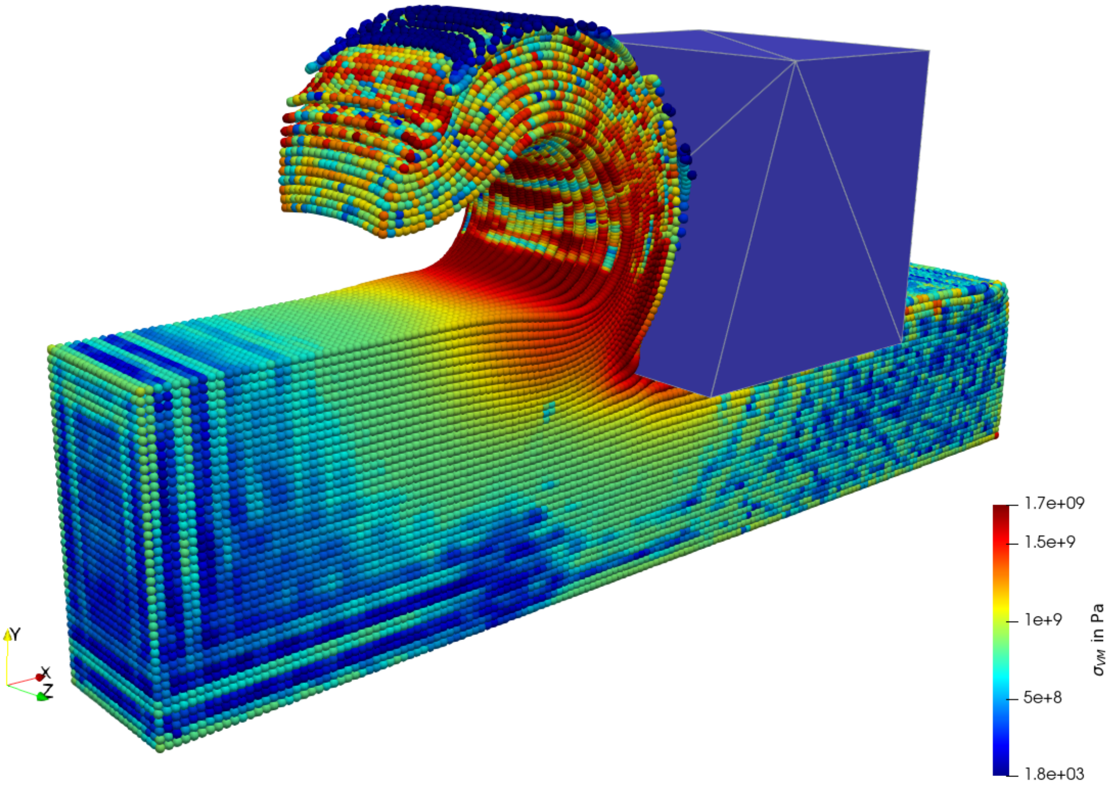


Figure 13.1: Example of a 3D orthogonal cutting MPM grid-shift analysis in ELSE.

The future working focus is to be set on the parallelization of the ELSE code. With rising computational demand in 3D analysis, this step is required to allow for simulations such as shown in figure 13.1 to be done in timescales common in research development focus and on higher resolutions. Subtractive manufacturing simulations, such as metal cutting, further require more complex material laws, which include damage in the future. The restriction of adiabatic conditions currently present in ELSE needs to be overcome by including heat conductivity, and the implementation for rigid body contact needs to be extended by non-convex shape for simulations of drilling and milling.

Research is required towards the stability enhancement of the explicit solution scheme in combination with MPM. Critical situations need to be identified which lead to simulation abortions and addresses by, e.g., partial damping strategies or material point splitting. The difficulty here is to ensure that such techniques are compatible with the dynamic, path-dependent problems in metal processing.

Adaptivity measures are another subject that could improve the simulation capabilities of MPM in the future. While some ideas for adaptive CBGs and material point splitting exist, their usage in highly dynamic processes with the need for resolving stress waves properly need to be investigated in detail.

The grid-shift technique offers a huge potential for improving MPM simulation without introducing additional complexity or computational expense. Still, interesting questions are to be answered regarding the type of shift. Future research could include rotations of the CBG's basis as well. The application of boundary conditions is probably the most cumbersome task when using the grid-shift technique. More smooth and intelligent ways must be developed to minimize the disturbances on a simulation introduced at regions with boundary conditions.

A Thesis notation on tensor calculus

In this chapter, a brief summary of the calculation rules of this thesis is given. In general, scalar quantities are denoted by italic characters, e.g., α, boldface-italic characters denote vectors or higher-order tensors, e.g., $\boldsymbol{a}$, $\boldsymbol{A}$. A vector, is defined with respect to a chosen basis, i.e., the orthonormal basis triad of the physical space and second-order tensors defined mapping operators for vectors

$$\boldsymbol{a} = a_i\,\boldsymbol{e}_i, \quad \boldsymbol{A} = A_{ij}\,\boldsymbol{e}_i \otimes \boldsymbol{e}_j. \tag{A.1}$$

For an orthonormal basis, the respective operations used in this work are:

$$\begin{array}{lll} \text{(diadic) vector product} & \boldsymbol{A} = \boldsymbol{a} \otimes \boldsymbol{b} & = a_i\,b_j\,\boldsymbol{e}_i \otimes \boldsymbol{e}_j \\ \text{single contraction} & \boldsymbol{a} = \boldsymbol{A} \cdot \boldsymbol{b} & = A_{ij}\,b_j\,\boldsymbol{e}_i, \\ & \boldsymbol{a} = \boldsymbol{A}^T \cdot \boldsymbol{b} & = A_{ji}\,b_j\,\boldsymbol{e}_i, \\ & a = \boldsymbol{a} \cdot \boldsymbol{b} & = a_i\,b_i, \\ \text{double contraction} & a = \boldsymbol{A} : \boldsymbol{B} & = A_{ij}\,B_{ij}. \end{array} \tag{A.2}$$

B Invariants

In this thesis, we use so-called principle I_i and main invariants J_i of second-order tensors, as defined, e.g., in Keip [74]. They are defined by scalar tensor function

$$\begin{aligned} &I_1 := \operatorname{tr}\boldsymbol{A}, \quad I_2 := \operatorname{tr}(\operatorname{cof}\boldsymbol{A}), \quad I_3 := \det\boldsymbol{A}. \\ &J_1 := \operatorname{tr}\boldsymbol{A}, \quad J_2 := \operatorname{tr}(\boldsymbol{A}\cdot\boldsymbol{A}), \quad J_3 := \operatorname{tr}(\boldsymbol{A}\cdot\boldsymbol{A}\cdot\boldsymbol{A}). \end{aligned} \tag{B.1}$$

Invariants are a priory invariant to rigid tensor rotations. Using, for example, the transformed left deformation tensor $\boldsymbol{b}^+$ as discussed in equation (4.2), we can state for the first and third invariant

$$\begin{aligned} \operatorname{tr}\boldsymbol{b}^+ &= \operatorname{tr}\big(\boldsymbol{Q}\cdot\boldsymbol{b}\cdot\boldsymbol{Q}^T\big) = \operatorname{tr}\big(\boldsymbol{Q}^T\cdot\boldsymbol{Q}\cdot\boldsymbol{b}\big) = \operatorname{tr}\boldsymbol{b} \quad \text{and} \\ \det\boldsymbol{b}^+ &= \det\big(\boldsymbol{Q}\cdot\boldsymbol{b}\cdot\boldsymbol{Q}^T\big) = \det\boldsymbol{Q}\,\det\boldsymbol{b}\,\det\boldsymbol{Q}^T = \det\boldsymbol{b}, \end{aligned} \tag{B.2}$$

by using $\boldsymbol{Q}^T \cdot \boldsymbol{Q} = \boldsymbol{I}$ and $\det\boldsymbol{Q} = 1$.

C von Mises stresses

The von Mises stress is used as a criterion for plastic deformations in ductile materials, such as metals. It depends on the second main invariant of the deviatoric stresses and hence is related to the non-volumetric deformation of a solid.

$$\sigma_{\mathtt{VM}} = \sqrt{\frac{3}{2}\,J_2(\operatorname{dev}\boldsymbol{\sigma})} = \sqrt{\frac{3}{2}\operatorname{dev}\boldsymbol{\sigma} : \operatorname{dev}\boldsymbol{\sigma}} = \sqrt{\frac{3}{2}}\,\|\operatorname{dev}\boldsymbol{\sigma}\| \tag{C.1}$$

Equally, component wise formulas can be derived, arriving at

$$\sigma_{\mathtt{VM}} = \sqrt{\sigma_{11}^2 + \sigma_{22}^2 + \sigma_{33}^2 - \sigma_{11}\,\sigma_{22} - \sigma_{11}\,\sigma_{33} - \sigma_{22}\,\sigma_{33} + 3\,(\sigma_{12}^2 + \sigma_{23}^2 + \sigma_{13}^2)}. \tag{C.2}$$

D Lie derivative of the elastic finger tensor

Lie derivatives are used to obtain objective rates in configurations other than the reference. The concept is to first relate the quantity to the reference configuration, perform the time differentiation, and at last a push-forward to the original configuration of the quantity. In the following, the Lie derivative of the elastic finger tensor (as used in chapter 4.3) is expressed in terms of the plastic deformation rate.

At first, the spatial quantity $\boldsymbol{b}^e = \boldsymbol{F}^e \cdot \boldsymbol{F}^{eT} = \boldsymbol{F} \cdot \boldsymbol{F}^{p-1} \cdot \boldsymbol{F}^{p-T} \cdot \boldsymbol{F}^T$ is related to the reference configuration, using a pull back, i.e.,

$$\boldsymbol{F}^{-1} \cdot \boldsymbol{b}^e \cdot \boldsymbol{F}^{-T} = \boldsymbol{F}^{-1} \cdot \boldsymbol{F} \cdot \boldsymbol{F}^{p-1} \cdot \boldsymbol{F}^{p-T} \cdot \boldsymbol{F}^T \cdot \boldsymbol{F}^{-T} = \boldsymbol{F}^{p-1} \cdot \boldsymbol{F}^{p-T} = \boldsymbol{C}^{p-1}. \tag{D.1}$$

In the next step, time differentiation is applied to the pull back quantity. With some algebraic manipulations, we arrive at

$$\begin{aligned} \frac{\partial}{\partial t}\left(\boldsymbol{F}^{p-1} \cdot \boldsymbol{F}^{p-T}\right) &= \boldsymbol{F}^{\dot{p}-1} \cdot \boldsymbol{F}^{p-T} + \boldsymbol{F}^{p-1} \cdot \boldsymbol{F}^{\dot{p}-T} \\ &= -\boldsymbol{F}^{p-1} \cdot \dot{\boldsymbol{F}}^{p} \cdot \boldsymbol{F}^{p-1} \cdot \boldsymbol{F}^{p-T} - \boldsymbol{F}^{p-1} \cdot \boldsymbol{F}^{p-T} \cdot \boldsymbol{F}^{\dot{p}-T} \cdot \boldsymbol{F}^{p-T} \end{aligned} \tag{D.2}$$

At last, the derived expression is pushed forward to the spatial configuration, respectively. This operation results by definition to the Lie derivative as

$$\begin{aligned} \pounds_{\boldsymbol{v}} \boldsymbol{b}^e &= \boldsymbol{F} \cdot \left(-\boldsymbol{F}^{p-1} \cdot \dot{\boldsymbol{F}}^{p} \cdot \boldsymbol{F}^{p-1} \cdot \boldsymbol{F}^{p-T} - \boldsymbol{F}^{p-1} \cdot \boldsymbol{F}^{p-T} \cdot \boldsymbol{F}^{\dot{p}-T} \cdot \boldsymbol{F}^{p-T}\right) \cdot \boldsymbol{F}^T \\ &= \boldsymbol{F}^e \cdot \boldsymbol{F}^p \cdot \left(-\boldsymbol{F}^{p-1} \cdot \dot{\boldsymbol{F}}^{p} \cdot \boldsymbol{F}^{p-1} \cdot \boldsymbol{F}^{p-T} - \boldsymbol{F}^{p-1} \cdot \boldsymbol{F}^{p-T} \cdot \boldsymbol{F}^{\dot{p}-T} \cdot \boldsymbol{F}^{p-T}\right) \cdot \boldsymbol{F}^{pT} \cdot \boldsymbol{F}^{eT} \\ &= \boldsymbol{F}^e \cdot \left(-\dot{\boldsymbol{F}}^{p} \cdot \boldsymbol{F}^{p-1} - \boldsymbol{F}^{p-T} \cdot \boldsymbol{F}^{\dot{p}-T}\right) \cdot \boldsymbol{F}^{eT} \\ &= \boldsymbol{F}^e \cdot \left(-\tilde{\boldsymbol{L}}^p - \tilde{\boldsymbol{L}}^{p^T}\right) \cdot \boldsymbol{F}^{eT} \\ &= -2\, \boldsymbol{F}^e \cdot \frac{1}{2}\left(\tilde{\boldsymbol{L}}^p + \tilde{\boldsymbol{L}}^{p^T}\right) \cdot \boldsymbol{F}^{eT} \end{aligned} \tag{D.3}$$

The symmetric part of the plastic velocity gradient can be interpreted as a plastic deformation rate, i.e., $\tilde{\boldsymbol{d}}^p = \frac{1}{2}\left(\tilde{\boldsymbol{L}}^p + \tilde{\boldsymbol{L}}^{p^T}\right)$, but defined with respect to the intermediate configuration

$$\pounds_{\boldsymbol{v}} \boldsymbol{b}^e = -2\, \boldsymbol{F}^e \cdot \tilde{\boldsymbol{d}}^p \cdot \boldsymbol{F}^{eT}. \tag{D.4}$$

E Regular computational background grid (CBG)

In this chapter an MPMGrid implementation for the `ELSE` code is presented. The specific implementation of this derived class (`RegularGrid_A`) renders a regular grid consisting of hexahedral cells. The regularity denotes that all cells are precisely parallelepipeds, with constant edge lengths dx, dy, and dz, with respect to a common, orthonormal grid bases triad. The main advantage of this design for a CBG to be used in an MPM simulation is that it is straightforward to compute the surrounding cell for a given coordinate.

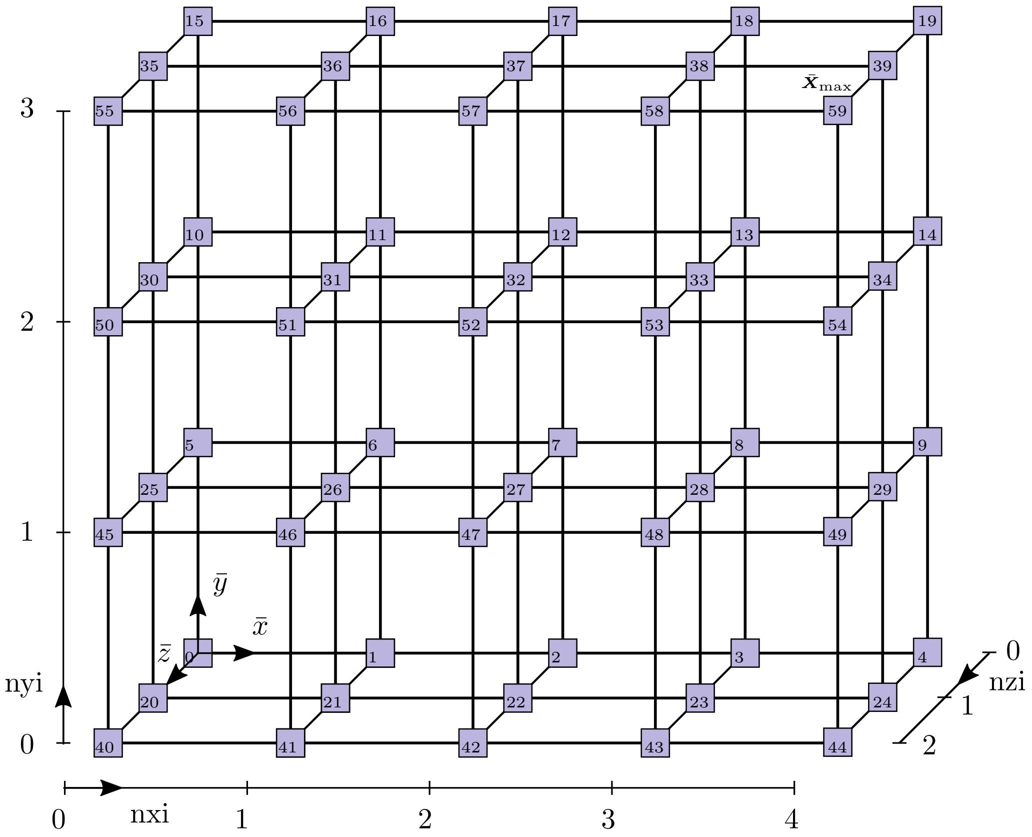


Figure E.1: Exemplary 3D regular grid of type `RegularGrid_A`, with 4x3x2 cells.

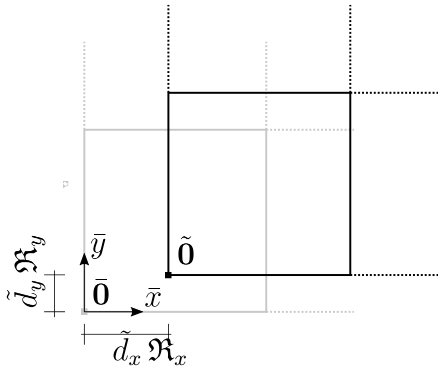


Figure E.2: Exemplary 2D illustration of a `RegularGrid_A`, subject to a grid-shift. The shape functions are computed from the shifted gird (black), which is obtained by a temporarily displacement applied to the initial definition of the grid (grey).

The construction of a regular grid in this thesis is based on a nodal index parametrization in nxi, nyi, nzi as depicted in figure E.1.

The global node numbers are acting as a unique identifier starting with 0. They are increasing, first in $\bar{x}$, then in $\bar{y}$, and lastly in the $\bar{z}$-direction. This notation makes it easy to compute nodal indexes efficiently.

The grid's extension is defined by the grid origin $\bar{\mathbf{0}} = (\bar{x}, \bar{y}, \bar{z})^T$ and a node with maximum distance from it $\underline{\bar{X}}_{\text{max}}$. Note that this is also the coordinate of the node in the grid with the highest global index. Notice that in the current implementation, the orthonormal basis of the CBG is equal to the orthonormal basis of physical space. Consequently, all edges of the grid align with a spatial direction. Furthermore, the cell division of the CBG is defined by maximum number of cells in each direction, i.e., `NoIx` in the x-direction, `NoIy` in the y-direction, and `NoIz` in the z-direction. Notice that the grids input in `ELSE` is the grids origin and the spacing parameters dx, dy, and dz. From this information, though, the other parameters can be computed. The grid has a function implemented to resize according to a set of material points to enclose them all. This way, the grid grows as needed, based on its fixed origin and the fixed spacing.

The main task of this CBG is to compute the enclosing cell for a given material point's current coordinate $\boldsymbol{x}_{\mathtt{MP}} = (x_{\mathtt{MP}}, y_{\mathtt{MP}}, z_{\mathtt{MP}})^T$. The enclosing cell is specified by its global node indexes `I` and global node coordinates $\boldsymbol{x}^{\mathtt{I}}$. For the grid shown in figure E.1, a cell consist of 8 nodes, with a scheme illustrated in figure (E.3).

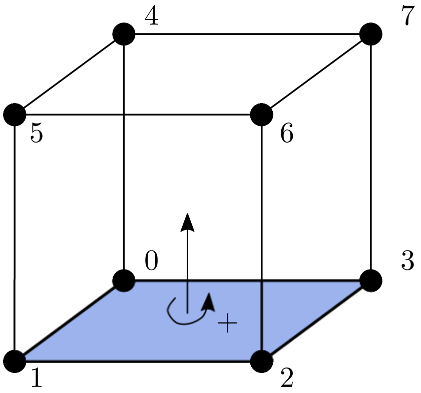


Figure E.3: A 3D 8-node hexahedral cell with local sequential node indexes, following right hand rule layout.

Both global grid node indexes I and coordinates $\boldsymbol{x}^I$ can be computed for the parametrization nxi, nyi, nzi. The parameterized representation of the 0th node of the enclosing cell for $\boldsymbol{x}_P$ is obtained via

$$\begin{aligned} \text{nxi} &= \text{floor}(x_{\mathtt{MP}}/dx), \\ \text{nyi} &= \text{floor}(y_{\mathtt{MP}}/dy), \\ \text{nzi} &= \text{floor}(z_{\mathtt{MP}}/dz). \end{aligned} \tag{E.1}$$

Nodal coordinates based on the parametrization are straightforward to compute via

$$\boldsymbol{x}^{\mathtt{I}} = (\text{nxi}\, dx, \text{nyi}\, dy, \text{nzi}\, dz)^T . \tag{E.2}$$

The global node number based on a parametrization can be computed with

$$\mathtt{I}(\text{nxi}, \text{nyi}, \text{nzi}) = \text{nzi}\,(\mathtt{NoIx}+1)\,(\mathtt{NoIy}+1) + \text{nyi}\,(\mathtt{NoIx}+1) + \text{nxi}. \tag{E.3}$$

Another advantage of this type of regular grids is that the grid basis functions can be computed with less computational effort. Based on the regular grid spacing, the nodal basis functions of CBG $\boldsymbol{N}$ for all nodes $\mathtt{I}$ of the surrounding cell for a material point at $\boldsymbol{x}_{\mathtt{MP}}$ can be computed directly, also derived with respect to the physical space.

following the sub sequential local order of figure E.3. Here the subscript $(\bullet)_{,x}$ denotes spatial derivative in the x-direction, $(\bullet)_{,y}$ in the y-direction and $(\bullet)_{,z}$ a spatial derivative in the z-direction, respectively.

$$\boldsymbol{N}(\boldsymbol{x}_{\mathtt{MP}}) = \begin{pmatrix} N^0 \\ N^1 \\ N^2 \\ N^3 \\ N^4 \\ N^5 \\ N^6 \\ N^7 \end{pmatrix} = \begin{pmatrix} a_0\,b_0\,c_0 \\ a_1\,b_0\,c_0 \\ a_1\,b_1\,c_0 \\ a_0\,b_1\,c_0 \\ a_0\,b_0\,c_1 \\ a_1\,b_0\,c_1 \\ a_1\,b_1\,c_1 \\ a_0\,b_1\,c_1 \end{pmatrix}. \tag{E.4}$$

The predictors introduced in equation (E.4) are computed based on the nodal coordinates of the cell, i.e.,

$$\begin{matrix} a_0 = \dfrac{x_{\max} - x_{\mathtt{MP}}}{dx} & a_1 = \dfrac{x_{\mathtt{MP}} - x_{\min}}{dx} \\ b_0 = \dfrac{y_{\max} - y_{\mathtt{MP}}}{dy} & b_1 = \dfrac{y_{\mathtt{MP}} - y_{\min}}{dy} \\ c_0 = \dfrac{z_{\max} - z_{\mathtt{MP}}}{dz} & c_1 = \dfrac{z_{\mathtt{MP}} - z_{\min}}{dz} \end{matrix} \quad \text{, with} \quad \begin{matrix} x_{\min} = x^0 & x_{\max} = x^1 \\ y_{\min} = y^0 & y_{\max} = y^3 \\ z_{\min} = z^0 & z_{\max} = z^5 \end{matrix} \ , \tag{E.5}$$

using the cells local node indexes, respectively. Also, computing the derivatives in this case is straightforward and does not require the chain rule

$$\boldsymbol{N}_{,x}(\boldsymbol{x}_{\mathtt{MP}}) = \begin{pmatrix} da_0\,b_0\,c_0 \\ da_1\,b_0\,c_0 \\ da_1\,b_1\,c_0 \\ da_0\,b_1\,c_0 \\ da_0\,b_0\,c_1 \\ da_1\,b_0\,c_1 \\ da_1\,b_1\,c_1 \\ da_0\,b_1\,c_1 \end{pmatrix}, \boldsymbol{N}_{,y}(\boldsymbol{x}_{\mathtt{MP}}) = \begin{pmatrix} a_0\,db_0\,c_0 \\ a_1\,db_0\,c_0 \\ a_1\,db_1\,c_0 \\ a_0\,db_1\,c_0 \\ a_0\,db_0\,c_1 \\ a_1\,db_0\,c_1 \\ a_1\,db_1\,c_1 \\ a_0\,db_1\,c_1 \end{pmatrix}, \boldsymbol{N}_{,z}(\boldsymbol{x}_{\mathtt{MP}}) = \begin{pmatrix} a_0\,b_0\,dc_0 \\ a_1\,b_0\,dc_0 \\ a_1\,b_1\,dc_0 \\ a_0\,b_1\,dc_0 \\ a_0\,b_0\,dc_1 \\ a_1\,b_0\,dc_1 \\ a_1\,b_1\,dc_1 \\ a_0\,b_1\,dc_1 \end{pmatrix}, \tag{E.6}$$

with the preliminary factors

$$\begin{aligned} da_0 &= -dx^{-1}, & db_0 &= -dy^{-1}, & dc_0 &= -dz^{-1}, \\ da_1 &= dx^{-1}, & db_1 &= dy^{-1}, & dc_1 &= dz^{-1}. \end{aligned} \tag{E.7}$$

The $\mathtt{RegularGrid\_A}$ implementation features a grid-shift technique. As stated before, the CBG expands from the grid origin $\bar{\mathbf{0}} = (\bar{x}, \bar{y}, \bar{z})^T$. The grid-shift technique displaces this origin in each direction for the current time step only. For the current time step, the origin $\tilde{\mathbf{0}}$ is used. The idea is to average out the displacements of the origin statistically at a large number of time-steps. In order to achieve this, each displacement component can be adjusted using multiplier $\tilde{d}_x$, $\tilde{d}_y$, $\tilde{d}_z$,

$$\tilde{\mathbf{0}} = \bar{\mathbf{0}} + \left(\tilde{d}_x\,\mathfrak{R}_x, \tilde{d}_y\,\mathfrak{R}_y, \tilde{d}_z\,\mathfrak{R}_z\right), \tag{E.8}$$

with a random number $\mathfrak{R}_i$ uniformly distributed on the interval $[-1, 1)$, distributed according to the probability density function $P(x|a, b) = \dfrac{1}{b-a}$. For the next time step, new random numbers are used, and the displacement is applied to the initial origin $\bar{\mathbf{0}}$. A 2D illustration of the grid-shift is given in figure E.2.

References

[1] B. Abali. *Computational Reality.* Springer, 2016.

[2] Afdahl, A. Jusuf, M. Kariem, and L. Gunawan. Development of a numerical model for simulations of Split Hopkinson Pressure Bar. *Journal of Engineering and Applied Sciences*, 11(10):6657–6662, 2016.

[3] M. Afrasiabi, L. Meier, M. Röthlin, H. Klippel, and K. Wegener. GPU-accelerated meshfree simulations for parameter identification of a friction model in metal machining. *International Journal of Mechanical Sciences*, 176:105571, jun 2020. doi: 10.1016/j.ijmecsci.2020.105571.

[4] M. Al-Mousawi, S. Reid, and W. Deans. The use of the split Hopkinson preasure bar techniques in high strain rate materials testing. *Proceedings of the Institution of Mechanical Engineers*, 211-C:273–292, 1997.

[5] B. Alder and T. Wainwright. Studies in Molecular Dynamics. I. General Method. *The Journal of Chemical Physics*, 31(2):459–466, Aug. 1959. doi: 10.1063/1.1730376.

[6] J. Amani, E. Oterkus, P. Areias, G. Zi, T. Nguyen-Thoi, and T. Rabczuk. A non-ordinary state-based peridynamics formulation for thermoplastic fracture. 87:83–94, 2016.

[7] R. Ambati, X. Pan, H. Yuan, and X. Zhang. Application of material point methods for cutting process simulations. *Computational Materials Science*, 57:102–110, 2012. doi: 10.1016/j.commatsci.2011.06.018.

[8] R. Aubry, S. Idelsohn, and E. Oñate. Particle finite element method in fluid-mechanics including thermal convection-diffusion. *Computers & Structures*, 83(17-18):1459–1475, June 2005. doi: 10.1016/j.compstruc.2004.10.021.

[9] B. Bagheri, M. Abbasi, A. Abdolahzadeh, and A. Kokabi. Numerical analysis of cooling and joining speed effects on friction stir welding by smoothed particle hydrodynamics (SPH). *Archive of Applied Mechanics*, 90(10):2275–2296, aug 2020. doi: 10.1007/s00419-020-01720-4.

[10] S. Bardenhagen. Energy Conservation Error in the Material Point Method for Solid Mechanics. *Journal of Computational Physics*, 180(1):383–403, jul 2002. doi: 10.1006/jcph.2002.7103.

[11] S. Bardenhagen and E. Kober. The generalized interpolation material point method. *Computer Modeling in Engineering and Sciences*, 5(6):477–495, 2004.

[12] J. Bergmann, J. Kimm, W. Theisen, and P. Wiederkehr. An approach to use subsurface characteristics for the prediction of process forces during cutting operations. *Procedia CIRP*, 88:276–281, 2020. doi: 10.1016/j.procir.2020.05.048.

[13] M. Berzins. Nonlinear stability and time step selection for the MPM method. *Computational Particle Mechanics*, 5(4):455–466, jan 2018. doi: 10.1007/s40571-018-0182-y.

[14] T. Bode, C. Weißenfels, and P. Wriggers. Mixed peridynamic formulations for compressible and incompressible finite deformations. *Computational Mechanics*, 65(5): 1365–1376, Feb. 2020. doi: 10.1007/s00466-020-01824-2.

[15] C. Bröcker. *Materialmodellierung für die simultane Kalt-/Warmumformung auf Basis erweiterter rheologischer Modelle*. PhD thesis, Insitute of Mechanics, University of Kassel, 2014.

[16] S. Butt and G. Meschke. Peridynamic analysis of dynamic fracture: influence of peridynamic horizon, dimensionality and specimen size. *Computational Mechanics*, 67(6):1719–1745, Apr. 2021. doi: 10.1007/s00466-021-02017-1.

[17] O. Buzzi, D. Pedroso, and A. Giacomini. Caveats on the implementation of the generalized material point method. *Computer Modeling in Engineering & Sciences*, 31(2):85–106, 2008. doi: 10.3970/cmes.2008.031.085.

[18] E. Campello. A description of rotations for DEM models of particle systems. *Computational Particle Mechanics*, 2(2):109–125, May 2015. doi: 10.1007/s40571-015-0041-z.

[19] J. M. Carbonell, J. M. Rodríguez, and E. Oñate. Modelling 3d metal cutting problems with the particle finite element method. *Computational Mechanics*, 66(3): 603–624, jun 2020. doi: 10.1007/s00466-020-01867-5.

[20] F. Ceccato and P. Simonini. Granular flow impact forces on protection structures: MPM numerical simulations with different constitutive models. *Procedia Engineering*, 158:164–169, 2016. doi: 10.1016/j.proeng.2016.08.423.

[21] A. Cecchi and N. Rizzi. Heterogeneous elastic solids: a mixed homogenization-rigidification technique. *International Journal of Solids and Structures*, 38:29–36, 2001.

[22] H. Chandrasekaran, R. M'Saoubi, and H. Chazal. Modelling of material flow stress in chip formation process from orthogonal milling and split hopkinson bar tests. *Machining Science and Technology*, 9(1):131–145, Mar. 2005. doi: 10.1081/mst-200051380.

[23] T. Charlton, W. Coombs, and C. Augarde. igimp: An implicit generalised interpolation material point method for large deformations. *Computers and Structures*, 190:108–125, 2017.

[24] M. Chaudry and P. Wriggers. On the computational aspects of comminution in discrete element method. *Computational Particle Mechanics*, 5(2):175–189, May 2017. doi: 10.1007/s40571-017-0161-8.

[25] G. Chen, C. Ren, x. Yang, and T. Guo. Evidence of thermoplastic instability about segmented chip formation process for ti–6al–4v alloy based on the finite-element method. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 225(6):1407–1417, May 2011. doi: 10.1177/0954406210396803.

[26] Z. Chen and R. Brannon. An Evaluation of the Material Point Method. Technical report, Feb. 2002. URL `https://doi.org/10.2172/793336`.

[27] C. Cheng and R. Mahnken. A multi-mechanism model for cutting simulation: A Ginzburg-Landau type phase gradient and numerical implementations. *International Journal of Solids and Structures*, 160:1–17, mar 2019. doi: 10.1016/j.ijsolstr.2018.10.007.

[28] T. Childs, K. Maekawa, T. Obikawa, and Y. Yamane. *Metal Machining.* Elsevier, 2000. doi: 10.1016/c2009-0-23990-0.

[29] S. Chowdhury, P. Roy, D. Roy, and J. Reddy. A modified peridynamics correspondence principle: Removal of zero-energy deformation and other implications. *Computer Methods in Applied Mechanics and Engineering*, 346:530–549, Apr. 2019. doi: 10.1016/j.cma.2018.11.025.

[30] C. Coetzee. *The Modelling of Granular Flow Using the Particle-inCell Method.* PhD thesis, Stellenbosch University, 2004.

[31] W. Coombs, T. Charlton, M. Cortis, and C. Augarde. Overcoming volumetric locking in material point methods. *Computer Methods in Applied Mechanics and Engineering*, 333:1–21, may 2018. doi: 10.1016/j.cma.2018.01.010.

[32] W. Coombs, C. Augarde, A. Brennan, M. Brown, T. Charlton, J. Knappett, J. Motlagh, and L. Wang. On Lagrangian mechanics and the implicit material point method for large deformation elasto-plasticity. *Computer Methods in Applied Mechanics and Engineering*, 358:112622, jan 2020. doi: 10.1016/j.cma.2019.112622.

[33] M. Cremonesi, A. Franci, S. Idelsohn, and E. Oñate. A State of the Art Review of the Particle Finite Element Method (PFEM). *Archives of Computational Methods in Engineering*, 27(5):1709–1735, aug 2020. doi: 10.1007/s11831-020-09468-4.

[34] P. Cundall and O. Strack. A discrete numerical model for granular assemblies. *Géotechnique*, 29(1):47–65, Mar. 1979. doi: 10.1680/geot.1979.29.1.47.

[35] A. de Vaucorbeil, V. Nguyen, and c. Nguyen-Thanh. Karamelo: an open source parallel C++ package for the material point method. *Computational Particle Mechanics*, 8(4):767–789, Oct. 2020. doi: 10.1007/s40571-020-00369-8.

[36] A. de Vaucorbeil, V. Nguyen, S. Sinaie, and J. Wu. Material point method after 25 years: Theory, implementation, and applications. In *Advances in Applied Mechanics*, pages 185–398. Elsevier, 2020.

[37] O. Ding and C. Schroeder. Penalty Force for Coupling Materials with Coulomb Friction. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, 2019. doi: 10.1109/tvcg.2019.2891591.

[38] Y. Dong and J. Grabe. Large scale parallelisation of the material point method with multiple GPUs. *Computers and Geotechnics*, 101:149 – 158, 2018. doi: https://doi.org/10.1016/j.compgeo.2018.04.001.

[39] H. Edelsbrunner and E. Mücke. Three-dimensional alpha shapes. In *Proceedings of the 1992 workshop on Volume visualization - VVS '92*. ACM Press, 1992. doi: 10.1145/147130.147153.

[40] A. Fernandes, H. Gomes, E. Campello, A. Müller, and P. Pimenta. A coupled FEM–DEM method for the modeling of fluids laden with particles. *Computational Particle Mechanics*, 8(2):349–368, June 2020. doi: 10.1007/s40571-020-00336-3.

[41] F. Fernández, E. Vargas, and R. Velloso. A 3d discretization procedure for the material point method (MPM). *Computational Particle Mechanics*, 7(4):725–733, dec 2019. doi: 10.1007/s40571-019-00303-7.

[42] J. Field, S. Walley, W. Proud, H. Goldrein, and C. Siviour. Review of experimental techniques for high rate deformation and shock studies. *International Journal of Impact Engineering*, 30(7):725–775, Aug. 2004. doi: 10.1016/j.ijimpeng.2004.03.005.

[43] J. Foster, S. Silling, and W. Chen. Viscoplasticity using peridynamics. *International Journal for Numerical Methods in Engineering*, 81:1242–1258, 2010.

[44] J.-P. Fürstenau, B. Avci, and P. Wriggers. A numerical review of multi-fluid SPH algorithms for high density ratios. In *Advances in Computational Fluid-Structure Interaction and Flow Simulation*, pages 139–150. Springer International Publishing, 2016. doi: 10.1007/978-3-319-40827-9_11.

[45] J.-P. Fürstenau, H. Wessels, C. Weißenfels, and P. Wriggers. Generating virtual process maps of SLM using powder-scale SPH simulations. *Computational Particle Mechanics*, 7(4):655–677, Nov. 2019. doi: 10.1007/s40571-019-00296-3.

[46] N. Galano, P. Moreno-Casas, and J. Abell. Extending the particle finite element method for sediment transport simulation. *Computer Methods in Applied Mechanics and Engineering*, 380:113772, July 2021. doi: 10.1016/j.cma.2021.113772.

[47] R. Gingold and J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181(3):375–389, Dec. 1977. doi: 10.1093/mnras/181.3.375.

[48] X. Gu, C. Dong, and T. Cheng. MPM simulations of high-speed machining of ti6al4v titanium alloy considering dynamic recrystallization phenomenon and thermal conductivity. *Applied Mathematical Modelling*, 56:517–538, apr 2018. doi: 10.1016/j.apm.2017.12.028.

[49] M. Gurusamy and B. Rao. A new constitutive relation for simulating plastic flow involving continuous-shear or shear-localisation during metal cutting. *Philosophical Magazine*, 100(4):486–511, Dec. 2019. doi: 10.1080/14786435.2019.1696023.

[50] J. Hallquist. *LS-DYNA Theory Manual.* Livermore Software Technology Corporation (LSTC), Berkeley, Livermore, 2006.

[51] C. Hammerquist and J. Nairn. A new method for material point method particle updates that reduces noise and enhances stability. *Computer Methods in Applied Mechanics and Engineering*, 318:724–738, may 2017. doi: 10.1016/j.cma.2017.01.035.

[52] F. Harlow. The particle-in-cell method for numerical solution of problems in fluid dynamics. Technical report, Mar. 1962.

[53] R. Hartley, T. J. Cloete, and G. Nurick. An experimental assessment of friction effects in the split hopkinson pressure bar using the ring compression test. *International Journal of Impact Engineering*, 34(10):1705–1728, oct 2007. doi: 10.1016/j.ijimpeng.2006.09.003.

[54] P. Hartmann, C. Weißenfels, and P. Wriggers. A curing model for the numerical simulation within additive manufacturing of soft polymers using peridynamics. *Computational Particle Mechanics*, 8(2):369–388, June 2020. doi: 10.1007/s40571-020-00337-2.

[55] S. Hassanifard, H. Mousavi, and A. Varvani-Farahani. The influence of low-plasticity burnishing process on the fatigue life of friction-stir-processed Al 7075-T6 samples. *Fatigue & Fracture of Engineering Materials & Structures*, 42(3):764–772, Nov. 2018. doi: 10.1111/ffe.12950.

[56] G. Holzapfel. *Nonlinear Solid Mechanics - A Continuum Approach for Engineering.* John Wiley and Sons, 2000.

[57] G. A. Holzapfel. *Nonlinear solid mechanics, a continuum approach for engineering.* Wiley, 2000.

[58] A. Hor, F. Morel, J.-L. Lebrun, and G. Germain. Modelling, identification and application of phenomenological constitutive laws over a large strain rate and temperature range. *Mechanics of Materials*, 64:91–110, Sept. 2013. doi: 10.1016/j.mechmat.2013.05.002. URL `https://doi.org/10.1016/j.mechmat.2013.05.002`.

[59] E. Hornbogen and H. Warlimont. *Metallkunde.* Springer Berlin Heidelberg, 1996.

[60] C. Hortig and B. Svendsen. Simulation of chip formation during high-speed cutting. *Journal of Materials Processing Technology*, 186(1-3):66–76, may 2007. doi: 10.1016/j.jmatprotec.2006.12.018.

[61] W. Hu, Y. Ha, and F. Bobaru. Peridynamic model for dynamic fracture in unidirectional fiber-reinforced composites. *Computer Methods in Applied Mechanics and Engineering*, 217:247–261, 2012.

[62] Y. Hu and E. Madenci. Bond-based peridynamics with an arbitrary poisson's ratio. In *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, Jan. 2016. doi: 10.2514/6.2016-1722.

[63] D. Huang, C. Weißenfels, and P. Wriggers. Modelling of serrated chip formation processes using the stabilized optimal transportation meshfree method. *International Journal of Mechanical Sciences*, 155:323–333, may 2019. doi: 10.1016/j.ijmecsci.2019.03.005.

[64] P. Huang, X. Zhang, S. Ma, and H. Wang. Shared Memory OpenMP Parallelization of Explicit MPM and Its Application to Hypervelocity Impact. *Computer Modeling in Engineering and Sciences*, 38, 12 2008.

[65] B. Hudobivnik and J. Korelc. Closed-form representation of matrix functions in the formulation of nonlinear material models. *Finite Elements in Analysis and Design*, 111:19–32, apr 2016. doi: 10.1016/j.finel.2015.12.002.

[66] S. Idelsohn, E. Oñate, and F. Pin. The particle finite element method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves. *International Journal for Numerical Methods in Engineering*, 61(7):964–989, Sept. 2004. doi: 10.1002/nme.1096.

[67] M. Islam, A. Bansal, and C. Peng. Numerical simulation of metal machining process with eulerian and total lagrangian SPH. *Engineering Analysis with Boundary Elements*, 117:269–283, aug 2020. doi: 10.1016/j.enganabound.2020.05.007.

[68] W. Jakob, J. Rhinelander, and D. Moldovan. pybind11 – Seamless operability between C++11 and Python, 2017. https://github.com/pybind/pybind11.

[69] A. Javili, A. McBride, J. Mergheim, and P. Steinmann. Towards elasto-plastic continuum-kinematics-inspired peridynamics. *Computer Methods in Applied Mechanics and Engineering*, 380:113809, July 2021. doi: 10.1016/j.cma.2021.113809.

[70] G. Johnson and W. Cook. A constitutive model and data for metals subjected to large strains, high strain rates and high temperatures. In *Proceedings of the 7th International Symposium on Ballistics*, volume 21, pages 541–547, 1983.

[71] G. Johnson and W. Cook. Fracture characteristics of three metals subjected to various strains, strain rates, temperatures and pressures. *Engineering Fracture Mechanics*, 21(1):31–48, 1985.

[72] G. Johnson and T. Holmquist. Evaluation of cylinder-impact test data for constitutive model constants. *Journal of Applied Physics*, 64(8):3901–3910, oct 1988. doi: 10.1063/1.341344.

[73] M. Kariem, R. Santiago, R. Govender, D. Shu, R. D., G. Nurick, M. Alves, G. Lu, and G. Langdon. Round-robin test of split hopkinson pressure bar. *International Journal of Impact Engineering*, 126:62–75, 2019. doi: 10.1016/j.ijimpeng.2018.12.003.

[74] M.-A. Keip. *Modeling of Electro-Mechanically Coupled Materials on Multiple Scales.* PhD thesis, University of Duisburg-Essen, 2012.

[75] J. Kimm, F. Pöhl, P. Wiederkehr, and W. Theisen. Subsurface Microstructure Evolution and Residual Stress in 42CrMo4 Steel after Single-Pass Pendulum Tests. *International Journal of Mechanical Engineering and Robotics Research*, pages 1072–1080, 2020. doi: 10.18178/ijmerr.9.8.1072-1080.

[76] J. Kimm, J. Bergmann, F. Wöste, F. Pöhl, P. Wiederkehr, and W. Theisen. Deformation behavior of 42CrMo4 over a wieder range of temperatures and strain rates: The effect of microstructure on the results of Split-Hopkinson pressure bar tests. *Material Science and Engineering: A*, (submitted), 2021.

[77] S. Klinkel. Theorie und Numerik eines Volumen-Schalen-Elementes bei finiten elastischen und plastischen Verzerrungen. *Dissertation thesis, Institut für Baustatik, Universität Karlsruhe*, 2000.

[78] J. Korelc. Automatic generation of finite-element code by simultaneous optimization of expressions. *Theoretical Computer Science*, 187(1):231–248, 1997.

[79] J. Korelc. Multi-language and Multi-environment Generation of Nonlinear Finite Element Codes. *Engineering with Computers*, 18:312–327, 2002.

[80] P. Koster, R. Tielen, E. Wobbes, and M. Möller. Extension of b-spline material point method for unstructured triangular grids using powell–sabin splines. *Computational Particle Mechanics*, mar 2020. doi: 10.1007/s40571-020-00328-3.

[81] S. Lahiri, K. Bhattacharya, A. Shaw, and L. Ramachandra. A stable SPH with adaptive B-spline kernel. *Journal of Computational Physics*, 422:109761, dec 2020. doi: 10.1016/j.jcp.2020.109761.

[82] X. Lei, S. He, and L. Wu. Stabilized generalized interpolation material point method for coupled hydro-mechanical problems. *Computational Particle Mechanics*, 8(4): 701–720, Oct. 2020. doi: 10.1007/s40571-020-00365-y.

[83] X. Li, Y. Xie, and M. Gutierrez. A soft–rigid contact model of MPM for granular flow impact on retaining structures. *Computational Particle Mechanics*, 5(4):529–537, feb 2018. doi: 10.1007/s40571-018-0188-5.

[84] Y. Lian, X. Zhang, X. Zhou, and Z. Ma. A FEMP method and its application in modeling dynamic response of reinforced concrete subjected to impact loading. *Computer Methods in Applied Mechanics and Engineering*, 200(17-20):1659–1670, apr 2011. doi: 10.1016/j.cma.2011.01.019.

[85] Y. Lian, X. Zhang, and Y. Liu. An adaptive finite element material point method and its application in extreme deformation problems. *Computer Methods in Applied Mechanics and Engineering*, 241-244:275–285, oct 2012. doi: 10.1016/j.cma.2012.06.015.

[86] Y. Liang, X. Zhang, and Y. Liu. Extended material point method for the three-dimensional crack problems. *International Journal for Numerical Methods in Engineering*, 122(12):3044–3069, Mar. 2021. doi: 10.1002/nme.6653.

[87] C. Liu and W. Sun. Shift boundary material point method: an image-to-simulation workflow for solids of complex geometries undergoing large deformation. *Computational Particle Mechanics*, 7(2):291–308, may 2019. doi: 10.1007/s40571-019-00239-y.

[88] W. Liu and J.-W. Hong. A coupling approach of discretized peridynamics with finite element method. *Computer Methods in Applied Mechanics and Engineering*, 245-46:163–175, 2012.

[89] A. Logg, K.-A. Mardal, and G. Wells. *Automated solution of differential equations by the finite element method: The FEniCS book*, volume 84. Springer Science & Business Media, 2012.

[90] E. Love and D. Sulsky. An energy-consistent material-point method for dynamic finite deformation plasticity. *International Journal for Numerical Methods in Engineering*, 65(10):1608–1638, 2006. doi: 10.1002/nme.1512.

[91] J. Lubliner. *Plasticity Theory.* Macmillan Publishing Company, New York, 1990.

[92] X. Ma, P. Giguere, B. Jayaraman, and Z. D.Z. Distribution coefficient algorithm for small mass nodes in material point method. *Journal of Computational Physics*, 229(20):7819–7833, oct 2010. doi: 10.1016/j.jcp.2010.06.041.

[93] Z. Ma, X. Zhang, and P. Huang. An object-oriented mpm framework for simulation of large deformation and contact of numerous grains. *Computer Modeling in Engineering & Sciences*, 55(1):61–88, 2010. ISSN 1526-1492.

[94] S. Maassen, R. Niekamp, J. Bergmann, F. Pöhl, J. Schröder, and P. Wiederkehr. Modeling of the Split-Hopkinson-Pressure-Bar experiment with the explicit material point method. *Computational Particle Mechanics.* doi: https://doi.org/10.1007/s40571-021-00399-w.

[95] T. Maeshima, Y. Kim, and T. Zohdi. Particle-scale numerical modeling of thermo-mechanical phenomena for additive manufacturing using the material point method. *Computational Particle Mechanics*, 8(3):613–623, Sept. 2020. doi: 10.1007/s40571-020-00358-x.

[96] A. Manes, L. Peroni, M. Scapin, and M. Giglio. Analysis of strain rate behavior of an Al 6061 T6 alloy. *Procedia Engineering*, 10:3477–3482, 2011. doi: 10.1016/j.proeng.2011.04.573.

[97] A. Manes, F. Serpellini, M. Pagani, M. Saponara, and M. Giglio. Perforation and penetration of aluminium target plates by armour piercing bullets. *International Journal of Impact Engineering*, 69:39–54, jul 2014. doi: 10.1016/j.ijimpeng.2014.02.010.

[98] K. Marchais, J. Girardot, C. Metton, and I. Iordanoff. A 3D DEM simulation to study the influence of material and process parameters on spreading of metallic powder in additive manufacturing. *Computational Particle Mechanics*, 8(4):943–953, Jan. 2021. doi: 10.1007/s40571-020-00380-z.

[99] C. Meier, S. Fuchs, A. Hart, and W. Wall. A novel smoothed particle hydrodynamics formulation for thermo-capillary phase change problems with focus on metal additive manufacturing melt pool modeling. *Computer Methods in Applied Mechanics and Engineering*, 381:113812, Aug. 2021. doi: 10.1016/j.cma.2021.113812.

[100] C. Miehe. *Kanonische Modelle multiplikativer Elasto-Plastizität. Thermodynamische Formulierung und Numerische Implementation.* PhD thesis, Universität Hannover, Institut für Baumechanik und Numerische Mechanik, Bericht-Nr. F93/1, 1993. Habilitationsschrift.

[101] L. Monforte, M. Arroyo, J. Carbonell, and A. Gens. Numerical simulation of undrained insertion problems in geotechnical engineering with the particle finite element method (PFEM). *Computers and Geotechnics*, 82:144–156, Feb. 2017. doi: 10.1016/j.compgeo.2016.08.013.

[102] J. Mroginski, H. Castro, J. Podestá, P. Beneyto, and A. Anonis. A fully coupled particle method for dynamic analysis of saturated soil. *Computational Particle Mechanics*, 8(4):845–857, Oct. 2020. doi: 10.1007/s40571-020-00373-y.

[103] J. Nairn. Numerical Simulation of Orthogonal Cutting using the Material Point Method. *Engineering Fracture Mechanics*, 149:262–275, 2015. doi: http://dx.doi.org/10.1016/j.engfracmech.2015.07.014.

[104] A. Neto, P. Pimenta, and P. Wriggers. Contact between spheres and general surfaces. *Computer Methods in Applied Mechanics and Engineering*, 328:686–716, Jan. 2018. doi: 10.1016/j.cma.2017.09.016.

[105] V. Nguyen, C. 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*, 126:50–64, apr 2017. doi: 10.1016/j.finel.2016.11.007.

[106] J. Ning and S. Liang. Model-driven determination of johnson-cook material constants using temperature and force measurements. *The International Journal of Advanced Manufacturing Technology*, 97(1-4):1053–1060, Apr. 2018. doi: 10.1007/s00170-018-2022-x. URL `https://doi.org/10.1007/s00170-018-2022-x`.

[107] E. Oñate and J. Rojek. Combination of discrete element and finite element methods for dynamic analysis of geomechanics problems. *Computer Methods in Applied Mechanics and Engineering*, 193(27-29):3087–3128, July 2004. doi: 10.1016/j.cma.2003.12.056.

[108] E. Oñate, S. Idelsohn, M. Celigueta, and R. Rossi. Advances in the particle finite element method for the analysis of fluid–multibody interaction and bed erosion in free surface flows. *Computer Methods in Applied Mechanics and Engineering*, 197 (19-20):1777–1800, Mar. 2008. doi: 10.1016/j.cma.2007.06.005.

[109] E. Oñate, A. Franci, and J. Carbonell. A particle finite element method for analysis of industrial forming processes. *Computational Mechanics*, 54(1):85–107, Apr. 2014. doi: 10.1007/s00466-014-1016-2.

[110] G. Ongaro, P. Seleson, U. Galvanetto, T. Ni, and M. Zaccariotto. Overall equilibrium in the coupling of peridynamics and classical continuum mechanics. *Computer Methods in Applied Mechanics and Engineering*, 381:113515, Aug. 2021. doi: 10.1016/j.cma.2020.113515.

[111] O. Pantalé, J.-L. Bacaria, O. Dalverny, R. Rakotomalala, and S. Caperaa. 2d and 3d numerical models of metal cutting with damage effects. *Computer Methods in Applied Mechanics and Engineering*, 193(39-41):4383–4399, Oct. 2004. doi: 10.1016/j.cma.2003.12.062. URL `https://doi.org/10.1016/j.cma.2003.12.062`.

[112] A. Popp and P. Wriggers, editors. *Contact Modeling for Solids and Particles*. Springer International Publishing, 2018. doi: 10.1007/978-3-319-90155-8.

[113] D. Rapaport. *The Art of Molecular Dynamics Simulation*. Cambridge University Press, Apr. 2004. doi: 10.1017/cbo9780511816581.

[114] S. Raymond, B. Jones, and J. Williams. Fracture shearing of polycrystalline material simulations using the material point method. *Computational Particle Mechanics*, 8 (2):259–272, Mar. 2020. doi: 10.1007/s40571-020-00327-4.

[115] J. M. Rodríguez, J. M. Carbonell, and P. Jonsén. Numerical methods for the modelling of chip formation. *Archives of Computational Methods in Engineering*, 27(2): 387–412, dec 2018. doi: 10.1007/s11831-018-09313-9.

[116] J. Rojek, E. Oñate, C. Labra, and H. Kargl. Discrete element simulation of rock cutting. *International Journal of Rock Mechanics and Mining Sciences*, 48(6):996–1010, Sept. 2011. doi: 10.1016/j.ijrmms.2011.06.003.

[117] M. Russell, A. Souto-Iglesias, and T. Zohdi. Numerical simulation of laser fusion additive manufacturing processes using the SPH method. *Computer Methods in Applied Mechanics and Engineering*, 341:163–187, Nov. 2018. doi: 10.1016/j.cma.2018.06.033.

[118] M. Sabel. *A particle finite element method for cutting of solids.* Technische Universität Kaiserslautern, Lehrstuhl für Technische Mechanik, Kaiserslautern, 2018. ISBN 978-3-942695-16-9.

[119] A. Sadeghirad, R. 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(12):1435–1456, mar 2011. doi: 10.1002/nme.3110.

[120] A. Sadeghirad, R. Brannon, and J. Guilkey. Second-order convected particle domain interpolation (CPDI2) with enrichment for weak discontinuities at material interfaces. *International Journal for Numerical Methods in Engineering*, 95(11):928–952, jul 2013. doi: 10.1002/nme.4526.

[121] F. Salazar, J. Irazábal, A. Larese, and E. Oñate. Numerical modelling of landslide-generated waves with the particle finite element method (PFEM) and a non-newtonian flow model. *International Journal for Numerical and Analytical Methods in Geomechanics*, 40(6):809–826, Sept. 2015. doi: 10.1002/nag.2428.

[122] M. Sandim, A. Paiva, and L. Henrique de Figueiredo. Simple and reliable boundary detection for meshfree particle methods using interval analysis. *Journal of Computational Physics*, 420:109702, nov 2020. doi: 10.1016/j.jcp.2020.109702.

[123] A. Savitzky and M. Golay. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry*, 36(8):1627–1639, July 1964. doi: 10.1021/ac60214a047.

[124] A. Schäuble. Variationally conistent inertia templates for speed-up and customization in explicit dynamics, 2019.

[125] J. Schöberl. C++ 11 implementation of finite elements in NGSolve. *Institute for Analysis and Scientific Computing, Vienna University of Technology*, 30, 2014.

[126] J. Schröder, T. Wick, S. Reese, P. Wriggers, R. Müller, S. Kollmannsberger, M. Kästner, A. Schwarz, M. Igelbüscher, N. Viebahn, H. Bayat, S. Wulfinghoff, K. Mang, E. Rank, T. Bog, D. D'Angella, M. Elhaddad, P. Hennig, A. Düster, W. Garhuom, S. Hubrich, M. Walloth, W. Wollner, C. Kuhn, and T. Heister. A Selection of Benchmark Problems in Solid Mechanics and Applied Mathematics. *Archives of Computational Methods in Engineering*, 28(2):713–751, 2021.

[127] H. Shin and J.-B. Kim. Evolution of specimen strain rate in Split Hopkinson Bar test. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 233(13):4667–4687, Jan. 2019. doi: 10.1177/0954406218813386.

[128] S. Silling. Reformulation of elasticity theory for discontinuities and long-range forces. *Journal of the Mechanics and Physics of Solids*, 48:175–209, 2000.

[129] S. Silling. Peridynamic modeling of the kalthoff-winkler experiment. In *Submission for the 2001 Sandia Prize in Computational Science*, 2002.

[130] S. Silling, M. Epton, O. Weckner, J. Xu, and E. Askari. Peridynamic states and constitutive modeling. *Journal of Elasticity*, 88(2):151–184, 2007.

[131] J. Simo. A framework for finite strain elastoplasticity based on maximum plastic dissipation and the multiplicative decomposition: Part I. continuum formulation. *Computer Methods in Applied Mechanics and Engineering*, 66:199–219, 1988.

[132] J. Simo. Algorithms for static and dynamic multiplicative plasticity that preserve the classical return mapping schemes of the infinitesimal theory. *Computer Methods in Applied Mechanics and Engineering*, 99:61–112, 1992.

[133] J. Simo and T. Hughes. *Computational Inelasticity.* Springer-Verlag., 1998.

[134] J. Simo and C. Miehe. Associative coupled thermoplasticity at finite strains: Formulation, numerical analysis and implementation. *Computer Methods in Applied Mechanics and Engineering*, 96:133–171, 1992.

[135] Simulia. *Abaqus Theory Manual.* Dassault Systémes, Providence, RI, USA, 2011.

[136] M. Steffen, R. Kirby, and M. Berzins. Analysis and reduction of quadrature errors in the material point method (MPM). *International Journal for Numerical Methods in Engineering*, 76(6):922–948, 2008.

[137] A. Stomakhin, C. Schroeder, L. Chai, J. Teran, and A. Selle. A material point method for snow simulation. 32(4):102, 2013.

[138] A. Stomakhin, C. Schroeder, C. Jiang, L. Chai, J. Teran, and A. Selle. Augmented MPM for phase-change and varied materials. *ACM Transactions on Graphics*, 33 (4):1–11, July 2014. doi: 10.1145/2601097.2601176.

[139] Y. Su, J. Tao, S. Jiang, Z. Chen, and J. Lu. Study on the fully coupled thermodynamic fluid–structure interaction with the material point method. *Computational Particle Mechanics*, 7(2):225–240, jul 2019. doi: 10.1007/s40571-019-00261-0.

[140] D. Sulsky and A. Kaul. Implicit dynamics in the material-point method. *Computer Methods in Applied Mechanics and Engineering*, 193(12-14):1137–1170, mar 2004. doi: 10.1016/j.cma.2003.12.011.

[141] D. Sulsky and H. Schreyer. Axisymmetric form of the material point method with applications to upsetting and Taylor impact problems. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):409–429, dec 1996. doi: 10.1016/s0045-7825(96)01091-2.

[142] D. Sulsky, Z. Chen, and H. Schreyer. A particle method for history-dependent materials. *Computer Methods in Applied Mechanics and Engineering*, 118(1-2): 179–196, sep 1994. doi: 10.1016/0045-7825(94)90112-0.

[143] D. Sulsky, S. Zhou, and H. Schreyer. Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications*, 87(1-2):236–252, may 1995. doi: 10.1016/0010-4655(94)00170-7.

[144] 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*, 112(C2), Feb. 2007. doi: 10.1029/2005jc003329.

[145] G. Taylor and H. Quinney. The latent energy remaining in a metal after cold working. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 143(849):307–326, Jan. 1934. doi: 10.1098/rspa.1934.0004.

[146] R. Tielen, E. Wobbes, M. Möller, and L. Beuth. A High Order Material Point Method. *Procedia Engineering*, 175:265–272, 2017. doi: 10.1016/j.proeng.2017.01.022.

[147] T. Trucano and D. Grady. Study of intermediate velocity penetration of steel spheres into deep aluminium targets. Technical Report SAND82-2338, Sandia National Laboratories, 1985.

[148] M. Tupek, J. Rimoli, and R. Radovitzky. An approach for incorporating classical continuum damage models in state-based peridynamics. *Computer Methods in Applied Mechanics and Engineering*, 263:20–26, 2013.

[149] R. Vacondio, C. Altomare, M. Leffe, X. Hu, Touzé, D., S. Lind, J.-C. Marongiu, S. Marrone, B. Rogers, and A. Souto-Iglesias. Grand challenges for smoothed particle hydrodynamics numerical schemes. *Computational Particle Mechanics*, 8(3): 575–588, Sept. 2020. doi: 10.1007/s40571-020-00354-1.

[150] G. van Rossum. The Python Language Reference. Release 3.8.9., 2021.

[151] L. Wang, W. Coombs, C. Augarde, M. Cortis, T. Charlton, M. Brown, J. Knappett, A. Brennan, C. Davidson, D. Richards, and A. Blake. On the use of domain-based material point methods for problems involving large distortion. *Computer Methods in Applied Mechanics and Engineering*, 355:1003–1025, oct 2019. doi: 10.1016/j.cma.2019.07.011.

[152] L. Wang, W. Coombs, C. Augarde, M. Cortis, M. Brown, A. Brennan, J. Knappett, C. Davidson, D. Richards, D. White, and A. Blake. An efficient and locking-free material point method for three-dimensional analysis with simplex elements. *International Journal for Numerical Methods in Engineering*, 122(15):3876–3899, May 2021. doi: 10.1002/nme.6685.

[153] H. Wessels, T. Bode, C. Weißenfels, P. Wriggers, and T. Zohdi. Investigation of heat source modeling for selective laser melting. *Computational Mechanics*, 63(5): 949–970, Sept. 2018. doi: 10.1007/s00466-018-1631-4.

[154] Z. Więckowski. The material point method in large strain engineering problems. *Computer Methods in Applied Mechanics and Engineering*, 193(39-41):4417–4438, Oct. 2004. doi: 10.1016/j.cma.2004.01.035.

[155] F. Wöste, J. Kimm, J. Bergmann, W. Theisen, and P. Wiederkehr. Investigation of the effect of residual stresses in the subsurface on process forces for consecutive orthogonal cuts. *Production Engineering*, May 2021. doi: 10.1007/s11740-021-01058-y.

[156] P. Wriggers. *Nichtlineare Finite-Element-Methoden*. Springer-Verlag, Berlin, 2001.

[157] P. Wriggers. *Computational Contact Mechanics*. Springer, 2003. ISBN 10 3-540-32608-1.

[158] P. Wriggers. *Nonlinear finite element methods*. Springer, 2008.

[159] Q. Xu, J. Zhao, and X. Ai. Cutting performance of tools made of different materials in the machining of 42CrMo4 high-strength steel: a comparative study. *The International Journal of Advanced Manufacturing Technology*, 93(5-8):2061–2069, June 2017. doi: 10.1007/s00170-017-0666-6.

[160] Y. Yamaguchi, S. Moriguchi, and K. Terada. Extended b-spline-based implicit material point method. *International Journal for Numerical Methods in Engineering*, 122(7):1746–1769, Dec. 2020. doi: 10.1002/nme.6598.

[161] A. Zabel, T. Rödder, and M. Tiffe. Material testing and chip formation simulation for different heat treated workpieces of 51crv4 steel. *Procedia CIRP*, 58:181–186, 2017. doi: 10.1016/j.procir.2017.03.218.

[162] X. Zhang, K. Y. Sze, and S. Ma. An explicit material point finite element method for hyper-velocity impact. *International Journal for Numerical Methods in Engineering*, 66(4):689–706, 2006. doi: 10.1002/nme.1579.

[163] X. Zhang, Z. Chen, and Y. Liu. *The Material Point Method*. Elsevier Science, 2016.

[164] H. Zhao and G. Gary. On the use of SHPB techniques to determine the dynamic behavior of materials in the range of small strains. *International Journal of Solids and Structures*, 33(23):3363–3375, sep 1996. doi: 10.1016/0020-7683(95)00186-7.

[165] O. C. Zienkiewicz and R. L. Taylor. *The finite element method for solid and structural mechanics*. Elsevier, 2005.

[166] T. Zohdi. Additive particle deposition and selective laser processing - a computational manufacturing framework. *Computational Mechanics*, 54:171–191, 2014.

Der Lebenslauf ist in der Online-Version aus Gründen des Datenschutzes nicht enthalten.

In Schriftenreihe des Instiuts für Mechanik bisher erschienene Berichte

Nr. 1 (2004) *Ein Modell zur Beschreibung finiter anisotroper elasto-plastischer Deformationen unter Berücksichtigung diskreter Rissausbreitung*, J. Löblein, Dissertation, 2004.

Nr. 2 (2006) *Polyconvex Anisotropic Energies and Modeling of Damage applied to Arterial Walls*, D. Balzani, Dissertation, 2006.

Nr. 3 (2006) *Kontinuumsmechanische Modellierung ferroelektrischer Materialien im Rahmen der Invariatentheorie*, H. Romanowski, Dissertation, 2006.

Nr. 4 (2007) *Mehrskalen-Modellierung polykristalliner Ferroelektrika basierend auf diskreten Orientierungsverteilungsfunktionen*, I. Kurzhöfer, Dissertation, 2007.

Nr. 5 (2007) *Proceedings of the First Seminar on the Mechanics of Multifunctional Materials*, J. Schröder, D.C. Lupascu, D. Balzani (Ed.), Tagungsband, 2007.

Nr. 6 (2008) *Zur Modellierung und Simulation diskreter Rissausbreitungsvorgänge*, O. Hilgert, Dissertation, 2008.

Nr. 7 (2009) *Least-Squares Mixed Finite Elements for Solid Mechanics*, A. Schwarz, Dissertation, 2009.

Nr. 8 (2010) *Design of Polyconvex Energy Functions for All Anisotropy Classes*, V. Ebbing, Dissertation, 2010.

Nr. 9 (2012) *Modeling of Electro-Mechanically Coupled Materials on Multiple Scales*, M.-A. Keip, Dissertation, 2012.

Nr. 10 (2012) *Geometrical Modeling and Numerical Simulation of Heterogeneous Materials*, D. Brands, Dissertation, 2012.

Nr. 11 (2012) *Modeling and simulation of arterial walls with focus on damage and residual stresses*, S. Brinkhues, Dissertation, 2012.

Nr. 12 (2014) *Proceedings of the Second Seminar on the Mechanics of Multifunctional Materials*, J. Schröder, D.C. Lupascu, M.-A. Keip, D. Brands (Ed.), Tagungsband, 2014.

Nr. 13 (2016) *Mixed least squares finite element methods based on inverse stress-strain relations in hyperelasticity*, B. Müller, Dissertation, 2016.

Nr. 14 (2016) *Electromechanical Modeling and Simulation of Thin Cardiac Tissue Constructs*, R. Frotscher, Dissertation, 2016.

Nr. 15 (2017) *Least-squares mixed finite elements for geometrically nonlinear solid mechanics*, K. Steeger, Dissertation, 2017.

Nr. 16 (2017) *Scale-Bridging of Elasto-Plastic Microstructures using Statistically Similar Representative Volume Elements*, L. Scheunemann, Dissertation, 2017.

Nr. 17 (2018) *Modeling of Self-healing Polymers and Polymeric Composite Systems*, S. Specht, Dissertation, 2017.

Nr. 18 (2018) *Proceedings of the Third Seminar on the Mechanics of Multifunctional Materials*, J. Schröder, D.C. Lupascu, H. Wende, D. Brands (Ed.), Tagungsband, 2018.

Nr. 19 (2018) *Least-squares finite element methods with applications in fluid and solid mechanics*, C. Nisters, Dissertation, 2018.

Nr. 20 (2018) *A two-scale homogenization scheme for the prediction of magneto-electric product properties*, M. Labusch, Dissertation, 2018.

Nr. 21 (2019) *Modeling the passive mechanical response of soft tissues: constitutive modeling approaches, efficient parameter selection and subsequent adjustments due to residual stresses*, M. von Hoegen, Dissertation, 2019

Nr. 22 (2019) *Constitutive modeling of female pelvic floor dysfunctions and reconstructive surgeries using prosthetic mesh implants*, A. Bhattarai, Dissertation, 2019

Nr. 23 (2019) *A contribution to stress-displacement based mixed galerkin finite elements for hyperelasticity*, N. Viebahn, Dissertation, 2019

Nr. 24 (2020) *Gefrier- und Auftauprozesse in gesättigten porösen Materialien - ein Modellierungskonzept im Rahmen der Theorie poröser Medien*, W.M. Bloßfeld, Dissertation, 2020

Nr. 25 (2021) *Electromechanical modelling and simulation of hiPSC-derived cardiac cell cultures*, A. Jung, Dissertation, 2021

Nr. 26 (2021) *Mixed and Hybrid Least-Squares FEM in Nonlinear Solid Mechanics*, M. Igelbüscher, Dissertation, 2021

DOI: 10.17185/duepublico/78073
URN: urn:nbn:de:hbz:465-20230628-115544-8