Regularized MPM for porous media

Vom Promotionsausschuss der
Technischen Universität Hamburg

zur Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte Dissertation

von
Alexander Chmelnizkij

aus
Kiew, Ukraine

2023

1. Gutachter: Prof. Dr.-Ing. Jürgen Grabe, TUHH
2. Gutachter: Prof. Dr.-Ing. habil. Alexander Düster, TUHH
3. Gutachter: Prof. Dr. Paolo Simonini, Universität Padua

Tag der mündlichen Prüfung: 20.12.2022

Regularized MPM for porous media

von

Alexander Chmelnizkij

Herausgegeben von

J. Grabe

Technische Universität Hamburg
Institut für Geotechnik und Baubetrieb

Veröffentlichungen des Instituts für
Geotechnik und Baubetrieb

57

Herausgeber:

Univ.-Prof. Dr.-Ing. Jürgen Grabe
Technische Universität Hamburg
Institut für Geotechnik und Baubetrieb
Harburger Schloßstraße 36
D – 21079 Hamburg
e-mail: *grabe@tuhh.de*

ISBN-13: 978-3-936310-59-7(Erstausgabe)

DOI: 10.15480/882.4941

Druckerei:

Druckzentrum Neumünster GmbH
Rungestraße 4
24537 Neumünster

In derselben Reihe erschienen:

1. J. Grabe (Hrsg.), 2000: Verbrennungsrückstände. Tagungsband, ISBN 3-936310-00-9
2. J. Grabe (Hrsg.), 2001: Schaden- und Risikomanagement im Tiefbau. Tagungsband, ISBN 3-936310-01-7
3. J. Grabe, 2003. Bodenmechanik und Grundbau. ISBN 3-936310-03-3
4. J. Grabe (Hrsg.), 2003: Euronormen in der Geotechnik – Was ändert sich? Tagungsband, ISBN 3-936310-04-1
5. J. Grabe (Hrsg.), 2003: Bodenverdichtung, Experimente - Modellierung - Geräteentwicklung - Baustellenberichte - F+E-Bedarf. Tagungsband, ISBN 3-936310-05-X
6. M. Kelm, 2004: Nummerische Simulation der Verdichtung rolliger Böden mittels Vibrationswalzen. Dissertation, ISBN 3-936310-06-8
7. J. Grabe (Hrsg.), 2004: Kaimauern - Messungen und Numerik. Tagungsband, ISBN 3-936310-07-6
8. J. Stein, 2005. Experimentelle und numerische Untersuchungen zum Düsenstrahlverfahren. Dissertation, ISBN 3-936310-09-2
9. J. Grabe (Hrsg.), 2005: Grenzschicht Wasser und Boden - Phänomene und Ansätze. Tagungsband, ISBN 3-936310-10-6
10. J. Grabe (Hrsg.), 2005: FEM in der Geotechnik - Qualität, Prüfung, Fallbeispiele - Tagungsband, ISBN 3-936310-11-4
11. B. Mardfeldt, 2006: Zum Tragverhalten von Kaikonstruktionen im Gebrauchszustand. Dissertation, ISBN 3-936310-12-2
12. J. Grabe (Hrsg.), 2006: Optimierung in der Geotechnik - Strategien und Fallbeispiele. Tagungsband, ISBN-13: 978-3-936310-13-9
13. T. Bierer, 2007: Bodenschwingungen aus Straßenverkehr auf unebener Fahrbahn im Zeitbereich - experimentelle und theoretische Untersuchungen. Dissertation, ISBN-13: 978-3-936310-14-6
14. J. Grabe (Hrsg.), 2007: Bemessen mit Finite-Elemente-Methoden. Tagungsband, ISBN-13: 978-3-936310-15-3
15. K.-P. Mahutka, 2008: Zur Verdichtung von rolligen Böden infolge dynamischer Pfahleinbringung und durch Oberflächenrüttler. Dissertation, ISBN-13: 978-3-936310-16-0

16. J. Grabe (Hrsg.), 2008: Seehäfen für Containerschiffe zukünftiger Generationen. Tagungsband, ISBN-13: 978-3-936310-17-7
17. F. König, 2008: Zur zeitlichen Traglastentwicklung von Pfählen und der nachträglichen Erweiterung bestehender Pfahlgründungen. Dissertation, ISBN-13: 978-3-936310-18-4
18. S. Henke, 2008: Herstellungseinflüsse aus Pfahlrammung im Kaimauerbau. Dissertation, ISBN-13: 978-3-936310-19-1
19. J. Grabe (Hrsg.), 2009: Spundwände – Profile, Tragverhalten, Bemessung, Einbringung und Wiedergewinnung. Tagungsband, ISBN-13: 978-3-936310-20-7
20. J. Dührkop, 2009: Zum Einfluss von Aufweitungen und zyklischen Lasten auf das Verformungsverhalten lateral beanspruchter Pfähle in Sand. Dissertation, ISBN-13: 978-3-936310-21-4
21. O. Möller, 2009: Zum Langzeit-Kompressionsverhalten weicher organischer Sedimente. Dissertation, ISBN-13: 978-3-936310-22-1
22. J. Grabe (Hrsg.), 2011: Ports of container ships of future generations. Tagungsband, ISBN-13: 978-3-936310-23-8
23. S. Kinzler, 2011: Zur Parameteridentifikation, Entwurfs- und Strukturoptimierung in der Geotechnik mittels numerischer Verfahren. Dissertation, ISBN-13: 978-3-936310-24-5
24. G. Qiu, 2012: Coupled Eulerian Lagrangian Simulations of Selected Soil-Structure Problems. Dissertation, ISBN-13: 978-3-936310-25-2
25. X. Ma, 2013: Nutzung der oberflächennahen Geothermie mittels Energiepfählen und Erdwärmesonden. Dissertation, ISBN-13: 978-3-936310-26-9
26. J. Grabe (Hrsg.), 2013: Proceedings of the Conference on Maritime Energy COME 2013. Tagungsband, ISBN-13: 978-3-936310-28-3
27. J. Grabe (Hrsg.), 2013: Bemessen mit numerischen Methoden. Tagungsband, ISBN-13: 978-3-936310-29-0
28. T. Pucker, 2013: Stoffmodell zur Modellierung von stetigen Materialübergängen im Rahmen der Optimierung geotechnischer Strukturen. Dissertation, ISBN-13: 978-3-936310-30-6
29. S. Henke, 2013: Untersuchungen zur Pfropfenbildung infolge der Installation offener Profile in granularen Böden. Habilitation, ISBN-13: 978-3-936310-31-3
30. J. Grabe (Hrsg.), 2014: Ports for Container Ships of Future Generations. Tagungsband, ISBN-13: 978-3-936310-32-0

31. J. Grabe (Hrsg.), 2014: Offshore Basishäfen, Tagungsband, ISBN-13: 978-3-936310-33-7
32. C. Rudolph, 2015. Untersuchungen zur Drift von Pfählen unter zyklischer, lateraler Last aus veränderlicher Richtung, Dissertation, ISBN-13: 978-3-936310-34-4
33. J. Grabe (Hrsg.), 2015: Morphodynamics 2015, Tagungsband, ISBN-13: 978-3-936310-35-1
34. T. Hamann, 2015: Zur Modellierung wassergesättigter Böden unter dynamischer Belastung und großen Bodenverformungen am Beispiel der Pfahleinbringung, Dissertation, ISBN-13: 978-3-936310-36-8
35. B. Schümann, 2015: Beitrag zum dynamischen Dreiphasenmodell für Boden auf Basis der Finite-Elemente-Methode, Dissertation, ISBN-13: 978-3-936310-37-5
36. M. Milatz, 2015: Untersuchungen zum Einfluss der Kapillarität auf das hydraulisch-mechanische Verhalten von granularer Tragschichten für Verkehrswege, Dissertation, ISBN-13: 978-3-936310-38-2
37. H. Kaya, 2016: Bodenverschleppung und Spaltbildung infolge der Einbringung von Profilen in Dichtungsschichten aus Ton, Dissertation, ISBN-13: 978-3-936310-39-9
38. J. Grabe (Hrsg.), 2017: Proceedings of the Conference on Maritime Energy COME 2017. Tagungsband, ISBN-13: 978-3-936310-40-5
39. B. Kocak, 2017: Zur numerischen Modellierung von hydraulisch-mechanisch gekoppelten Prozessen in gesättigten granularen Böden mittels Smoothed Particle Hydrodynamics, Dissertation, ISBN-13: 978-3-936310-41-2
40. K. Siegl, 2017: Zur Pfahldynamik von gerammten Großrohrpfählen und der daraus resultierenden Wellenausbreitung in Wasser und im Meeresboden, Dissertation, ISBN-13: 978-3-936310-42-9
41. J. Grabe (Hrsg.), 2017: Numerical Methods in Geotechnics, Tagungsband, ISBN-13: 978-3-936310-43-6
42. J. Grabe (Hrsg.), 2018: Digitale Infrastruktur und Geotechnik (DIG 2018), Tagungsband, ISBN-13: 978-3-936310-44-3
43. D. Osthoff, 2018: Zur Ursache von Schlosssprengungen und zu einbringbedingten Lageabweichungen von Spundwänden, Dissertation, ISBN-13: 978-3-936310-45-0
44. E. Heins, 2018: Numerical based identification of the pile-soil interaction in terms of the axial pile bearing capacity, Dissertation, ISBN-13: 978-3-936310-46-7

45. K.-F. Seitz, 2021: Zur Topologieoptimierung von geotechnischen Strukturen und zur Tragfähigkeitssteigerung des Baugrunds durch Scherfugenverfestigung, Dissertation, ISBN-13: 978-3-936310-47-4

46. D. Plenker, 2021: Physical and numerical investigations of the dynamic interaction of saturated granulates and fluid, Dissertation, ISBN-13: 978-3-936310-48-1

47. J. Grabe, J.-O. Backhaus & P. Vogel, 2021: Bauprojektmanagement, ISBN-13: 978-3-936310-49-8

48. M. Kanitz, 2021: Experimental and numerical investigations of particle-fluid systems in geotechnical engineering, Dissertation, ISBN-13: 978-3-936310-50-4

49. J.-O. Backhaus, 2021: A methodology for the numeric time-cost forecast and pareto optimization of large injection projects in tunneling, Dissertation, ISBN-13: 978-3-936310-51-1

50. S. N. Sinduri, 2021: Optimisation of deep compaction as liquefaction mitigation measure, Dissertation, ISBN-13: 978-3-936310-52-8

51. J. Bubel, 2022: Zum Versagen von Unterwasserböschungen im Seegang, Dissertation, ISBN-13: 978-3-936310-53-5

52. J. Grabe (Hrsg.), 2022: Adaptive Structures At Shore, Tagungsband, ISBN-13: 978-3-936310-54-2

53. J. Grabe (Hrsg.), 2022: Numerische Methoden in der Geotechnik, Tagungsband, ISBN-13: 978-3-936310-55-9

54. J. Grabe (Hrsg.), 2022: Processes in natural and technical Particle-Fluid-Systems (PintPFS), Tagungsband, ISBN-13: 978-3-936310-56-6

55. T. Törzs, 2022: Zum hydraulisch-mechanisch gekoppelten Verhalten teilgesättigter granularer Geomaterialien infolge hydraulischer und mechanischer Beanspruchungen, Dissertation, ISBN-13: 978-3-936310-57-3

56. M. Milatz, 2022: Investigation of capillary effects on the grain scale by means of in situ experiments, imaging and numerical simulations, Habilitation, ISBN-13: 978-3-936310-58-0

To my family and friends

Editor's Preface

Die sogenannte „Standard-Material-Point-Method (MPM)“ wurde in den 90er Jahren vorgestellt. Coetzee hat die Standard MPM im Jahr 2004 für die Modellierung von großen Verformungen im Boden unter Nutzung der Biot-Gleichungen erweitert und damit eine neue Forschungsrichtung in der numerischen Geotechnik initiiert. Mit MPM gelingt es beispiels-weise das Versagen einer Bodensäule unter Eigengewicht, die dynamische Einbringung von Profilen in den Untergrund, die Verdichtung von Böden mittels schwerer Fallmassen oder die Explosion von Blindgängern im Untergrund numerisch zu modellieren. Solche dynamischen Herstellungsvorgänge mit großen Verformungen bilden eine der größten Herausforderungen in der numerischen Geotechnik.

Seit der Arbeit von Coetzee (2004) befassen sich Wissenschaftler aus der ganzen Welt mit der Weiterentwicklung der MPM auf geotechnische Problemstellungen mit großen Verformungen. Zu nennen ist hierbei die von Professor Peter Vermeer angestoßene internationale Forschergruppe (MPM Community), deren Programmcode unter Anura3D bekannt geworden ist, einer Gruppe in China und einer in Polen. Alexander Chmelnizkij ist seit Jahren Mitglied der Kernentwicklungsgruppe des Programms Anura3D, welches inzwischen in Teilen als open-source-code der Wissenschaft und Praxis zur Verfügung steht.

Alexander Chmelnizkij befasst sich im Rahmen seiner Dissertation mit der Double-Point MPM (2P-MPM), um beispielsweise eine Separierung von Wasser und gesättigtem Boden abbilden zu können. Zur Erzielung einer ausreichenden Genauigkeit und Konvergenz ist es zwingend notwendig, dass genügend numerische Materialpunkte in den Gebietselementen vorhanden sind, da es sonst zu ungewünschten Reflexionen kommen kann. Um dies zu gewährleisten, entwickelt Alexander Chmelnizkij im Rahmen seiner Dissertation eine regularisierte 2P-MPM.

Die Arbeit ist zu wesentlichen Teilen im Rahmen des von der DFG geförderten Forschungsprojektes GR 1024/45 „Untersuchung geotechnischer Randwertprobleme mit Boden-Wasser-Interaktion mittels Double-Point Material Point Method (2P-MPM)“ entstanden.

Zielsetzung der Arbeit von Alexander Chmelnizkij ist gemäß Kapitel 3, Seite 5:

„*From the investigations made, five main objectives emerge*:

- The first objective of the thesis is the derivation and overview of governing equations...
- The second and main objective is the review of major developments in MPM with emphasis on the Improved MPM (iMPM) recently developed by Sulsky and Gong (2016).... A new approach based on matrix regularization is proposed ..

- The third objective is the extension of the proposed modification of iMPM to the coupled formulation for saturated soil.
- The forth objective is the parallelization of the 2P-MPM. ..
- The fifth objective, resulting from the previous points, is the development of the Open-Source code in MATLAB including the developed methods.“

Alexander Chmelnizkij führt valide Grundsatzuntersuchungen und Weiterentwicklungen zur MPM für poröse Materialien (Boden) durch. Dazu hat er einen eigenen Code in Matlab geschrieben und damit die Möglichkeit bekommen, Schritt für Schritt einzelne Entwicklungen zu analysieren, voran zu treiben und auszuprobieren. Die Vorgehensweise ist dabei äußerst systematisch. Hier kommt ihm sein Studium der Mathematik zu Gute.

In Kapitel 1 führt er knapp in die Thematik ein. Der Stand der Forschung ist in Kapitel 2 auf lediglich 2 Seiten dargestellt. Dies begründet sich dadurch, dass er die Ableitung und den Überblick der wesentlichen Gleichungen als erstes Ziel ausgegeben hat. Entsprechende Ausführungen finden sich daher in den nachfolgenden Kapiteln 4 und 5. In Kapitel 4 führt er die wesentlichen Gleichungen und Grundlagen für poröse Materialien auf. In Kapitel 5 behandelt er die Grundlagen der Material Point Method. Wichtig ist dabei seine Analyse der Probleme der Standard MPM, wie beispielsweise „Grid Crossing“, im Abschnitt 5.2. Im Abschnitt 5.3 geht er auf Verbesserungen wie die „Generalized Interpolation Material Point Method (GIMP)“ ein. Anschaulich werden die Unterschiede durch die Bilder 5.4 und 5.5 für das Beispiel eines auf eine starre Platte fallenden Balls. In Kapitel 6 geht es um die Regularisierung der MPM, um eine höhere Genauigkeit und Robustheit zu erzielen. Abschnitt 6.3.1 behandelt die adaptive Regularisierung. Die Double-Point MPM wird in Kapitel 7 hergeleitet, siehe Bild 7.1. Die mit 2P-MPM simulierte quasistatische Konsolidierung einer Bodensäule wird in Bild 7.4 mit der analytischen Lösung verglichen. Die 1dimensionale Wellenausbreitung in gesättigtem Boden wird in Abschnitt 7.6 untersucht. In Kapitel 8 beschreibt er die von ihm implementierte Parallelisierung der 2P-MPM. Bilder 8.2 und 8.3 zeigen den Vergleich für zwei benchmarks und Bild 8.4 die damit erzielte Beschleunigung der Rechenzeit. In Kapitel 9 wendet er seinen Code auf das Problem eines im Boden explodierenden Blindgängers (Kampfmittel) an, siehe Bild 9.2. Die Arbeit schließt mit einem knapp formulierten Fazit und einem Ausblick in Kapitel 10.

Die von Alexander Chmelnizkij vorgelegte Arbeit besticht durch die äußerst systematische Vorgehensweise bei der Herleitung und Ableitung von Verbesserungen sowie Erweiterungen der 2P-MPM für poröse Materialien. Diese umfangreichen mathematisch-numerischen Untersuchungen und Weiterentwicklungen bilden den Schwer¬punkt seiner Arbeit. Nicht enthalten in der von ihm vorgelegten Dissertation sind seine Code-Entwicklungen in Anura3D. Ihm ist als Mathematiker die Weiterentwicklung der MPM zur 2P-MPM mit adaptiver Regularisierung offensichtlich wissenschaftlich wichtiger als die ausführliche Darstellung der von ihm durchgeführten Messungen und Modellierungen zur Einbringung von Profilen in den Baugrund, zur Baugrundverbesserung mittels dynamischer Fallgewichte sowie zur Untersuchung der Wellenausbreitung von Detonationen im gesättigten Boden.

Alexander Chmelnizkij hat bisher in zwei Buchbeiträgen, in einer qualitätsgesicherten Fachzeitschrift und in 11 Beiträgen auf internationalen und nationalen Tagungen publiziert. Es ist nun vordinglich den Kern seiner Dissertation in qualitätsgesicherten Fachzeitschriften der Fachwelt zur Verfügung zu stellen.

Zusammenfassend ist festzuhalten, dass Alexander Chmelnizkij mit seinen mathematisch-numerischen Entwicklungen einen wesentlichen Beitrag zum Stand der Wissenschaft in der Material Point Method geleistet hat.

Ich freue mich sehr, dass Alexander Chmelnizkij auch weiterhin - nun als Postdoktorand - bei uns am Institut tätig sein wird.

Hamburg, 28. Dezember 2022

Jürgen Grabe

Author's Preface

Die vorliegende Dissertation ist während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Geotechnik und Baubetrieb der Technischen Universität Hamburg entstanden.

In dieser Zeit hatte ich das Vergnügen, Teil der Anura3D MPM Research Community zu sein und im regelmäßigen wissenschaftlichen Austausch mit meinen geschätzten internationalen Kollegen zu stehen. Viele Ideen und Ansätze dieser Arbeit sind durch diesen Austausch entstanden. Ich bin sehr glücklich darüber, dass aus der kollegialen Zusammenarbeit Freundschaften entstanden sind, die unsere gemeinsame Forschungsarbeit noch erfreulicher gemacht haben. Ich freue mich auch in der Zukunft weiter innerhalb dieser Gruppe arbeiten zu können und an der Entwicklung der MPM beteiligt zu sein.

Für die Unterstützung bei der Ausfertigung der Dissertation möchte ich besonders Georg, Martin, Paco, Dominik und Sparsha danken.

Mein besonderer Dank gilt Herrn Prof. Dr.-Ing. Jürgen Grabe für die vielseitigen und spannenden Möglichkeiten an unserem Institut, die zur Fertigstellung dieser Dissertation geführt haben. Weiterhin möchte ich Herrn Prof. Dr.-Ing. habil. Alexander Düster und Herrn Prof. Dr. Paolo Simonini für die Begutachtung und das Interesse an meinem Forschungsthema danken. Bei Herrn Prof. Dr.-Ing. Bastian Oesterle bedanke ich mich für die Übernahme des Prüfungsvorsitzes.

Ich möchte mich herzlich bei meinen Kollegen am Institut für Geotechnik und Baubetrieb, die mir immer unterstützend zur Seite gestanden haben bedanken und hoffe, dass der fachliche und gesellige Austausch weiterhin bestehen bleibt.

Meiner Familie möchte ich einen ganz besonderen Dank aussprechen, der weit über das Fachliche und Wissenschaftliche hinausgeht. Eure Unterstützung und Wertschätzung ist die größte Motivation für mich.

Hamburg, Februar 2023

Alexander Chmelnizkij

Schlagwörter:

Material-Point-Methode (MPM), Wellenausbreitung in Gesättigten Porösen Medien, Moving-Least-Squares (MLS), Regularisierung, Große Verfomungen, Parallelisierung

Keywords:

Material Point Method (MPM), Wave propagation in saturated porous media, Moving-Least-Squares (MLS), Regularization, Large Deformation, Parallization

Contents

Chapter 1

Introduction

In geotechnical engineering often, problems are considered in which both wave propagation and large deformations play an essential role. In addition, the soil under consideration is often layered and saturated with water. In many respects, the numerical simulation of such problems is currently being investigated using various methods. Two aspects are of particular interest in this investigation.

On one hand, the question of correct governing equations is not resolved, and one can find numerous publications that use different sets of equations. On the other hand, a suitable numerical method for solving the governing equations has to be answered. Thereby, the numerical method must be able to represent large deformations in addition to the corresponding accuracy and stability.

Within the framework of continuum mechanics, there are different two-phase models for saturated soil that describe the internal interactions between soil and pore water. In order to consider dynamic problems, theories are required to take into account inertia effects. Biot (1956a) and Biot (1956b) developed the first set of dynamic equations, and a similar formulation was presented in the same year by Jong (1956). These formulations are based on some assumptions which, according to Zienkiewicz, Chan, *et al.* (1999) and van Dalen (2013) can be described as:

- The continuum for the grain structure and the pore water is assumed to be superposed.
- The representative volume element in the derivation of the governing equations is considered to be small in comparison to the occurring wavelengths and large in comparison to the soil grains.
- The grain structure is assumed to deform according to a linearly elastic stress-strain-model.
- Small deformations of the soil are assumed, so the porosity and permeability remain constant.
- The pore fluid is assumed to have no shear strength.

- The solid and liquid phases are homogeneous and isotropic.
- The continuum should obey mass and momentum conservation.
- Thermal and chemical processes are neglected.

Biot's formulation considers, in contrast to Terzaghi's consolidation theory, the compressibility of the soil grains and pore water. An intensive numerical study of Biot's equations was carried out in Zienkiewicz, Chan, *et al.* (1999). Theories considering large deformations and therefore a changing porosity field were developed for example in the framework of the mixture theory. A detailed overview of the development of different formulations can be found in Boer (2000).

One aim of this thesis is to develop a numerical method for solving dynamic boundary value problems undergoing large deformations for saturated soils. Therefore, in particular the 2P-MPM (also known as Double-Point- or Dual-Layer-Formulation Material Point Method) Więckowski (2013) method is considered. As a starting point existing methods are implemented and investigated. These implementations are then improved based on two main aspects. The stability and accuracy of MPM and the optimization of the calculation time, which enables to calculate a larger number of material points (MP) and elements.

The origins of MPM can be found in fluid mechanics with the Particle in Cell (PIC) method (Harlow, 1962). The so-called Standard MPM, was first introduced by Sulsky, Chen, *et al.* (1994), which further developed the idea of PIC for solids. The application of PIC and MPM for the simulation of geotechnical boundary value problems was first shown in C. J. Coetzee (2004) and C. Coetzee *et al.* (2005). In Al-Kafaji (2013) and Więckowski (2013) the first dynamic formulation for water-saturated soil was presented. Numerous extensions and improvements of the method have been published in the last decades, which are summarized in de Vaucorbeil *et al.* (2020). The Double-Point-MPM (2P-MPM) (Więckowski, 2013) is of particular interest for the simulation of geotechnical problems and the interaction between structures, water and soil. It is able to represent deformations of any size for all considered materials. The mentioned formulations for saturated material were derived based on the Standard MPM approach and inherited some numerical issues, which will be explained further along this thesis. It is desirable to extend the latest improvements in MPM to the 2P-MPM in order to minimize the numerical errors of the method.

Chapter 2

State of research

In geotechnical engineering, MPM has become of immense interest in recent years. In particular, the Anura3D MPM Research Community (www.anura3d.com) focuses on applying MPM in geotechnics. The release of an Open-Source code, the organization of two international conferences and the publication of an MPM book (Fern *et al.*, 2019) for geotechnical applications raised the attention of the method in the geotechnical world. Besides the applications for geotechnical problems, MPM continues to develop through contributions from numerous fields. Usage of MPM encompassed the non-scientific communities in the form of animation for Disney movies like Frozen, Big Hero 6 and Zootopia (Jiang *et al.*, 2016).

Due to different applications, many variants of MPM have been developed over the years. In particular, most of the MPM variants differ in applying different nodal and material point (MP) shape functions. The reason for that being the overwhelming spatial error occurring in large deformation problems. Bardenhagen and Kober (2004) introduced the generalized interpolation material point (GIMP) method, which attempted to mitigate the well-known grid-crossing error in the Standard MPM by smoother shape functions. The characteristic function with compact support was introduced as MP shape function. M.Steffen *et al.* (2008) suggested using higher order B-Splines instead of linear nodal shape functions and thus introduced the B-Spline MPM (BS-MPM). Despite the improvements for the grid-crossing issue made by GIMP and BS-MPM, the methods still suffered from undesired gaps in the form of empty elements, leading to unphysical separation of materials. To overcome this issue, Sadeghirad, R. M. Brannon, *et al.* (2011) and Sadeghirad, R. Brannon, *et al.* (2013) introduced the Convected Particle Domain Interpolation (CPDI) and later the CPDI2, which could handle tensile deformations at the cost of introducing deformable domains for MP. The use of these deformable domains led to similar problems of mesh distortion as found in Lagrangian FEM. The fact that the number of MP and nodes is not equal in MPM led to the discovery of the so-called Null-Space problem by Gritton *et al.* (2015). This problem is caused by the rank deficiency of the global and local mapping matrices. Gritton *et al.* (2015) introduced a Null-Space filter to overcome this issue.

Recent developments by Sulsky and Gong (2016) connected MPM closer to the framework of meshfree methods. The computational grid in MPM can be regarded as a circumvention for discretising differential operators on scattered data points and a simplified neighbour searching tool. In particular, the similarity of weighted least squares (WLS) (Wallstedt and Guilkey, 2011) or moving least squares (MLS) (Edwards and Bridson, 2012) approximations and the mapping between MP and nodes, allowed the application of methods developed for meshfree methods in MPM. Many meshfree methods are suitable, for the simulation of large deformations of solid material as no mesh distortion appears. Research regarding the reproduction of polynomials, Partition of Unity property, and treatment of boundary conditions can be directly used for MPM. Findings in the Smoothed Particle Hydrodynamics (SPH) (Gingold and Monaghan, 1977), the Element Free Galerkin (EFG) (Belytschko, Lu, *et al.*, 1994), the Reproducing Kernel Particle Method (RKPM) (Liu *et al.*, 1995), the Particle Finite Element Method (PFEM) (Idelsohn *et al.*, 2006; Sabel *et al.*, 2014), the optimal transport meshfree method (OTM) (B. Li *et al.*, 2010) and many others can be adopted to develop better MPM variants.

Chapter 3

Objectives

The present work deals with the subject of numerical accuracy and robustness in MPM. In particular the application of MPM to formulations used in geotechnics for dry and saturated soil is considered. From the investigations made, five main objectives emerge.

The first objective of the thesis is the derivation and overview of governing equations often used in geotechnics for saturated soils. In addition a brief link between different formulations is made with emphasis on their different assumptions. Finally an overview of analytical solutions is given.

The second and main objective is the review of major developments in MPM with emphasis on the Improved MPM (iMPM) recently developed by Sulsky and Gong (2016). In particular, the MLS approximation is investigated, focusing on approximation errors. A new approach based on matrix regularization is proposed to improve the robustness and accuracy of iMPM. Further, a new weighting function, the Wendland function, well known in scattered data approximation for MPM, is introduced. The convergence of the proposed methods for small and large deformations is checked by the method of manufactured solutions (MMS).

The third objective is the extension of the proposed modifications of iMPM to the coupled formulation for saturated soil. In particular, modifications are applied to Double-Point MPM (2P-MPM). The correctness of the proposed method is checked with analytical solutions.

The fourth objective is the parallelization of the 2P-MPM. Therefore, the vectorization of the algorithm is described. The vectorized code is implemented in MATLAB and is benchmarked on a CPU and GPU to estimate the gained speed-up.

The fifth objective, resulting from the previous points, is the development of an Open-Source code in MATLAB including the developed methods. The code should be compact and easy to extend with further material models or numerical schemes.

Chapter 4

Governing Equations

The governing equations for dry and saturated porous media are derived in this chapter. A comparison with Biot's Poroelasticity is carried out, and analytical approaches to obtain the solutions are discussed.

4.1 Fundamentals

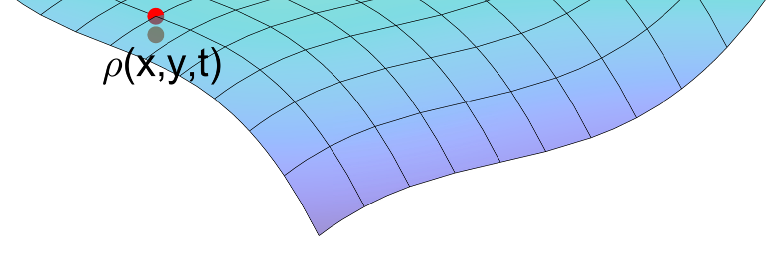


Figure 4.1: Material point moving through the scalar field ρ. The scalar field is first evaluated at the point (x,y) at time t, i.e. $\rho(x,y,t)$. At the same point, however, at time $t + \Delta t$, the function value changes according to $\rho(x,y,t+\Delta t)$. The new value experienced by a material point in the scalar field as it moves from point (x,y) to point $(x+\Delta x,y+\Delta y)$ during time Δt is given by $\rho(x+\Delta x,y+\Delta y,t+\Delta t)$.

Substantial Derivative Consider a material point moving through a scalar field ρ, i.e. density or temperature, from $(x{,}y)$ to $(x+\Delta x{,}y+\Delta y)$ as shown in Fig. 4.1. The grid in the figure is only intended to represent the curvature of the scalar field and should not be confused with a computational grid. Additionally, the scalar field is changing in time. Assuming that the scalar field is continuously differentiable, we can expand it around the point $(x{,}y)$ in a Taylor series (Anderson and Wendt, 1995) with respect to all variables.

$$\begin{aligned}\rho(x+\Delta x{,}y+\Delta y{,}t+\Delta t) = \rho(x{,}y{,}t) &+ \frac{\partial\rho}{\partial x}(x{,}y{,}t)\Delta x \\ &+\frac{\partial\rho}{\partial y}(x{,}y{,}t)\Delta y + \frac{\partial\rho}{\partial t}(x{,}y{,}t)\Delta t \\ &+R(\Delta x,\Delta y{,}\Delta t)\end{aligned} \tag{4.1}$$

Where the function $R(\Delta x,\Delta y{,}\Delta t)$ contains higher order terms. We rearrange Eqn. 4.1 to obtain the difference quotient for the total change of ρ with time.

$$\begin{aligned}\frac{\rho(x+\Delta x{,}y+\Delta y{,}t+\Delta t)-\rho(x{,}y{,}t)}{\Delta t} &= \frac{\partial\rho}{\partial x}(x{,}y{,}t)\frac{\Delta x}{\Delta t} \\ &+\frac{\partial\rho}{\partial y}(x{,}y{,}t)\frac{\Delta y}{\Delta t}+\frac{\partial\rho}{\partial t}(x{,}y{,}t) \\ &+R(\Delta x,\Delta y{,}\Delta t)\end{aligned} \tag{4.2}$$

The limit case $\Delta t{,}\Delta x{,}\Delta y \to 0$ in Eqn. 4.2 yields the substantial or material derivative of ρ in the direction $\mathbf{v} := \lim_{\Delta t,\Delta x,\Delta y\to 0}(\Delta x/\Delta t\,,\,\Delta y/\Delta t)$. This derivative is often expressed as follows,

$$\frac{\mathrm{D}\rho}{\mathrm{D}t}\,(x{,}y{,}t) = \frac{\partial\rho}{\partial t}\,(x{,}y{,}t) + \mathbf{grad}\,(\rho)\cdot\mathbf{v}\,. \tag{4.3}$$

This equation is generally valid for any direction $\mathbf{v}$. In the case where x and y are functions of t, $\mathbf{v}$ is the underlying velocity field where a material point moves along the trajectory $(x(t){,}y(t))$ corresponding to $\mathbf{v} = (\partial x/\partial t\,,\,\partial y/\partial t)$. The material derivative corresponds to the total time derivative and thus describes the total change of a quantity at the material point as it moves.

Configuration The derivation of the governing equations can be carried out in the Lagrangian or Eulerian manner. Both formulations are mathematically equivalent but have different advantages and disadvantages for specific problems. In order to describe the unique trajectory of a point in two dimensions by $(x(t),y(t))$, one must know a reference position (X,Y) at a certain point in time. This reference point is often called the initial position in the initial or reference configuration, which is considered at time $t=t_0$. Therefore, the position (x,y) of a material point can also be written as, $(x(X{,}t)\,,y(Y{,}t))$. Hence, the location of that material point is given by its initial position and time t. This perspective is often referred to as the Lagrangian description. Solving the function $(x(X{,}t)\,,y(Y{,}t))$ for X and Y, determines the initial position for each material point from its current position $(x(t)\,, y(t))$ and obtains the functions $X(x(t))$ and $Y(y(t))$. Therefore, the movement is now not explicitly described by (X, Y) but by the current position (x, y) and time. Functions expressed in terms of (X, Y, t) are referred to the Lagrangian and in terms of (x, y,t) to the Eulerian description. The conversion between the differentials of the reference and current configuration in three dimensions can be described as follows

$$\mathrm{d}\mathbf{x} = \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{X}}\,\mathrm{d}\mathbf{X} = \begin{pmatrix} \frac{\partial x}{\partial X} & \frac{\partial x}{\partial Y} & \frac{\partial x}{\partial Z} \\ \frac{\partial y}{\partial X} & \frac{\partial y}{\partial Y} & \frac{\partial y}{\partial Z} \\ \frac{\partial z}{\partial X} & \frac{\partial z}{\partial Y} & \frac{\partial z}{\partial Z} \end{pmatrix} \mathrm{d}\mathbf{X} = \mathbf{F}\mathrm{d}\mathbf{X}\,. \tag{4.4}$$

Where $\mathrm{d}\mathbf{x} = (\mathrm{d}x, \mathrm{d}y, \mathrm{d}z)^T$ is used for the current and correspondingly $\mathrm{d}\mathbf{X} = (\mathrm{d}X, \mathrm{d}Y, \mathrm{d}Z)^T$ for the reference configuration. Here, the matrix $\mathbf{F}$ denotes the deformation gradient describing the whole deformation between the reference and current configuration.

4.2 Conservation Equations

Mass Conservation The domain of the control mass in the reference configuration is denoted by Ω_0 and in the current configuration by Ω_t. All functions considered in the following are assumed to be continuous. Assuming the mass inside the volume to be conserved for all configurations, the following relationship between the reference and current configuration is obtained

$$M = \int_{\Omega_0} \rho\,(\mathbf{X},t_0)\,\mathrm{d}\Omega_0 = \int_{\Omega_t} \rho\,(\mathbf{x},t)\,\mathrm{d}\Omega_t\,. \tag{4.5}$$

Here $\mathbf{x} = (x, y, z)^T$ denotes the three-dimensional coordinate vector. The volume change between the reference and current configuration is described by the determinant of the deformation gradient $J = \det \mathbf{F}$. With this, the material derivation of the mass can now be written as follows,

$$\frac{\mathrm{D}M}{\mathrm{D}t} = \frac{\mathrm{D}}{\mathrm{D}t}\int_{\Omega_t} \rho(\mathbf{x},t)\,\mathrm{d}\Omega_t = \frac{\mathrm{D}}{\mathrm{D}t}\int_{\Omega_0} \rho(\mathbf{X},t)J(\mathbf{X},t)\,\mathrm{d}\Omega_0\,. \tag{4.6}$$

Since Ω_0 and $\mathbf{X} = (X, Y, Z)^T$ do not depend on time, we can continue writing,

$$\frac{\mathrm{D}}{\mathrm{D}t}\int_{\Omega_0} \rho(\mathbf{X},t)J(\mathbf{X},t)\,\mathrm{d}\Omega_0 = \int_{\Omega_0} \left[\frac{\partial \rho}{\partial t}J + \rho\frac{\partial J}{\partial t}\right]\mathrm{d}\Omega_0\,. \tag{4.7}$$

Where we used the fact that the initial coordinates $\mathbf{X}$ do not depend on time. In the following, we omit the coordinates $\mathbf{x}$ and $\mathbf{X}$, since by specifying Ω_0 and Ω_t, respectively, it is clear which approach is used. The time derivative of the determinant of the deformation gradient can be expressed as $\partial J/\partial t = J\,\mathbf{div}(\mathbf{v})$. Thus Eqn. 4.7 can be further written as,

$$\frac{\mathrm{D}M}{\mathrm{D}t} = \int_{\Omega_0} \left[\frac{\partial \rho}{\partial t}J + \rho\,\mathbf{div}\,(\mathbf{v})\,J\right]\mathrm{d}\Omega_0 = \int_{\Omega_t} \left[\frac{\mathrm{D}\rho}{\mathrm{D}t} + \rho\,\mathbf{div}\,(\mathbf{v})\right]\mathrm{d}\Omega_t = 0\,. \tag{4.8}$$

Since the volume Ω_t is arbitrary and under the assumption of continuous integrands, the conservation of mass can be written in differential form as

$$\frac{\mathrm{D}\rho}{\mathrm{D}t} + \rho\,\mathbf{div}\,(\mathbf{v}) = 0\,. \tag{4.9}$$

We obtain another beneficial relationship for densities from Eqn. 4.5 and 4.8 by equating the integrands.

$$\int_{\Omega_0} \rho\,(\mathbf{X},t_0)\,\mathrm{d}\Omega_0 = \int_{\Omega_0} \rho(\mathbf{X},t) J(\mathbf{X},t)\,\mathrm{d}\Omega_0 \implies \rho\,(\mathbf{X},t_0) = \rho(\mathbf{X},t) J(\mathbf{X},t) \tag{4.10}$$

Momentum Equation Using Newton's second law, one can describe the change of momentum through the traction $\mathbf{t}$ acting on its surface and the body forces $\mathbf{b}$ acting over the entire volume. In the following, we consider the volume in the current configuration. Thus the traction can be written as the Cauchy stress tensor $\boldsymbol{\sigma}$ times the normal unit vector $\mathbf{n}$ to the surface $\partial\Omega_t$ of Ω_t, i.e. $\mathbf{t} = \boldsymbol{\sigma}\mathbf{n}$.

$$\frac{\mathrm{D}}{\mathrm{D}t} \int_{\Omega_t} \rho\mathbf{v}\,\mathrm{d}\Omega_t = \int_{\partial\Omega_t} \mathbf{t}\,\mathrm{d}S + \int_{\Omega_t} \rho\mathbf{b}\,\mathrm{d}\Omega_t \tag{4.11}$$

The surface increment of $\partial\Omega_t$ is denoted by $\mathrm{d}S$. Reynold's transport theorem implies that the change of momentum can also be expressed as follows,

$$\frac{\mathrm{D}}{\mathrm{D}t} \int_{\Omega_t} \rho\mathbf{v}\,\mathrm{d}\Omega_t = \int_{\Omega_t} \rho\frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t}\,\mathrm{d}\Omega_t\,. \tag{4.12}$$

This, together with applying the divergence theorem to the surface integral, yields

$$\int_{\Omega_t} \rho\frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t}\,\mathrm{d}\Omega_t = \int_{\Omega_t} \left[\mathbf{div}\,(\boldsymbol{\sigma}) + \rho\mathbf{b}\right]\,\mathrm{d}\Omega_t\,. \tag{4.13}$$

Since Ω_t is arbitrary, we obtain the differential form of the momentum equation

$$\rho\frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t} = \mathbf{div}\,(\boldsymbol{\sigma}) + \rho\mathbf{b}\,. \tag{4.14}$$

The Cauchy stress tensor can be decomposed into pressure and deviatoric stress, i.e. $\boldsymbol{\sigma} = \boldsymbol{\tau} - p\mathbf{I}$ with the identity tensor $\mathbf{I}$. With this decomposition Eqn. 4.14 results in

$$\rho\frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t} = -\mathbf{grad}\,(p\mathbf{I}) + \mathbf{div}\,(\boldsymbol{\tau}) + \rho\mathbf{b} \tag{4.15}$$

where

$$\boldsymbol{\tau} = \begin{pmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{pmatrix} \tag{4.16}$$

and the divergence of τ is taken column-wise resulting in the vector.

$$\mathbf{div}\,(\boldsymbol{\tau}) = \begin{pmatrix} \dfrac{\partial \tau_{xx}}{\partial x} + \dfrac{\partial \tau_{yx}}{\partial y} + \dfrac{\partial \tau_{zx}}{\partial z} \\ \dfrac{\partial \tau_{xy}}{\partial x} + \dfrac{\partial \tau_{yy}}{\partial y} + \dfrac{\partial \tau_{zy}}{\partial z} \\ \dfrac{\partial \tau_{xz}}{\partial x} + \dfrac{\partial \tau_{yz}}{\partial y} + \dfrac{\partial \tau_{zz}}{\partial z} \end{pmatrix} \tag{4.17}$$

Note that pressures are defined to be positive and stresses negative in compression. Similarly, conservation of momentum can be derived in Lagrangian coordinates (Belytschko, Liu, *et al.*, 2013). Here, the first Piola-Kirchhoff $\mathbf{P}$ stress tensor is used instead of the Cauchy. These can be transformed into each other using the deformation gradient and its determinant, i.e. $\mathbf{PF^T} = J\boldsymbol{\sigma}$. The momentum equation can then be written as

$$\rho(\mathbf{X},t)\frac{\mathrm{D}v}{\mathrm{D}t}(\mathbf{X},t) = \mathbf{div}(\mathbf{P}(\mathbf{X},t)) + \rho(\mathbf{X},t)\mathbf{b}(\mathbf{X},t)\,. \tag{4.18}$$

4.2.1 Dry porous media

We now consider porous media occupying the domain $\Omega = \Omega_s \cup \Omega_v$, with Ω_s being the domain fraction of the solid material and Ω_v the fraction of voids. We denote the volume of a domain Ω by $|\Omega|$. In the following, the Lagrangian description is applied. The conservation equations are then valid for the partial density $\overline{\rho} = (1-n)\rho$, with the porosity $n = |\Omega_v|/|\Omega|$ and the solid density ρ. In geotechnical applications, the solid is represented by soil grains, which are often assumed to be incompressible, equivalent to ρ and $|\Omega_s|$ being constant. Therefore, the mass conservation, as shown in Eqn. 4.10, needs to be modified as the initial $|\Omega^0|$ and the current volume $|\Omega|$ are now related as

$$|\Omega| = |\Omega_s| + |\Omega_v| = |\Omega_s| + J|\Omega_v^0|\,. \tag{4.19}$$

We consider the one-dimensional case and a linear elastic constitutive behaviour for simplicity. The linear-elastic constitutive behaviour for uniaxial (in x-direction) tension and compression is expressed as $\boldsymbol{\sigma} = \sigma_{xx} = E_s\epsilon_{xx} = E_s\boldsymbol{\epsilon}$ with the elastic modulus E_s. The strain $\boldsymbol{\epsilon}$ is derived from the deformation gradient. It should be noted that linear elasticity in reality applies only to very small strains and is used here only as an academic example. In the one-dimensional case, the deformation gradient is obtained as

$$\mathbf{F} = \mathrm{F}_{xx} = 1 + \frac{\partial u}{\partial X}\,. \tag{4.20}$$

The polar decomposition of the deformation gradient decomposes $\mathbf{F} = \mathbf{RU}$ in a rotation (rigid motion) $\mathbf{R}$ and deformation $\mathbf{U}$. In one dimension, we consider the case $\mathbf{R} = 1$ and therefore obtain $\mathbf{F} = \mathbf{U}$. The strain can then be defined as $\boldsymbol{\epsilon} = \mathbf{U} - 1 = \frac{\partial u}{\partial X}$, which in the considered case will correspond to the linear strain. The volumetric strain ϵ_v, which describes volumetric changes, can be determined from the determinant of the deformation gradient $\epsilon_v = \mathbf{det}(\mathbf{F}) = J = 1 + \frac{\partial u}{\partial X}$. Therefore, the relation between Cauchy and the first Piola-Kirchhoff stress simplifies to $\mathbf{P} = \boldsymbol{\sigma}$. Using the constitutive model and strain $\boldsymbol{\epsilon} = \mathbf{U} - 1 = \frac{\partial u}{\partial X}$, Eqn. 4.18 simplifies then to a scalar equation

$$(1-n)\rho \frac{\partial^2 u}{\partial t^2} = E_s \frac{\partial^2 u}{\partial X^2} + (1-n)\rho b \,, \tag{4.21}$$

where, the relation between displacement and velocity $\partial u / \partial t = v$ was used. Eqn. 4.21 is the inhomogeneous wave equation, which leads to solutions in the form of propagating waves for given boundary and initial conditions. Let us consider two simplified cases of Eqn. 4.21. The simplest case is the static one ($\frac{\partial^2 u}{\partial t^2} = 0$) without body forces ($b = 0$), which leads to the homogeneous ordinary differential equation in terms of the stress $\boldsymbol{\sigma}$,

$$\frac{\partial \boldsymbol{\sigma}}{\partial X} = 0 \,. \tag{4.22}$$

For given constant boundary condition $\boldsymbol{\sigma}(X = 0) = \boldsymbol{\sigma}_0$, the solution is $\boldsymbol{\sigma}(X) = \boldsymbol{\sigma}_0$. The second case is the static equation with body forces. This could for example, model a vertical column of dry soil under gravity, with $b = -(1-n)\rho g$ and and the boundary condition $\boldsymbol{\sigma}(X = 0) = \boldsymbol{\sigma}_0$,

$$\frac{\partial \boldsymbol{\sigma}}{\partial X} = -(1-n)\rho g \,. \tag{4.23}$$

The solution to Eqn. 4.23 is $\boldsymbol{\sigma}(X) = -(1-n)gX + \boldsymbol{\sigma}_\mathbf{0}$. In the case of large deformations, the boundary at $X = 0$, where the load is applied, moves as the material deforms according to the constitutive model. In the case of linear elasticity, the boundary will move exactly $\Delta X = \boldsymbol{\sigma}_0 / (E_s \, L)$, where L represents the length of the considered interval. A common assumption in soil mechanics is that soil grains can be considered nearly incompressible, which means ρ in Eqn. 4.23 is approximately constant. Therefore any volume change leads to changes in the porosity n, following the idea that volume changes in soil are mainly due to the rearrangement of the grains. This is equivalent to the volume transformation described in Eqn. 4.19. So far, the solutions of Eqn. 4.22 and Eqn. 4.23 were derived on the assumption that n is constant. To obtain the solution for a changing porosity, we consider Eqn. 4.19 for the relation between initial volumes and porosity ($|\Omega^0|, |\Omega_v^0|, n^0$) and final ones ($|\Omega|, |\Omega_v|, n$), where the volume $|\Omega^s|$ is assumed to be constant due to the incompressibility of the soil grains

$$n = \frac{|\Omega_v^t|}{|\Omega_s| + |\Omega_v^t|} = \frac{J|\Omega_v^0|}{|\Omega_s| + J|\Omega_v^0|} = \frac{\left(1 + \frac{\partial u}{\partial X}\right) n^0}{1 + \frac{\partial u}{\partial X} n^0} = \frac{(1 + \boldsymbol{\epsilon})\, n^0}{1 + \boldsymbol{\epsilon} n^0} \,. \tag{4.24}$$

Eqn. 4.24 replaces the mass conservation by describing the changes of $|\Omega_v|$.

Example: One-dimensional compression In the following, we consider a one-dimensional compaction example and determine some unknowns just from geometry. Afterwards, we try to formulate the obtained results in the sense of a boundary value problem (BVP) based on Eqn. 4.23 and 4.24. We consider the one-dimensional problem shown in Fig. 4.2.

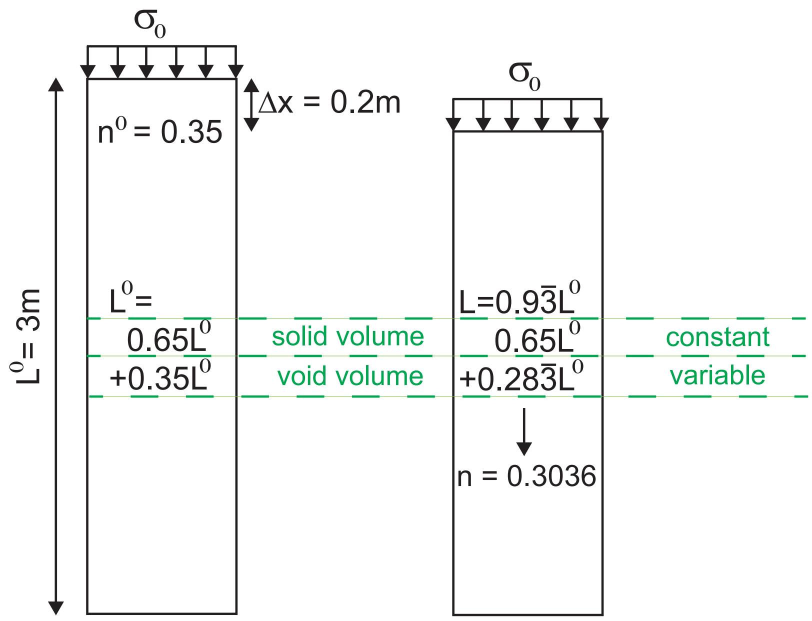


Figure 4.2: Example of the one-dimensional compaction of a porous soil column.

If we know the initial porosity n^0 and the settlement of the column ΔX, we can calculate the final porosity. Let us now consider the shown scenario as a BVP. Assuming $X = 0$ to define the top of the undeformed column and knowing the settlement to be $\Delta X = 0.2\,\mathrm{m}$, we can choose either the E_s or $\boldsymbol{\sigma}_0$ to obtain the other. Choosing $\boldsymbol{\sigma}_0 = -10000\,\mathrm{kN/m^2}$ results in an elastic modulus $E_s = 50000\,\mathrm{kN/m^2}$. Setting the grain density to $\rho = 2650\,\mathrm{kN/m^3}$ and $g = 9.81\,\mathrm{m/s^2}$ the BVP is complete. First we use Eqn. 4.24 to calculate the final porosity

$$n = \frac{\left(1 + \frac{\Delta X}{n^0 L_0}\right) n^0}{1 + \frac{\Delta X}{n^0 L_0} n^0} = \frac{(1 - 0.1905) \cdot 0.35}{1 - 0.1905 \cdot 0.35} = 0.3036\,. \tag{4.25}$$

The calculated porosity in Eqn. 4.25 is the same as the geometrical conclusion indicated in Fig. 4.2. We calculated the strain in Eqn. 4.25 as $\boldsymbol{\epsilon} = \frac{\Delta X}{n^0 L_0}$. Inserting the final porosity in the solution for Eqn. 4.23 and taking into account that due to the movement of the boundary the solution is now defined on the interval $[\Delta X\,, 3]$, we obtain the stress $\boldsymbol{\sigma}(X)$. As shown in Fig. 4.3, there is a significant difference in the solutions if a constant porosity is considered. This is also true if the porosity is calculated within a constitutive model and is not changed accordingly in the governing equations. The stress at the top and bottom of the column in both cases in Fig. 4.3 is the same, but the column length is different. The shown scenario should emphasize the need for updated porosity in cases of large deformations.

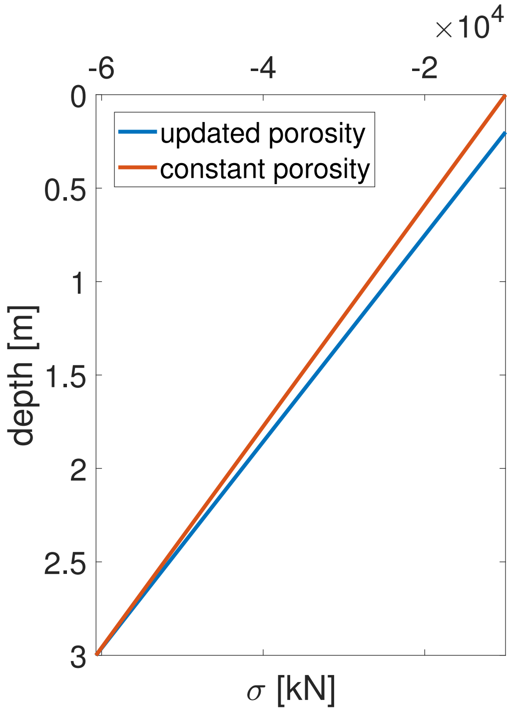


Figure 4.3: Comparison of solutions for the one-dimensional compaction:
1. Using Eqn. 4.23 and 4.24 (blue)
2. Using Eqn. 4.23 with constant porosity (red)

4.2.2 Two-Phase-Formulation

Let us now consider two materials occupying the volumes V_s and V_f. The total volume is $V = V_s \cup V_f$. The indices s and f correspond to solid and fluid materials. Defining the porosities $n_s = V_s/V$ and $n_f = V_s/V$, mass conservation in Eqn. 4.9 can be written in terms of partial densities $\overline{\rho}_\alpha = \rho_\alpha n_\alpha$,

$$\frac{\mathrm{D}(\rho_s n_s)}{\mathrm{D}t} + \rho_s n_s \mathbf{div}\,(\mathbf{v}_s) = 0 \tag{4.26a}$$

$$\frac{\mathrm{D}(\rho_f n_f)}{\mathrm{D}t} + \rho_f n_f \mathbf{div}\,(\mathbf{v}_f) = 0\,. \tag{4.26b}$$

For a saturated materials, $n_s + n_f = 1$ holds. In the case of mass transfer, the right-hand side of Eqn. 4.26a and 4.26b contains source and sink terms instead of zeros. Sometimes it is useful to define the substantial derivative $\mathrm{D}^\alpha/\mathrm{D}t$ for a scalar function f in the direction $\mathbf{v}^\alpha$ of the phase α

$$\frac{\mathrm{D}^\alpha f}{\mathrm{D}t} := \frac{\partial f}{\partial t} + \mathbf{grad}(f) \cdot \mathbf{v}_\alpha \tag{4.27}$$

In general, a fluid particle moves in a different direction compared to a soil particle. Therefore, Eqn. 4.26a and 4.26b describe the motion of particles along two different velocity fields $\mathbf{v}_s$ and $\mathbf{v}_f$.

Incompressible Conditions Assuming that the densities of the soil and fluid particles are constant (both are incompressible), Eqn. 4.26a and 4.26b describe the change in porosity in the respective directions $\mathbf{v}_s$ and $\mathbf{v}_f$.

$$\frac{\mathrm{D}^s(n_s)}{\mathrm{D}t} + n_s\mathbf{div}\,(\mathbf{v}_s) = \frac{\partial n_s}{\partial t} + \mathbf{grad}\,(n_s)\cdot\mathbf{v}_s + n_s\mathbf{div}\,(\mathbf{v}_s) = 0 \tag{4.28a}$$

$$\frac{\mathrm{D}^f(n_f)}{\mathrm{D}t} + n_f\mathbf{div}\,(\mathbf{v}_f) = \frac{\partial n_f}{\partial t} + \mathbf{grad}\,(n_f)\cdot\mathbf{v}_f + n_f\mathbf{div}\,(\mathbf{v}_f) = 0 \tag{4.28b}$$

Using the saturation condition $n_s + n_f = 1$, the sum of Eqn. 4.28a and 4.28b yield the following expression

$$\mathbf{grad}\,(n_f)\cdot(\mathbf{v}_f - \mathbf{v}_s) + (1 - n_f)\,\mathbf{div}\,(\mathbf{v}_s) + n_f\mathbf{div}\,(\mathbf{v}_f) = 0\,. \tag{4.29}$$

The partial time derivatives cancel each other out due to opposite signs. It should be noted that the assumption of incompressible constituents and no source or sink terms in the mass conservation leads to no changes in the porosity, as nothing can compress or extend in the control mass, restricting the formulation to volume preserving deformations. Therefore Eqn. 4.29 is satisfied because each summand is equal zero. In Sec. 4.2.1 the pore space could change as the void was not containing any material. In the presence of a incompressible pore fluid with conserved mass this it not the case any more.

Slightly Compressible Fluid The assumption of incompressibility is now weakened to calculate the pressure of the pore fluid. Applying the product rule, Eqn. 4.26b can be rewritten as

$$\begin{aligned}&\frac{\mathrm{D}^f(n_f\rho_f)}{\mathrm{D}t} + (n_f\rho_f)\cdot\mathbf{div}\,(\mathbf{v}_f)\\ =&\frac{\partial(n_f\rho_f)}{\partial t} + \mathbf{grad}\,(n_f\rho_f)\cdot\mathbf{v}_f + (n_f\rho_f)\mathbf{div}\,(\mathbf{v}_f) = 0\,.\end{aligned} \tag{4.30}$$

We assume the pore fluid to be compressible having compressibility C_f,

$$\frac{\partial p_f}{\partial t} = \frac{1}{\rho_f C_f}\frac{\partial \rho_f}{\partial t}\,. \tag{4.31}$$

Replacing the time derivatives of ρ_f in Eqn. 4.30 by Eqn. 4.31 leads to

$$\begin{aligned}\rho_f\frac{\partial n_f}{\partial t} &+ n_f\rho_f C_f\frac{\partial p_f}{\partial t} + n_f\mathbf{grad}\,(\rho_f)\cdot\mathbf{v}_f\\ &+ \rho_f\mathbf{grad}\,(n_f)\cdot\mathbf{v}_f + n_f\rho_f\mathbf{div}\,(\mathbf{v}_f) = 0\,.\end{aligned} \tag{4.32}$$

Under the assumption that the expression $n_f\mathbf{grad}\,(\rho_f)\cdot\mathbf{v}_f$ is negligibly small (Verruijt, 2010), the sum of Eqn. 4.28a and 4.32 yields

$$\begin{aligned}n_f C_f\frac{\partial p_f}{\partial t} &+ \mathbf{grad}\,(n_f)\cdot(\mathbf{v}_f - \mathbf{v}_s)\\ &+ (1 - n_f)\,\mathbf{div}\,(\mathbf{v}_s) + n_f\mathbf{div}\,(\mathbf{v}_f) = 0\,.\end{aligned} \tag{4.33}$$

With known velocities $\mathbf{v}_s$ and $\mathbf{v}_f$, Eqn. 4.33 can be used to calculate the pore water pressure as

$$\frac{\partial p_f}{\partial t} = -\frac{1}{n_f C_f}\Big(\mathbf{grad}\,(n_f) \cdot (\mathbf{v}_f - \mathbf{v}_s) + (1 - n_f)\,\mathbf{div}\,(\mathbf{v}_s) + n_f \mathbf{div}\,(\mathbf{v}_f)\Big)\,. \tag{4.34}$$

Momentum Equation Let us again consider two materials occupying the volumes V_s and V_f. Similar to the mass conservation, we rewrite the momentum Eqn. 4.14 for both materials. In contrast to mass conservation, we allow momentum exchange between both materials. This exchange is denoted by force $\mathbf{f}_d$ and it is acting with opposite signs on both materials, as

$$n_s\rho_s \frac{\mathrm{D}^s\mathbf{v}_s}{\mathrm{D}t} = \mathbf{div}\,(\boldsymbol{\sigma}_s) + n_s\rho_s\mathbf{b} - \mathbf{f}_d\,, \tag{4.35a}$$

$$n_f\rho_f \frac{\mathrm{D}^f\mathbf{v}_f}{\mathrm{D}t} = \mathbf{div}\,(\boldsymbol{\sigma}_f) + n_f\rho_f\mathbf{b} + \mathbf{f}_d\,. \tag{4.35b}$$

The stress tensors $\boldsymbol{\sigma}_s$ and $\boldsymbol{\sigma}_f$ represent the partial stresses for the solid and liquid. They can be additively decomposed (Bandara, 2013) into the following parts

$$\boldsymbol{\sigma}_f = n_f\boldsymbol{\tau}_f - n_f p_f\mathbf{I} \quad \text{and} \quad \boldsymbol{\sigma}_s = \boldsymbol{\sigma}' - n_s p_s\mathbf{I}\,. \tag{4.36}$$

For saturated conditions, the assumption is made that the pressure p_s exerted by the solid skeleton on the pore fluid is equal to the pore water pressure p_f (X. Zhang *et al.*, 2017). Furthermore, the viscous stresses $\boldsymbol{\tau}_f$ can be neglected for the pore fluid. Conservation of angular momentum leads to a symmetric effective stress tensor $\boldsymbol{\sigma}'$ and therefore, a symmetric tensor $\boldsymbol{\sigma}_s$. If not mentioned otherwise, the stresses are expressed in terms of Cauchy stresses. With these assumptions, Eqn. 4.35a and 4.35b can be rewritten as

$$n_s\rho_s \frac{\mathrm{D}^s\mathbf{v}_s}{\mathrm{D}t} = \mathbf{div}\,(\boldsymbol{\sigma}') - \mathbf{grad}\,(n_s\,p_f) + n_s\rho_s\mathbf{b} - \mathbf{f}_d \tag{4.37a}$$

$$n_f\rho_f \frac{\mathrm{D}^f\mathbf{v}_f}{\mathrm{D}t} = -\mathbf{grad}\,(n_f\,p_f) + n_f\rho_f\mathbf{b} + \mathbf{f}_d\,. \tag{4.37b}$$

The Drag-Force $\mathbf{f}_d$ describes the coupling between soil and the pore fluid, which is caused by the friction or resistance $\overline{\mathbf{f}}_d$ and buoyancy force $p_f\,\mathbf{grad}(n_f)$ (Drumheller, 2000),

$$\mathbf{f}_d = \overline{\mathbf{f}}_d + p_f\,\mathbf{grad}(n_f)\,. \tag{4.38}$$

Substituting Eqn. 4.38 into Eqn. 4.37a and 4.37b and applying the product rule leads to the following system of equations

$$n_s\rho_s \frac{\mathrm{D}^s\mathbf{v}_s}{\mathrm{D}t} = \mathbf{div}\,(\boldsymbol{\sigma}') - n_s\mathbf{grad}\,(p_f) + n_s\rho_s\mathbf{b} - \overline{\mathbf{f}}_d \tag{4.39a}$$

$$n_f\rho_f \frac{\mathrm{D}^f\mathbf{v}_f}{\mathrm{D}t} = -n_f\mathbf{grad}\,(p_f) + n_f\rho_f\mathbf{b} + \overline{\mathbf{f}}_d\,. \tag{4.39b}$$

The force $\overline{\mathbf{f}}_d$ acts at the contact surfaces between the pore fluid and the soil skeleton. In the following, we chose the Drag-Force model according to Ergun (1952). Nevertheless, it should be noted that there are also different approaches to model the Drag-Force, for example, presented in Schiller (1933) and Koch and Hill (2001) and others. In the following, we consider

$$\overline{\mathbf{f}}_d = -n_f^2 \left[\frac{\mu}{\kappa} + n_f \rho_f \frac{F}{\sqrt{\kappa}} \left| \mathbf{v}_f - \mathbf{v}_s \right| \right] \left(\mathbf{v}_f - \mathbf{v}_s \right) \tag{4.40}$$

$$\text{with the coefficient} \quad F = \frac{B}{\sqrt{A} n_f^{3/2}} \, .$$

The constants A and B are 150 and 1.75 according to Ergun (1952) and μ denotes the dynamic viscosity of the fluid. The intrinsic permeability κ in Eqn. 4.40 can be calculated by the Kozeny-Carman formula (Bear, 1988) as

$$\kappa = \frac{D_p^2}{A} \frac{n_f^3}{(1 - n_f)^2} \, , \tag{4.41}$$

where the coefficient D_p describes the mean diameter of soil particles, which for the sake of simplicity are assumed to be spherical. If the second part in the square brackets in Eqn. 4.40 and all inertia forces are neglected, Darcy's law for steady flows can be recovered from the momentum Eqn. 4.39b as

$$\mathbf{grad}\left(p_f\right) = -\frac{\mu}{\kappa} \underbrace{n_f \left(\mathbf{v}_f - \mathbf{v}_s\right)}_{\substack{=:\mathbf{q} \\ \text{specific discharge}}} + \rho_f \mathbf{b} \, . \tag{4.42}$$

In the case of a rigid grain skeleton, $\mathbf{v}_s$ is equal to zero and thus Eqn. 4.42 is sufficient to completely describe the flow of the pore fluid. Setting $\mathbf{v}_s$ equal to zero in Eqn. 4.42 and solving for the velocity of the pore fluid results in (Boer, 2000)

$$\mathbf{v}_f = \frac{\kappa}{\mu} \left(-\mathbf{grad}\left(p_f\right) + \rho_f \mathbf{b} \right) \, . \tag{4.43}$$

Small Strains Now we consider situations with small deformations, where the derived equations can be simplified further. Therefore, we assume the porosity to be constant. It follows that all parts containing the derivatives of porosity are omitted. In addition, the intrinsic permeability in Eqn. 4.41 is assumed to be constant. Because of the buoyancy force in Eqn. 4.38, the gradient of porosity no longer appears in Eqn. 4.39a and 4.39b. In Eqn. 4.34 on the other hand, the gradient of porosity disappears, which leads to the following equation

$$\frac{\partial p_f}{\partial t} = -\frac{1}{n_f C_f} \Big(\left(1 - n_f\right) \mathbf{div}\left(\mathbf{v}_s\right) + n_f \mathbf{div}\left(\mathbf{v}_f\right) \Big) \, . \tag{4.44}$$

Time integration of Eqn. 4.44 leads to an explicit expression for the pore water pressure p_f. By integrating Eqn. 4.44 and assuming integration constants to be zero (excess pore pressure), the velocities $\mathbf{v}_s$ and $\mathbf{v}_f$ are replaced by the corresponding displacements $\mathbf{u}_s$ and $\mathbf{u}_f$ as

$$p_f = -\frac{1}{n_f C_f}\big((1-n_f)\,\mathbf{div}\,(\mathbf{u}_s) + n_f\mathbf{div}\,(\mathbf{u}_f)\big)\,. \tag{4.45}$$

Neglecting the body forces and the second part in the square brackets in Eqn. 4.40, Eqn. 4.45 can be substituted into the Eqn. 4.39a and 4.39b as

$$\begin{aligned} n_s\rho_s\frac{\mathrm{D}^s\mathbf{v}_s}{\mathrm{D}t} &= \mathbf{div}\,(\boldsymbol{\sigma}') + \frac{1-n_f}{n_f C_f}\mathbf{grad}\big((1-n_f)\,\mathbf{div}\,(\mathbf{u}_s) + n_f\mathbf{div}\,(\mathbf{u}_f)\big) \\ &\quad + n_f^2\frac{\mu}{\kappa}(\mathbf{v}_f - \mathbf{v}_s) \end{aligned} \tag{4.46a}$$

$$\begin{aligned} n_f\rho_f\frac{\mathrm{D}^f\mathbf{v}_f}{\mathrm{D}t} &= \frac{1}{C_f}\mathbf{grad}\big((1-n_f)\,\mathbf{div}\,(\mathbf{u}_s) + n_f\mathbf{div}\,(\mathbf{u}_f)\big) \\ &\quad - n_f^2\frac{\mu}{\kappa}(\mathbf{v}_f - \mathbf{v}_s)\,. \end{aligned} \tag{4.46b}$$

In some formulations, the momentum balance of the entire mixture is used, which is obtained from the sum of Eqn. 4.46a and 4.46b as

$$\begin{aligned} n_s\rho_s\frac{\mathrm{D}^s\mathbf{v}_s}{\mathrm{D}t} + n_f\rho_f\frac{\mathrm{D}^f\mathbf{v}_f}{\mathrm{D}t} &= \mathbf{div}\,(\boldsymbol{\sigma}') \\ + \left(\frac{1}{n_f C_f}\right)\mathbf{grad}\big((1-n_f)\,&\mathbf{div}\,(\mathbf{u}_s) + n_f\mathbf{div}\,(\mathbf{u}_f)\big)\,. \end{aligned} \tag{4.47}$$

Constitutive Behaviour A constitutive model for the solid skeleton can express the divergence of effective stresses depending on displacements. We consider the case of small deformations and assume an approximate linear material behaviour. Therefore, the generalised Hook's law can be applied to calculate the entries of $\boldsymbol{\sigma}'$ as follows

$$\begin{pmatrix} \sigma'_{xx} & \sigma'_{xy} & \sigma'_{xz} \\ \cdots & \sigma'_{yy} & \sigma'_{yz} \\ \cdots & \cdots & \sigma'_{zz} \end{pmatrix} = \frac{E}{(1-\nu)}\begin{pmatrix} \left(\frac{1-\nu}{1-2\nu}\right)\varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \cdots & \left(\frac{1-\nu}{1-2\nu}\right)\varepsilon_{yy} & \varepsilon_{yz} \\ \cdots & \cdots & \left(\frac{1-\nu}{1-2\nu}\right)\varepsilon_{zz} \end{pmatrix}. \tag{4.48}$$

In Eqn. 4.48, ν is the Poisson's ratio and E the Young's modulus. The matrix on the right-hand side of Eqn. 4.48 represents the strain tensor $\boldsymbol{\epsilon}$, which for small strains can be calculated from the displacements as follows

$$\begin{pmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz} \end{pmatrix} = \begin{pmatrix} \frac{\partial u_{sx}}{\partial x} & \frac{1}{2}\left(\frac{\partial u_{sx}}{\partial y} + \frac{\partial u_{sy}}{\partial x}\right) & \frac{1}{2}\left(\frac{\partial u_{sx}}{\partial z} + \frac{\partial u_{sz}}{\partial x}\right) \\ \frac{1}{2}\left(\frac{\partial u_{sy}}{\partial x} + \frac{\partial u_{sx}}{\partial y}\right) & \frac{\partial u_{sy}}{\partial y} & \frac{1}{2}\left(\frac{\partial u_{sy}}{\partial z} + \frac{\partial u_{sz}}{\partial y}\right) \\ \frac{1}{2}\left(\frac{\partial u_{sz}}{\partial x} + \frac{\partial u_{sx}}{\partial z}\right) & \frac{1}{2}\left(\frac{\partial u_{sz}}{\partial y} + \frac{\partial u_{sy}}{\partial z}\right) & \frac{\partial u_{sz}}{\partial z} \end{pmatrix}. \tag{4.49}$$

A double index is used for the components of the displacement vector, in which the first subscript indicates whether it is solid or fluid and second, the Cartesian coordinate, $\mathbf{u}_s = (u_{sx}, u_{sy}, u_{sz})^T$. The strain tensor in Eqn. 4.49 is symmetric so only six components need to be calculated. Substituting Eqn. 4.49 into the right-hand side of Eqn. 4.48 leads to

$$\begin{pmatrix} \sigma'_{xx} & \sigma'_{xy} & \sigma'_{xz} \\ \cdots & \sigma'_{yy} & \sigma'_{yz} \\ \cdots & \cdots & \sigma'_{zz} \end{pmatrix} = \begin{pmatrix} \frac{E}{1-2\nu}\frac{\partial u_{sx}}{\partial x} & \frac{E}{2(1-\nu)}\left(\frac{\partial u_{sx}}{\partial y} + \frac{\partial u_{sy}}{\partial x}\right) & \frac{E}{2(1-\nu)}\left(\frac{\partial u_{sx}}{\partial z} + \frac{\partial u_{sz}}{\partial x}\right) \\ \cdots & \frac{E}{1-2\nu}\frac{\partial u_{sy}}{\partial y} & \frac{E}{2(1-\nu)}\left(\frac{\partial u_{sy}}{\partial z} + \frac{\partial u_{sz}}{\partial y}\right) \\ \cdots & \cdots & \frac{E}{1-2\nu}\frac{\partial u_{sz}}{\partial z}\,. \end{pmatrix} \tag{4.50}$$

The divergence of $\boldsymbol{\sigma}'$ is calculated according to Eqn. 4.17. The derived equations for small deformations correspond to the so-called w-v-Formulation, first derived by Jong (1956). In the same year Biot (1956a) published his equations of poroelasticity. In the following section we will show that the w-v-Formulation can be related to Biot's equations of poroelasticity.

4.3 Comparison with Biot‘s Poroelasticity

For simplicity, we will only consider the one-dimensional case. There is no transverse deformation in the one-dimensional case, which corresponds to Poisson's ratio of $\nu = 0$. Thus, the stress-strain relation $\sigma'_{xx} = E\frac{\partial u_{sx}}{\partial x}$ is derived from Eqn. 4.49. Furthermore, the following relations apply to the differential operators in the one-dimensional case,

$$\mathbf{grad}(\mathbf{div}(u)) = \frac{\partial^2 u}{\partial x^2} \quad \text{and} \quad \mathbf{div}\left(\frac{\partial u}{\partial x}\right) = \frac{\partial^2 u}{\partial x^2}\,. \tag{4.51}$$

In the following, we denote the velocities by $v_{sx} = w$, $v_{fx} = v$ and the displacements by $u_{sx} = W$, $u_{fx} = V$, which corresponds to the notation of the w-v-Formulation in Verruijt (2010). Furthermore, the volume fractions n_f and n_s are expressed as n and $1-n$. Using these relations for Eqn. 4.46a and 4.47, leads to the following system of partial differential equations,

$$\rho_f \frac{\mathrm{D}v}{\mathrm{D}t} = \frac{1}{C_f}\left(\frac{\partial^2 V}{\partial x^2} + \frac{(1-n)}{n}\frac{\partial^2 W}{\partial x^2}\right) - \frac{n\mu}{\kappa}(v-w) \tag{4.52a}$$

$$(1-n)\,\rho_s\frac{\mathrm{D}w}{\mathrm{D}t} + n\rho_f\frac{Dv}{\mathrm{D}t} = \frac{1}{C_f}\frac{\partial^2 V}{\partial x^2} + \left(E + \frac{(1-n)}{C_f n}\right)\frac{\partial^2 W}{\partial x^2}\,. \tag{4.52b}$$

Next, we consider the one-dimensional Biot's equations as described in Simon *et al.* (1984). This is the so-called u-U-Formulation. In this case, the displacements u_s and u_f are denoted by u and U. In addition, dots above the displacements denote the substantial or total time derivatives, respectively. The u-U-Formulation is written as

$$\rho_f \ddot{U} = \alpha Q n\left(\frac{1}{\alpha}\frac{\partial^2 U}{\partial x^2} + \left(\frac{1}{n} - \frac{1}{\alpha}\right)\frac{\partial^2 u}{\partial x^2}\right) - \frac{n}{k}\left(\dot{U} - \dot{u}\right) \tag{4.53a}$$

$$(1-n)\,\rho_s\ddot{u} + \rho_f n\ddot{U} = \alpha Q n\frac{\partial^2 U}{\partial x^2} + \left(\lambda + 2\mu + \alpha^2 Q - \alpha Q n\right)\frac{\partial^2 u}{\partial x^2}\,. \tag{4.53b}$$

We now can identify the coefficients introduced in Eqn. 4.53a and 4.53b with the ones in Eqn. 4.52a and 4.52b. The corresponding coefficients that are equal are marked with the same colour. The coefficients λ and μ denote the Lamé parameters. The coefficient α in Biot's equations describes the relationship between the compressibility of the porous material and solid grains. This coefficient equals to one if the grains are assumed to be incompressible, as in the previous section. This results in the following relationships (for $\alpha = 1$) between the coefficients:

$$Q = \frac{1}{n \cdot C_f} \qquad \text{and} \qquad \lambda + 2\mu = E\,. \tag{4.54}$$

In addition, we define two new coefficients $\varkappa$ and β as,

$$\varkappa := \frac{Q}{\lambda + 2\mu + \alpha^2 Q} \qquad \text{and} \qquad \beta := \frac{\rho_f}{\rho}\,, \tag{4.55}$$

where, ρ is the density of the mixture $\rho = (1-n)\rho_s + n\rho_f$. These coefficients are helpful for later examinations in the following section.

4.4 Analytical Investigations

Now, we consider equations derived in the previous section. The existence of an analytical solution offers one possibility to check the numerical methods for their correctness. The starting point of this research were the formulations introduced by Biot (1956a), Biot (1956b), and Jong (1956). Biot's model is one of the most common models for simulating saturated soils. Many authors have been involved in the research of an analytical solution to Biot's equations. First, analytical solutions were derived for specific scenarios, such as strong or weak coupling of the two phases (Garg, Nayfeh, *et al.*, 1974; Garg, Brownell Jr, *et al.*, 1975; Hong *et al.*, 1988) and later also for more general cases (Simon *et al.*, 1984; Verruijt, 2010). For the analytical solutions, a linear-elastic material behaviour for the soil skeleton is assumed. While in the formulations of Biot, the displacements are the primary unknowns, Jong (1956) and later Verruijt (2010) have presented an equivalent formulation for the velocities. Compared to Biot's equations, one of the differences is that by using velocities as primary variables, a system of differential equations of first order in time is obtained. This fact facilitates the numerical integration of the equations.

Fourier Approach In Zienkiewicz, Chang, *et al.* (1980) a frequency-dependent classification of the different mathematical models for the poroelastic column was made using harmonically oscillating boundary conditions. Verruijt (2010) presented an analytical solution to the problem in Eqn. 4.52a and 4.52b, assuming solutions of the form:

$$v = V e^{i(\Lambda x - \omega t)} \text{ and } w = W e^{i(\Lambda x - \omega t)}\,, \tag{4.56}$$

where, ω is a given frequency and Λ is unknown. Zienkiewicz, Chang, *et al.* (1980) and Verruijt (2010) used the Fourier approach to obtain a solution in the complex domain.

For the one-dimensional wave propagation, these solutions could predict the existence of two compressional waves for specific material properties. How pronounced the second compressional wave is, depends mainly on the permeability of the soil. Verruijt (2010) derived some beneficial relationships of the analytical solution for specific cases that can be used to verify numerical results. In the following, we will refer to the two compressional waves as the drained and undrained waves. In the case of incompressible grains $\alpha = 1$, the propagation velocities of both waves are

$$C_1 = \sqrt{\frac{1}{\rho m_v} + \frac{1}{\rho n C_f}} \quad \text{and} \quad C_2 = \sqrt{\frac{1}{C_f \rho_f} \cdot \frac{n C_f}{(1-n) m_v + n C_f}} \tag{4.57}$$

$$\text{with } m_v = 1/E \,,$$

where, C_1 refers to the undrained wave and C_2 refers to the drained wave, as presented in Verruijt (2010). Furthermore, Verruijt (2010) derived the distribution of effective stress and pore pressure in the undrained wave due to a suddenly applied constant load σ_0 on the solid skeleton as

$$p = \frac{\alpha m_v}{\alpha^2 m_v + S_p} \sigma_0 \quad \text{and} \quad \sigma' = \frac{S_p}{\alpha^2 m_v + S_p} \sigma_0 \,. \tag{4.58}$$

Laplace Approach Considering Eqn. 4.53a and 4.53b a solution in the time-domain based on Laplace Transformation was presented by Simon *et al.* (1984). In the following, we briefly show the procedures that were used for this purpose. First, we define the constant $V_c = \sqrt{(\lambda + 2\mu + \alpha^2 Q)/\rho}$ for a simpler representation. Then $\varkappa$ from Eqn. 4.55 can be written in terms of V_c as $\varkappa = Q/(\rho V_c^2)$. Following Simon *et al.* (1984), we perform a coordinate transformation with the new variables:

$$\xi = \frac{x}{\rho\, k\, V_c} \qquad \text{and} \qquad \tau = \frac{t}{\rho\, k} \,. \tag{4.59}$$

The first derivatives of the displacements with respect to the new variables are calculated as

$$\begin{aligned} \frac{\partial u}{\partial \xi}(x,t) &= \frac{\partial u}{\partial x}\frac{\partial x}{\partial \xi} = \rho\, k\, V_c\, \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial \tau}(x,t) &= \frac{\partial u}{\partial t}\frac{\partial t}{\partial \tau} = \rho\, k\, \frac{\partial u}{\partial t} \end{aligned} \tag{4.60}$$

and the second derivatives as

$$\begin{aligned} \frac{\partial^2 u}{\partial \xi^2}(x,t) &= \frac{\partial}{\partial \xi}\left(\rho\, k\, V_c\, \frac{\partial u}{\partial x}\right) = \rho^2\, k^2\, V_c^2\, \frac{\partial^2 u}{\partial x^2} \\ \frac{\partial^2 u}{\partial \tau^2}(x,t) &= \frac{\partial}{\partial \tau}\left(\rho\, k\, \frac{\partial u}{\partial t}\right) = \rho^2\, k^2 \frac{\partial^2 u}{\partial t^2}(x,t) \,. \end{aligned} \tag{4.61}$$

Introducing the relative fluid displacement $w = n(U - u)$, Eqn. 4.53a and 4.53b can be written as a new system of differential equations in terms of ξ and τ (Simon *et al.*, 1984),

$$\frac{\partial^2 u}{\partial \xi^2} + \alpha\varkappa \frac{\partial^2 w}{\partial \xi^2} = \frac{\partial^2 u}{\partial \tau^2} + \beta \frac{\partial^2 w}{\partial \tau^2} \tag{4.62a}$$

$$\alpha\varkappa \frac{\partial^2 u}{\partial \xi^2} + \varkappa \frac{\partial^2 w}{\partial \xi^2} = \beta \frac{\partial^2 u}{\partial \tau^2} + \gamma \frac{\partial^2 w}{\partial \tau^2} + \frac{\partial w}{\partial \tau}\,. \tag{4.62b}$$

The application of the Laplace transform to Eqn. 4.62a and 4.62b leads from a system of partial differential equations to a system of ordinary differential equations. The Laplace transform of a function (here $u(\xi,\tau)$) is defined as follows:

$$\overline{u}(\xi, s) \equiv \int_0^\infty \mathrm{e}^{-s\tau} u(\xi, \tau) \mathrm{d}\tau\,. \tag{4.63}$$

After the Laplace transformation Eqn. 4.62a and 4.62b results in the following ordinary differential equations (Simon *et al.*, 1984). For a more compact notation, the partial derivatives are replaced by the index notation, e.g. $\frac{\partial u}{\partial \xi} = u_\xi$ and $\frac{\partial^2 u}{\partial \xi^2} = u_{\xi\xi}$.

$$\overline{u}_{\xi\xi} + \alpha\,\varkappa\,\overline{w}_{\xi\xi} = s^2\left(\overline{u} + \beta\,\overline{w}\right) \tag{4.64a}$$

$$\alpha\,\varkappa\,\overline{u}_{\xi\xi} + \varkappa\,\overline{w}_{\xi\xi} = s^2\left(\beta\,\overline{u} + \gamma\,\overline{w}\right) + s\,\overline{w} \tag{4.64b}$$

Eqn. 4.64a and 4.64b represent a system of ordinary second order differential equations. These can be transformed into a first order system as follows,

$$\overline{u}_\xi = \mathbf{a}(\xi,\tau) \tag{4.65a}$$

$$\overline{w}_\xi = \mathbf{b}(\xi,\tau) \tag{4.65b}$$

$$\mathbf{a}_\xi = \left(-\frac{s^2 - \alpha\,\beta\,\varkappa\,s^2}{\alpha^2\,\varkappa^2 - 1}\right)\overline{u} + \left(-\frac{\beta\,s^2 - \alpha\,\varkappa\left(\gamma\,s^2 + s\right)}{\alpha^2\,\varkappa^2 - 1}\right)\overline{w} \tag{4.65c}$$

$$\mathbf{b}_\xi = \left(\frac{\beta\,s^2 - \alpha\,\varkappa\,s^2}{\varkappa - \alpha^2\,\varkappa^2}\right)\overline{u} + \left(\frac{s + \gamma\,s^2 - \alpha\,\beta\,\varkappa\,s^2}{\varkappa - \alpha^2\,\varkappa^2}\right)\overline{w}\,. \tag{4.65d}$$

Eqn. 4.65a - 4.65d can be written in matrix-vector form as

$$\mathbf{y}_\xi = \mathbf{A}\,\mathbf{y} \tag{4.66}$$

with

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{s^2 - \alpha\,\beta\,\varkappa\,s^2}{\alpha^2\,\varkappa^2 - 1} & -\frac{\beta\,s^2 - \alpha\,\varkappa\left(\gamma\,s^2 + s\right)}{\alpha^2\,\varkappa^2 - 1} & 0 & 0 \\ \frac{\beta\,s^2 - \alpha\,\varkappa\,s^2}{\varkappa - \alpha^2\,\varkappa^2} & \frac{s + \gamma\,s^2 - \alpha\,\beta\,\varkappa\,s^2}{\varkappa - \alpha^2\,\varkappa^2} & 0 & 0 \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} \overline{\mathbf{u}} \\ \overline{\mathbf{w}} \\ \mathbf{a} \\ \mathbf{b} \end{pmatrix} \quad \text{and} \quad \mathbf{y}_\xi = \begin{pmatrix} \overline{\mathbf{u}}_\xi \\ \overline{\mathbf{w}}_\xi \\ \mathbf{a}_\xi \\ \mathbf{b}_\xi \end{pmatrix}$$

The solution of this system can be found by finding the eigenvalues and eigenvectors of the matrix **A**. Therefore roots of the characteristic polynomial must be determined. The obtained solution depends on the Laplace-parameter s and lives in the complex frequency domain, also known as the s-domain. To obtain a solution in the time domain, the Inverse Laplace Transformation needs to be applied. This is possible for a specific case as was shown by Simon *et al.* (1984).

Dynamic Compatibility The case where a time-domain solution can be obtained was examined by Biot (1956b) and Simon *et al.* (1984) where $\beta = \alpha\,\varkappa$. Biot called this condition "dynamically compatible". To get the solution in the time domain, the Inverse Laplace Transformation is applied. This cannot always be done analytically. Numerical methods can be used (Schanz and Cheng, 2000) in such cases, but in the end, they only give a semi-analytical solution since a numerical error is introduced. However, for this particular case of dynamic compatibility Simon *et al.* (1984) presented a time domain solution. With $\beta = \alpha\,\varkappa$ the eigenvalues of matrix **A** in Eqn.4.4 are

$$\begin{aligned} p_1 &= s \quad , \qquad p_2 = -s \quad , \\ p_3 &= \frac{\sqrt{s\,(\varkappa-\beta^2)\,(-s\,\beta^2+\gamma\,s+1)}}{\varkappa-\beta^2} \quad , \\ p_4 &= -\frac{\sqrt{s\,(\varkappa-\beta^2)\,(-s\,\beta^2+\gamma\,s+1)}}{\varkappa-\beta^2} \end{aligned} \tag{4.67}$$

and the corresponding eigenvectors are

$$\mathbf{v}_1 = \begin{pmatrix} \frac{1}{s} \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad , \quad \mathbf{v}_2 = \begin{pmatrix} \frac{-1}{s} \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad , \quad \mathbf{v}_3 = \begin{pmatrix} \sigma_2 \\ \sigma_3 \\ \sigma_1 \\ 1 \end{pmatrix} \quad \text{and} \quad \mathbf{v}_4 = \begin{pmatrix} -\sigma_2 \\ -\sigma_3 \\ \sigma_1 \\ 1 \end{pmatrix}$$

with

$$\begin{aligned} \sigma_1 &= \frac{(\varkappa-\beta^2)\,(\beta-\beta\,s+\beta\,\gamma\,s)}{(\beta^2-1)\,(\gamma\,s-\varkappa\,s+1)} \\ \sigma_2 &= \frac{\sigma_4\,(\beta\,\varkappa+\beta^3\,s-\beta^3-\beta\,\varkappa\,s-\beta^3\,\gamma\,s+\beta\,\gamma\,\varkappa\,s)}{(\beta^2-1)\,(-\beta^2\,s^2+\gamma\,s^2+s)\,(\gamma\,s-\varkappa\,s+1)} \\ \sigma_3 &= \frac{\sigma_4}{-\beta^2\,s^2+\gamma\,s^2+s} \\ \sigma_4 &= \sqrt{s\,(\varkappa-\beta^2)\,(-s\,\beta^2+\gamma\,s+1)}\,. \end{aligned} \tag{4.68}$$

The dynamically compatible solution of Eqn. 4.66 in the s-domain can now be derived using the eigenvalues p_1, p_2, p_3 and p_4 and the corresponding eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ and $\mathbf{v}_4$ as

$$\mathbf{y}(\xi,\tau) = \mathrm{K}_1\,\mathbf{v}_1\,e^{p_1\,\xi} + \mathrm{K}_2\,\mathbf{v}_2\,e^{p_2\,\xi} + \mathrm{K}_3\,\mathbf{v}_3\,e^{p_3\,\xi} + \mathrm{K}_4\,\mathbf{v}_4\,e^{p_4\,\xi}\,. \tag{4.69}$$

The constants $\mathrm{K}_1, \mathrm{K}_2, \mathrm{K}_3$ and K_4 are determined by the boundary conditions. For a detailed derivation of the time domain solution see Simon *et al.* (1984).

Incompressible grains In geotechnical applications, soil grains are often considered to be incompressible, which corresponds to the condition $\alpha = 1$. Therefore, we will apply this condition instead of the dynamical compatibility to derive an analytical solution. The matrix $\mathbf{A}$ in Eqn. 4.66 simplifies under the assumption $\alpha = 1$ to:

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{s^2-\sigma_1}{\varkappa^2-1} & \frac{\varkappa\left(\gamma\, s^2+s\right)-\beta\, s^2}{\varkappa^2-1} & 0 & 0 \\ \frac{\beta\, s^2-\varkappa\, s^2}{\varkappa-\varkappa^2} & \frac{s+\gamma\, s^2-\sigma_1}{\varkappa-\varkappa^2} & 0 & 0 \end{pmatrix}, \tag{4.70}$$

where $\sigma_1 = \beta\,\varkappa\, s^2$. To obtain the solution as in Eqn. 4.69 the eigenvalues and eigenvectors of $\mathbf{A}$ are needed. These are presented in detail in Appx. B. This new solution is defined in the s-domain and needs to be inverted, if possible, by the Inverse Laplace Transformation to the time domain. The inversion of this solution is a task for the future and will not be discussed further here.

Chapter 5

Material Point Method

This chapter gives an overview of the developments of the material point method (MPM). The motivation for using the MPM in simulating large deformations is explained. Governing equations are discretized and solved using various MPM approaches. The disadvantages of the method are discussed, and different approaches to their solution are presented. Subsequently, MPM is described in the context of scattered data approximation, and several approaches are identified as specific cases of the so-called moving least squares or MLS-MPM. A new approach to improve the method's robustness and accuracy is presented and convergence benchmarks are performed for small and large deformations.

5.1 Standard MPM

The Standard MPM is often referred to as the method published in Sulsky, Chen, *et al.* (1994). For a simple dynamic MPM formulation, we consider Eqn. 4.14 for one material,

$$\rho \frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t} = \mathbf{div}(\boldsymbol{\sigma}) + \rho \mathbf{b}\,. \tag{5.1}$$

In Eqn. 5.1, $\mathbf{v}$ is the velocity vector, $\boldsymbol{\sigma}$ is the Cauchy stress tensor, and $\mathbf{b}$ is the body force vector, e.g., gravity. The density change due to element deformation is calculated by using the determinant of the deformation gradient $J = \det(\mathbf{F})$. Therefore, the last calculated configuration, here denoted by $\mathbf{x}^t$, is taken as reference, i.e. $\mathbf{F} = \partial \mathbf{x}^{t+\Delta t} / \partial \mathbf{x}^t$, and thus an updated Lagrangian formulation (Belytschko, Liu, *et al.*, 2013) is used instead of the total. This corresponds to the definitions introduced in Ch. 4 by setting $\mathbf{X} = \mathbf{x}^t$ and $\mathbf{x} = \mathbf{x}^{t+\Delta t}$. In the derivation of the Standard MPM in Sulsky, Chen, *et al.* (1994), the linearized strain tensor was used for a simpler representation of the calculation of strain. However, this is suitable only if small deformations occur during the incremental step. For spatial discretization Eqn. 5.1 is considered in the weak form. A detailed derivation of several MPM variants can be found in de Vaucorbeil *et al.* (2020). We will give a brief overview here. Therefore, we define the acceleration $\mathbf{a} = (a_x, a_y, a_z)^T := \mathrm{D}\mathbf{v}/\mathrm{D}t$. We now multiply Eqn. 5.1 by the vector-valued test function $\delta \boldsymbol{u}$ and integrate over the domain

Ω. Further we apply the divergence theorem and integration by parts to shift the spatial derivatives to the test function and obtain the weak form of Eqn. 5.1 as

$$\int_{\Omega} [\rho\, \mathbf{a} \cdot \delta \boldsymbol{u} + \boldsymbol{\sigma} : \nabla \delta \boldsymbol{u}] \,\mathrm{d}\Omega = \int_{\Omega} \rho\, \mathbf{b} \cdot \delta \boldsymbol{u} \,\mathrm{d}\Omega + \int_{\partial\Omega} \mathbf{t} \cdot \delta \boldsymbol{u} \,\mathrm{d}S\,. \tag{5.2}$$

The traction $\mathbf{t}$ along the boundary $\partial\Omega$ is defined as in Ch. 4. In Standard MPM, Eqn. 5.2 is solved on a grid in the same manner as in the finite element method (FEM). The difference between FEM and MPM is that initially, all the values connected to the considered material are defined on a set of material points (MP) and not on nodes. Therefore, a method is needed to calculate the nodal values from the set of MP. In the following, we will see that the inverse calculation of values at MP from nodal values is also required. This is the main idea behind MPM to overcome element distortion in case of large deformation. The choice of a proper method to recover values from a set of data points is one of the core topics of this thesis.

In the Standard MPM the one-dimensional nodal shape functions are chosen as

$$N_i(x) = \begin{cases} 1 - |x - x_i| \,/ h & \text{if } |x - x_i| \le h \\ 0 & \text{otherwise} \end{cases} \,. \tag{5.3}$$

Following the ideas of FEM, we choose the test function to be equal to $\delta \boldsymbol{u} = \sum_i^{n_n} \delta \boldsymbol{u}_i N_i$, where the index i indicates the nodal values at the i-th node and n_n the total number of nodes. The length of the considered element is denoted by h. The three-dimensional linear shape function is obtained by multiplication, $N_i(x,y,z) = N_i(x)N_i(y)N_i(z)$. As the nabla operator in Eqn. 5.2 implies, it is necessary to compute the spatial derivatives of the shape function as

$$N_{i,x}(x) = \frac{\partial N_i(x)}{\partial x} = \begin{cases} -\operatorname{sign}(x - x_i) \,/ h & \text{if } |x - x_i| \le h \\ 0 & \text{otherwise} \end{cases} \,. \tag{5.4}$$

The three-dimensional case for the derivatives is again obtained by multiplication, e.g. $N_{i,x}(x,y,z) = N_{i,x}(x)N_i(y)N_i(z)$ etc. The next step is to approximate the functions in Eqn. 5.2 by nodal values and shape functions. For any function $f(\mathbf{x})$ with $\mathbf{x} = (x,y,z)$ the approximation is as follows:

$$f(\mathbf{x}) \approx \sum_{i=1}^{n_n} N_i(\mathbf{x}) f_i \implies f_p = f(\mathbf{x_p}) \approx \sum_{i=1}^{n_n} N_i(\mathbf{x_p}) f_i \ . \tag{5.5}$$

In particular, this approximation is also used to approximate the function f at the MP location $\mathbf{x}_p$, which is denoted by f_p. In Eqn. 5.5, f_i denotes the nodal values of the function f. As previously mentioned, initially, the variables are defined at the MP. Therefore, we need to approximate the nodal values f_i by the MP values f_p first. Therefore, the linear nodal shape functions in the Standard MPM are utilized again,

$$f_i \approx \sum_{p=1}^{n_p} N_i\left(\mathbf{x}_p\right) f_p\,. \tag{5.6}$$

We denote the total number of MP in Eqn. 5.6 by n_p. The approximation of integrals in the Standard MPM is calculated as a weighted sum of MP values,

$$\int_\Omega f(\mathbf{x})\mathrm{d}\Omega \approx \sum_{i=1}^{n_p} V_p\, f_p\,. \tag{5.7}$$

The weights V_p in Eqn. 5.7 correspond to the volumes of MP. In the Standard MPM, V_p is initialized by a portion of the element volume. The density field is defined by

$$\rho(\mathbf{x},t) = \sum_{p=1}^{n_p} m_p \delta(\mathbf{x} - \mathbf{x}_p)\,, \tag{5.8}$$

where, δ is the Dirac delta distribution and m_p the MP mass. Using Eqn. 5.6 - 5.8 and the arbitrariness of the $\delta \boldsymbol{u}_i$'s we can rewrite Eqn. 5.2 in the form

$$\textstyle\sum_{j=1}^{n_n} m_{ij}\hat{\mathbf{a}}_j = (\mathbf{f}_{ext})_i - (\mathbf{f}_{int})_i = \mathbf{f}_i \qquad \text{for } i = 1,2,...,n_n$$

with the nodal forces and masses

$$\begin{aligned}
(\mathbf{f}_{ext})_i &= \textstyle\sum_{p=1}^{n_p} m_p N_i(\mathbf{x}_p)\mathbf{b}(\mathbf{x}_p) + \int_{\partial\Omega} \mathbf{t} N_i(\mathbf{x}_p)\mathrm{d}S\,,\\
(\mathbf{f}_{int})_i &= \textstyle\sum_{p=1}^{n_p} V_p\, \boldsymbol{\sigma_p}\ \mathbf{grad}(N_i)(\mathbf{x}_p)\,,\\
m_{ij} &= \textstyle\sum_{p=1}^{n_p} m_p N_i(\mathbf{x}_p) N_j(\mathbf{x}_p)\,.
\end{aligned} \tag{5.9}$$

Eqn. 5.9 is a system of equations for the nodal accelerations $\hat{\mathbf{a}}_j = (a_x^j, a_y^j, a_z^j)^T$ with the $3 \times 3-$matrix for the MP stress $\boldsymbol{\sigma_p}$. Often the nodal mass is lumped. This approximation approach leads to additional numerical errors but is more efficient for numerical implementation.

$$m_i = \sum_{j=1}^{n_n} m_{ij} = \sum_{j=1}^{n_n}\sum_{p=1}^{n_p} m_p N_i(\mathbf{x}_p) N_j(\mathbf{x}_p) = \sum_{p=1}^{n_p} m_p N_i(\mathbf{x}_p) \tag{5.10}$$

The lumping procedure is shown in Eqn. 5.10. Furthermore, using Eqn. 5.6 and 5.10 and the property of the nodal shape functions to be a partition of unity $\left(\sum_j N_j(\mathrm{x}) = 1\ \forall \mathrm{x}\right)$ it can be shown that the total mass at the nodes is equal to the total mass at the MP.

$$\sum_{i=1}^{n_n} m_i = \sum_{i=1}^{n_n}\left(\sum_{p=1}^{n_p} m_p N_i(\mathbf{x}_p)\right) = \sum_{p=1}^{n_p} m_p \left(\sum_{i=1}^{n_n} N_i(\mathbf{x}_p)\right) = \sum_{p=1}^{n_p} m_p \tag{5.11}$$

It is beneficial to rewrite Eqn. 5.6 in a different and more general way. Therefore, M.Steffen *et al.* (2008) considered the following approximation of nodal values,

$$f_i(\mathbf{x}) = \sum_{p=1}^{n_p} f(\mathbf{x}_p) \cdot \underbrace{\left(\frac{\int_\Omega N_i(\mathbf{x})\chi(\mathbf{x} - \mathbf{x}_p)\,\mathrm{d}\Omega}{\int_\Omega \chi(\mathbf{x} - \mathbf{x}_p)\,\mathrm{d}\Omega}\right)}_{:=\phi_{ip}(\mathbf{x})}\,. \tag{5.12}$$

The quotient of the integrals in Eqn. 5.12 defines the weighting function $\phi_{ip}(\mathbf{x})$. The function $\chi(\mathbf{x})$ represents the MP shape function. The choice of different shape functions for the MP leads to different MPM variants, as will be shown later. The Standard MPM can be recovered by setting $\chi(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_p)$. Inserting $\chi(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_p)$ into Eqn. 5.12 leads to $f_i = \sum_{p=1}^{n_p} f_p N_i(\mathbf{x}_p)$, which is exactly Eqn. 5.6. The calculation steps of the Standard MPM can be described as a cyclic sequence of four steps, as shown in Fig. 5.1. The initialization of nodal values in **MP → Nodes** is always the beginning of the cycle. If, for a simpler representation, the external forces $(\mathbf{f}_{ext})_i$ used in Eqn. 5.2 are set to zero, the calculation cycle shown in Fig. 5.1 can be simplified using the equations in Tab. 5.1. For more details, see de Vaucorbeil *et al.* (2020).

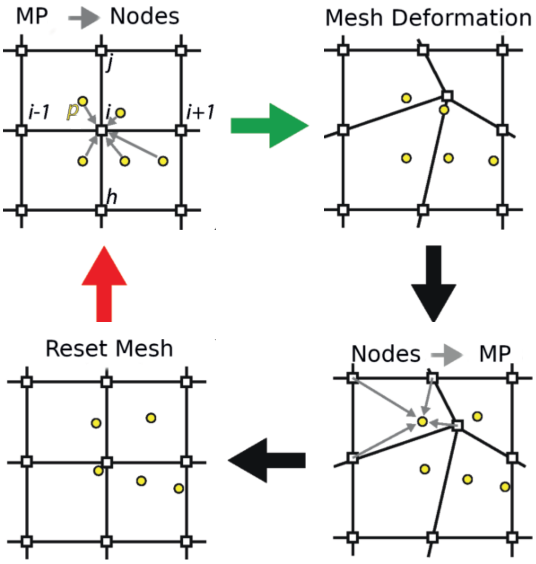


Figure 5.1: Calculation steps of MPM, beginning with the initialization at nodes in step **MP → Nodes** and ending with the mesh reset in step **ResetMesh**.

MP → Nodes	**Mesh Deformation**	**Nodes → MP**
$m_i^t = \sum_p N_i\left(\mathbf{x}_p^t\right) m_p$	$(m\mathbf{v})_i^{t+\Delta t} = (m\mathbf{v})_i^t + \mathbf{f}_i^t \Delta t$	$\mathbf{v}_p^{t+\Delta t} = \mathbf{v}_p^t + \Delta t \sum_i N_i\left(\mathbf{x}_p^t\right) \mathbf{f}_i^t / m_i^t$
$(m\mathbf{v})_i^t = \sum_p N_i\left(\mathbf{x}_p^t\right) (m\mathbf{v})_p^t$	$\mathbf{x}_i^{t+\Delta t} = \mathbf{x}_i^t + \Delta t \mathbf{v}_i^{t+\Delta t}$	$\mathbf{x}_p^{t+\Delta t} = \mathbf{x}_p^t + \Delta t \sum_i N_i\left(\mathbf{x}_p^t\right) (m\mathbf{v})_i^{t+\Delta t} / m_i^t$
$(\mathbf{f}_{int})_i^t = \sum_p V_p^t \boldsymbol{\sigma}_p^t \, \mathbf{grad}(N_i)\left(\mathbf{x}_p^t\right)$		$\mathbf{L}_p^{t+\Delta t} = \sum_i \mathbf{grad}(N_i)\left(\mathbf{x}_p^t\right) \cdot (\mathbf{v}_i^{t+\Delta t})^T$
$(\mathbf{f}_{ext})_i^t = \sum_{p=1}^{n_p} m_p^t N_i(\mathbf{x}_p^t)\mathbf{b}(\mathbf{x}_p^t)$		$\mathbf{F}_p^{t+\Delta t} = \left(\mathbf{I} + \mathbf{L}_p^{t+\Delta t} \Delta t\right) \mathbf{F}_p^t$
		$V_p^{t+\Delta t} = \det\left(\mathbf{F}_p^{t+\Delta t}\right) V_p^0$
		$\boldsymbol{\sigma}_p^{t+\Delta t} = \boldsymbol{\sigma}_p^t + \Delta \boldsymbol{\sigma}_p$

Table 5.1: Simplified equations for the Standard MPM steps in Fig. 5.1 using explicit time integration. The velocity gradient at the MP is denoted as $\mathbf{L}_p^{t+\Delta t}$ and is calculated from the dyadic products of the gradients of shape functions and the velocity vectors.

5.2 Problems of the Standard MPM

The standard MPM, as described in Tab. 5.1, leads to several numerical problems, which can lead to divergence and unphysical behaviour of the solutions. These problems led to the development of new MPM variants (Bardenhagen and Kober, 2004; Sadeghirad, R. M. Brannon, *et al.*, 2011; Sadeghirad, R. Brannon, *et al.*, 2013) and many more. One of the issues in the Standard MPM are the linear shape functions that are applied for the nodes (see Eqn. 5.3), whose derivatives have a discontinuity at the node (see Eqn. 5.4). In the following, we will discuss the known issues of the standard MPM in more detail.

Grid Crossing As shown in Tab. 5.1, the gradients of the shape function in the steps **MP → Nodes** and **Nodes → MP** are used to calculate $(\mathbf{f}_{int})_i^t$ and $\mathbf{L}_p^{t+\Delta t}$. Eqn. 5.4 shows that the derivative of the shape function in the elements adjacent to the i-th node is a piecewise constant function with opposing signs. Thus, the contribution of MP to the internal force changes sign when MP transition from one element to the neighboring one, which leads to an unphysical jump of the internal forces (Bardenhagen and Kober, 2004). An obvious solution to the problem is to use shape functions whose derivatives at the nodes are continuous, such as B-Splines (M.Steffen *et al.*, 2008; Steffen *et al.*, 2008) or Bernstein polynomials (de Vaucorbeil *et al.*, 2020). Another solution is to choose functions with a larger compact support instead of the delta distribution in Eqn. 5.12 for $\chi(\mathbf{x})$ (Bardenhagen and Kober, 2004; Sadeghirad, R. M. Brannon, *et al.*, 2011; Sadeghirad, R. Brannon, *et al.*, 2013). To shed more light on the grid crossing problem, we consider the one-dimensional case, where $N_{\mathrm{p}}^{\mathrm{e}}$ represents the number of MP in the $e-$th element, $\boldsymbol{\sigma}_p$ the stress and V_p the volume of the $p-$th material point. The internal force at node i is calculated as follows

$$(\mathbf{f}_{int})_i^t = \frac{1}{h}\sum_{\mathrm{p}=1}^{N_{\mathrm{p}}^{\mathrm{e}}} \boldsymbol{\sigma}_p V_p - \frac{1}{h}\sum_{\mathrm{p}=1}^{N_{\mathrm{p}}^{\mathrm{e}+1}} \boldsymbol{\sigma}_p V_p \,. \tag{5.13}$$

Assuming that the volumes and stresses of the MP are almost equal for all MP ($\boldsymbol{\sigma}_p \approx \boldsymbol{\sigma}\ \forall \mathrm{p}$ and $V_p \approx V\ \forall \mathrm{p}$), Eq. 5.13 can be rewritten as

$$(\mathbf{f}_{int})_i^t = \frac{\boldsymbol{\sigma} V}{h}(N_{\mathrm{p}}^{\mathrm{e}} - N_{\mathrm{p}}^{\mathrm{e}+1})\,. \tag{5.14}$$

Thus the internal force in the node i depends on the number of MP in the elements e and e + 1. If one MP in element e, which is close to the node i, crosses the grid and enters element e + 1, a sudden change in the internal force of $2\boldsymbol{\sigma}V/h$ occurs. This is a well-known numerical issue of the standard MPM called grid crossing. There are many different approaches to circumvent or mitigate grid crossing issues, which will be discussed in detail in Sec. 5.3.

Loss of contact Another problem is that the distance on which MP can interact according to the field equations depends on the size of the compact support of the weighting function and the element size. In standard MPM, the compact support of the weighting function in the one-dimensional case is equal to $2h$. If, for example, the continuum described by the MP would be torn apart due to an applied tension and a gap of one or more empty elements arise between them, there will be no information transfer between these MP anymore. This loss of information transfer is not a physical phenomenon but a numerical artefact of the MPM since it depends directly on the degree of discretization or the element size. A loss of the internal cohesion of the continuum should only be described within the material models. A partly solution to this problem is, as mentioned above, to choose the shape functions $\chi(\mathbf{x})$ of the MP in such a way that the weighting functions $\phi_{ip}(\mathbf{x})$ get a larger compact support (Bardenhagen and Kober, 2004). This would increase the number of possible empty elements between MP until the contact is lost but not solve the problem entirely.

Problem of small masses The interpolation of MP masses to nodes leads to another numerical issue in the Standard MPM. In the one-dimensional case if one MP is located near the left node of an element, its contribution to the nodal mass of the right node is $m_R^t = N_R(\mathbf{x}_p^t)m_p^t$. Here the index R denotes the right node. Since in the standard MPM the weighting function $N_R(\mathbf{x}_p^t)$ is defined according to Eqn. 5.3, its value at the left node is by definition zero. Thus the contribution to the nodal mass of the right node is very small. In the step **MeshDeformation** in Tab. 5.1, the nodal momentum must be divided by the nodal mass to calculate the nodal velocity, resulting in very large values. In the step **Nodes → MP**, these nodal velocities are used to calculate the velocity gradient and thus also the deformation gradient, resulting in an incorrect deformation of MP. Sulsky, Zhou, *et al.* (1995) introduced the Modified Update Stress Load (MUSL) method to solve this problem, where an additional interpolation step is performed between MP and nodes so that nodal velocities are calculated from the updated MP velocities as

$$\mathbf{v}_i^{t+\Delta t} = \frac{\sum_p N_i(\mathbf{x}_p^t)(m\mathbf{v})_p^{t+\Delta t}}{\sum_p N_i(\mathbf{x}_p^t)m_p} \,. \tag{5.15}$$

The additional interpolation step causes the weighting functions to appear in the numerator and denominator in Eqn. 5.15. As a result, nodal velocities no longer take extreme values. Larger compact support of the weighting functions would also reduce the problem of small nodal masses at least at the neighbouring node but would not solve it completely.

Element Integration The element integration in the standard MPM uses the MP as integration points. The accuracy of such integration depends highly on the location of the MP. In contrast, the Gauss integration is based on a fixed amount of integration points in an optimal location to achieve high accuracy. Beuth (2012) proposed to use a combination of both integration schemes to improve the accuracy of element integration. The Gauss integration is not suitable for partially filled elements as the whole element volume is used for the integration. In contrast, the sum of the material point volumes can be smaller. Therefore, it is only recommended to apply Gauss integration if the total particle volume inside one element is at least 90 per cent of its volume (Al-Kafaji, 2013).

5.3 Improved variants of MPM

5.3.1 GIMP

The Generalized Interpolation Material Point Method (GIMP) was first presented in Bardenhagen and Kober (2004) and discussed using two special GIMP variants, the unchanged GIMP (uGIMP) and contiguous particle GIMP (cpGIMP). The basic idea of the GIMP method is to choose the MP shape function $\chi(\mathbf{x})$ in Eqn. 5.12, different from the delta distribution $\delta(\mathbf{x}-\mathbf{x_p})$. For both GIMP variants, the shape functions were chosen as the characteristic function with compact supports corresponding to the MP volume. In the following, we will distinguish between the MP domain $\Omega_p \subset \Omega$ and its volume V_p. The domain Ω_p defines the compact support of the MP shape function,

$$\chi\left(\mathbf{x}-\mathbf{x}_p\right)=\begin{cases}1 & \text{if} \quad \mathbf{x}\in\Omega_p \\ 0 & \text{otherwise}\end{cases}. \tag{5.16}$$

Integration of Eqn. 5.16 over Ω gives $\int_\Omega \chi\left(\mathbf{x}-\mathbf{x}_p\right)\mathrm{d}\Omega = V_p$. Therefore, the weighting function ϕ_{ip} from Eqn. 5.12 simplifies to

$$\phi_{ip}\left(\mathbf{x}\right)=\frac{\int_\Omega N_i(\mathbf{x})\chi\left(\mathbf{x}-\mathbf{x}_p\right)\mathrm{d}\Omega}{\int_\Omega \chi\left(\mathbf{x}-\mathbf{x}_p\right)\mathrm{d}\Omega}=\frac{1}{V_p}\int_{\Omega_p} N_i(\mathbf{x})\mathrm{d}\Omega\,. \tag{5.17}$$

The choice of incorporating the changes of Ω_p in the integral in Eqn. 5.17 is the key for all GIMP-based methods. The goal of choosing a proper method is to overcome the problems mentioned in Sec. 5.2 as much as possible.

uGIMP The uGIMP is the simplest GIMP variant. Here Ω_p is assumed to be constant or undeformable in all directions with the initial MP volume V_p^0. The weighting function in Eqn. 5.17 can be solved analytically for the uGIMP as one-dimensional integrals for each spatial direction. If ϕ_{ip}^x and ϕ_{ip}^y are the one-dimensional weighting functions for the x- and y-direction respectively, then $\phi_{ip}(x,y) = \phi_{ip}^x(x)\cdot\phi_{ip}^y(y)$ is the two-dimensional weighting function. Fig. 5.2 shows the p-th MP (yellow dot with horizontal bar) moving through a one-dimensional mesh from left to right.

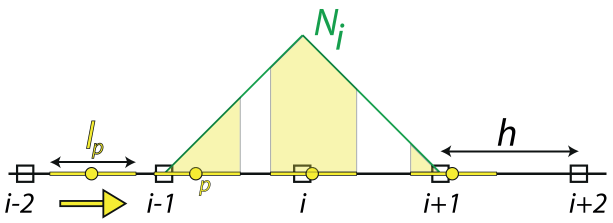


Figure 5.2: One-dimensional convolution integral between χ and N_i. The compact support of χ has the length l_p. The yellow arrow indicated the movement of particle p. The yellow colored areas represent the contribution to the convolution integral, which is non-zero when the particle moves from left to right.

The domain Ω_p corresponds in one dimension to the length l_p (length of the bar). The distance between the nodes is h. The green line represents the linear shape function N_i of the i-th node. The yellow areas under N_i indicate the integral of Eqn. 5.17 for different positions of the p-th MP. A coordinate system is chosen so that $x = 0$ is located at the i-th node, and x is associated with the centre of l_p. Thus, Eqn. 5.17 can be integrated piecewise (M.Steffen *et al.*, 2008),

$$\phi_{ip}^x(x) = \begin{cases} 1 - \left(4x^2 + l_p^2\right) / \left(4hl_p\right) & |x| < 0.5l_p \\ 1 - |x|/h & 0.5l_p \leq |x| \leq h_x - 0.5l_p \\ \left(h + l_p/2 - |x|\right)^2 / \left(2hl_p\right) & h - 0.5l_p \leq |x| < h + 0.5l_p \\ 0 & \text{otherwise} \end{cases} . \tag{5.18}$$

Eqn. 5.18 shows the weighting function ϕ_{ip}^x for uGIMP. Compared to the Standard MPM, the compact support of $\phi_{ip}^x(x)$ is enlarged by $l_p/2$ in each direction, which allows MP to interact for greater distance. The derivative of $\phi_{ip}^x(x)$ can be derived piecewise as

$$\frac{\partial \phi_{ip}^x}{\partial x}(x) = \begin{cases} -8x/\left(4hl_p\right) & |x| < 0.5l_p \\ -\left(1/h\right)\operatorname{sign}(x) & 0.5l_p \leq |x| \leq h - 0.5l_p \\ -\operatorname{sign}(x)\left(h + l_p/2 - |x|\right)/\left(hl_p\right) & h - 0.5l_p \leq |x| < h + 0.5l_p \\ 0 & \text{otherwise} \end{cases} . \tag{5.19}$$

The derivative of $\phi_{ip}(x)$ is shown in Eqn. 5.19. It can be seen that the derivative of the weighting function, unlike the standard MPM, has no discontinuities, which mitigates the grid crossing problem.

cpGIMP With the uGIMP method, the disadvantages of the Standard MPM are reduced by increasing the compact support of ϕ_{ip} and making the derivatives of ϕ_{ip} a continuous function. However, it is a problem that l_p is a constant length in Eqn. 5.18, and this can cause overlaps or gaps between the MP. The integration domain Ω_p in Eqn. 5.17 is not changing according to the deformation gradient. In cpGIMP this is improved, and changes of l_p and corresponding Ω_p are considered. The cpGIMP follows a simple approach, where in the two-dimensional case the MP domain Ω_p^t at time t is defined by the Cartesian product $l_p^x(t) \times l_p^y(t)$. In case of deformation, only the diagonal elements of the deformation gradient $\mathbf{F}$ are used to calculate the deformed domain $\Omega_p^{t+\Delta t} = l_p^x(t)F_{xx}^{t+\Delta t} \times l_p^y(t)F_{yy}^{t+\Delta t}$. Correspondingly, the domain Ω_p^t changes along the x- and y-coordinate. This approach is not suitable for problems with large shear deformations because the non-diagonal elements of $\mathbf{F}$ are not considered.

5.3.2 CPDI/CPDI2

The Convected Particle Domain Interpolation (CPDI) method (Sadeghirad, R. M. Brannon, *et al.*, 2011) was the first MPM variant to consider the off-diagonal elements of the deformation gradient in Eqn. 5.17. At the same time, it is important that the integral in Eqn. 5.17 can be calculated as simple as possible. The CPDI method considers parallelepipeds for the MP domains Ω_p. An improved CPDI variant, which will be referred to as CPDI2 hereafter, was introduced in Sadeghirad, R. Brannon, *et al.* (2013). It considers arbitrary arrangement of corner points of the initially cuboid shaped Ω_p. Fig. 5.3 shows a two-dimensional example of an initially rectangular domain Ω_p^0 representing the compact support of $\chi(\mathbf{x})$ for the CPDI and CPDI2 method.

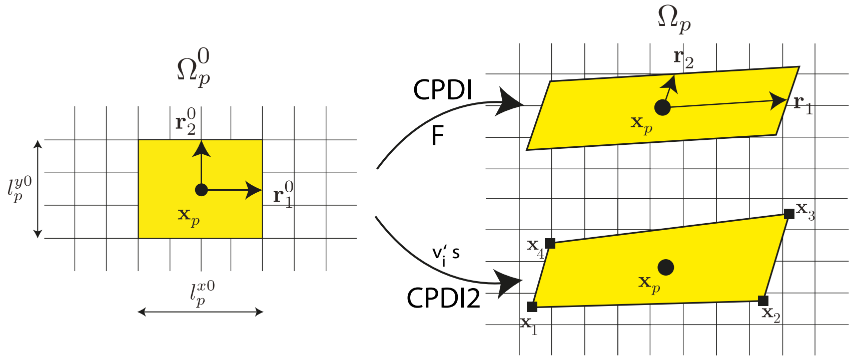


Figure 5.3: Deformation step in the CPDI and CPDI2 method: While in the CPDI method, the two vectors $\mathbf{r}_1^0$ and $\mathbf{r}_2^0$ in the initial state Ω_p^0 are deformed according to the deformation gradient $\mathbf{F}$ into the state Ω_p, in CPDI2 the four corner points of the initial state Ω_p^0 are decisive. These are transformed into the deformed configuration based on the adjacent nodal velocities.

For CPDI the area is characterized by the two vectors $\mathbf{r_1}$ and $\mathbf{r_2}$. The deformation of the domain is described by the deformation gradient $\mathbf{F}$ according to

$$\begin{aligned} \mathbf{r}_1 &= \mathbf{F}\,\mathbf{r}_1^0\,, \\ \mathbf{r}_2 &= \mathbf{F}\,\mathbf{r}_2^0\,. \end{aligned} \tag{5.20}$$

The deformation described in Eqn. 5.20 creates a new parallelogram, which is described by the rotated and stretched vectors $\mathbf{r}_1$ and $\mathbf{r}_2$. The deformation in CPDI2 is calculated using the vertices $\mathbf{x}_c$ of the domain

$$\mathbf{x}_c^{t+\Delta t} = \mathbf{x}_c^t + \Delta t \sum_i N_i\left(\mathbf{x}_c\right)\mathbf{v}_i^{t+\Delta t} \qquad \text{for} \quad c \in \{1\,,2\,,3\,,4\}\,. \tag{5.21}$$

In Eqn. 5.21 the CPDI2 calculation of the new vertex coordinates is shown using the nodal velocities $\mathbf{v}_i$'s. Unlike CPDI, CPDI2 also allows deformations that do not occur along the vectors $\mathbf{r_1}$ and $\mathbf{r_2}$. It was shown by Sadeghirad, R. Brannon, *et al.* (2013) that this modification significantly improves the accuracy of the method. In addition to the modified compact support, CPDI and CPDI2 use modified shape functions N_i^{mod} at the nodes to make the integration in Eqn. 5.17 easier.

$$N_i(\mathbf{x}) \approx N_i^{\mathrm{mod}}(\mathbf{x}) = \sum_{c=1}^{4} M_c(\mathbf{x}) N_i\left(\mathbf{x}_c\right) \tag{5.22}$$

In Eqn. 5.22, $M_c(\mathbf{x})$ denotes the linear shape functions defined over the area Ω_p, for which $M_d(\mathbf{x}_c) = \delta_{cd}$ applies, where δ_{cd} is the Kronecker delta function. The introduction of the modified shape functions allows to evaluate the integral of Eqn. 5.12 only by integrating $M_c(\mathbf{x})$ over Ω_p.

$$\begin{aligned}\phi_{ip}(\mathbf{x}) &= \frac{1}{V_p}\int_{\Omega_p} N_i^{\mathrm{app}}(\mathbf{x})\mathrm{d}\Omega = \frac{1}{V_p}\int_{\Omega_p}\left[\sum_{c=1}^{4} M_c(\mathbf{x})N_i\left(\mathbf{x}_c\right)\right]\mathrm{d}\Omega \\ &= \frac{1}{V_p}\sum_{c=1}^{4}\left[\int_{\Omega_p} M_c(\mathbf{x})\mathrm{d}\Omega\right]N_i\left(\mathbf{x}_c\right)\end{aligned} \tag{5.23}$$

Eqn. 5.23 shows the calculation of the weighting functions for CPDI and CPDI2. Due to the definition of the modified shape functions in Eqn. 5.22, no convolution integral of the shape functions has to be calculated anymore because it is evaluated only at the vertices of Ω_p, which is a big advantage. CPDI2 also has several advantages over CPDI since CPDI2 can be easily defined not only for parallelepipeds but for different polygons, according to Eqn. 5.21. In V. P. Nguyen, C. T. Nguyen, *et al.* (2017) CPDI2 weighting functions for different dimensions and geometries are presented. Another advantage of CDPI2 is that it does not have gaps or overlaps between the areas Ω_p due to the conformal deformation in Eqn. 5.21.

Comparison of different formulations To demonstrate the described methods we consider an elastic bouncing ball problem in two dimensions. We use a square element discretization with the length of $h = 0.05\,\mathrm{m}$ and one MP per element. The radius of the ball is $r = 0.2\,\mathrm{m}$. The centre of the ball is initially located at $\mathbf{x} = (0.5\,, 0.5)$ and is then subjected to gravity. The constitutive behaviour of the ball is modelled by the Neo-Hookean material model

$$\boldsymbol{\sigma} = \frac{1}{J}\left(\lambda_N \ln(J)\mathbf{I} + \mu(\mathbf{F}\mathbf{F}^T - \mathbf{I})\right)\ . \tag{5.24}$$

In Eqn.5.24 the material parameters λ_N and μ are the Lamé constants and are set to 0 and $50000\,\mathrm{Ns/m^2}$ correspondingly. The density is of the ball is $\rho = 2000\,\mathrm{kg/m^3}$. The Cauchy stress tensor $\boldsymbol{\sigma}$ and deformation gradient in Eqn. 5.24 are represented by $2 \times 2-$matrices in the considered two-dimensional case.

The results of three different MPM variants are shown in Fig. 5.5. It can be seen that different approaches to consider the changes of the MP domains Ω_p affect the vertical stresses and the calculated velocities.

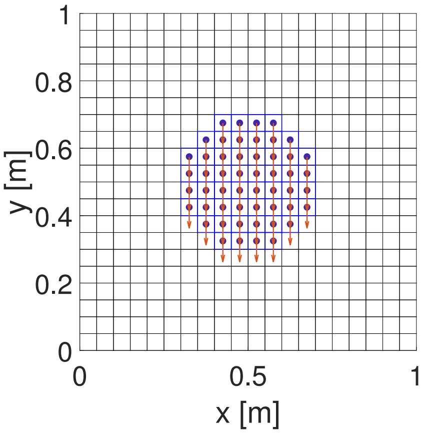


Figure 5.4: Initial state of the ball before the fall. The computational grid consists of square elements with length $h = 0.05\,\text{m}$. The blue squares represent the MP domains Ω_p, which initially coincide with the size of elements. The velocity of the MP is indicated by red arrows.

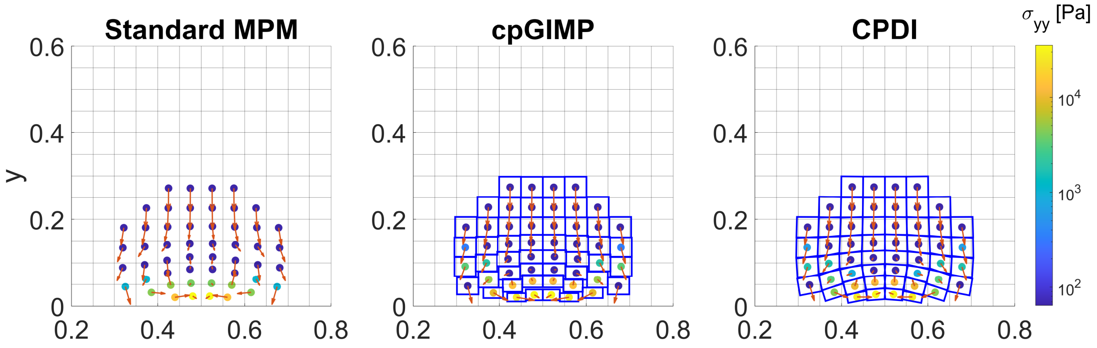


Figure 5.5: Comparison of different MPM variants for the impact of the ball after $t = 0.285\,\text{s}$. The left figure shows the results of Standard MPM, where χ is the Dirac delta distribution and therefore Ω_p coincides with the MP locations. The middle figure shows the cpGIMP results, where Ω_p only deforms in x- and y-directions. The right figure shows the CPDI results, where Ω_p deforms according to Eqn. 5.20.

5.3.3 B-Spline MPM (BSMPM)

M.Steffen *et al.* (2008) and Steffen *et al.* (2008) suggested replacing the nodal linear shape functions with higher-order B-splines, as their derivatives are not discontinuous. Furthermore, they are a partition of unity, which is essential for conservation properties. There are different ways to construct B-splines, and some of them are presented below. First we consider one-dimensional B-Splines for the Knot-Vector $i \in \{0; 1; 2; 3; 4; 5\}$. The recursive formula presented by Boor (1978) can be used to construct B-Splines of different order as

$$\phi_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$\text{and for } k \geq 1 \tag{5.25}$$

$$\phi_{i,k}(\xi) = \frac{\xi-\xi_i}{\xi_{i+k}-\xi_i}\phi_{i,k-1}(\xi) + \frac{\xi_{i+k+1}-\xi}{\xi_{i+k+1}-\xi_{i+1}}\phi_{i+1,k-1}(\xi)\,.$$

In the recursive formula in Eqn. 5.25, the index k indicates the polynomial order of the B-Spline. To calculate higher-order B-splines, the boundary knots are used several times. For example, the knot vector to construct a cubic spline (k=3) looks like this: $\{0;0;0;0;1;2;3;4;5;5;5;5\}$. In the following, the quadratic and cubic B-Splines with their derivatives are shown.

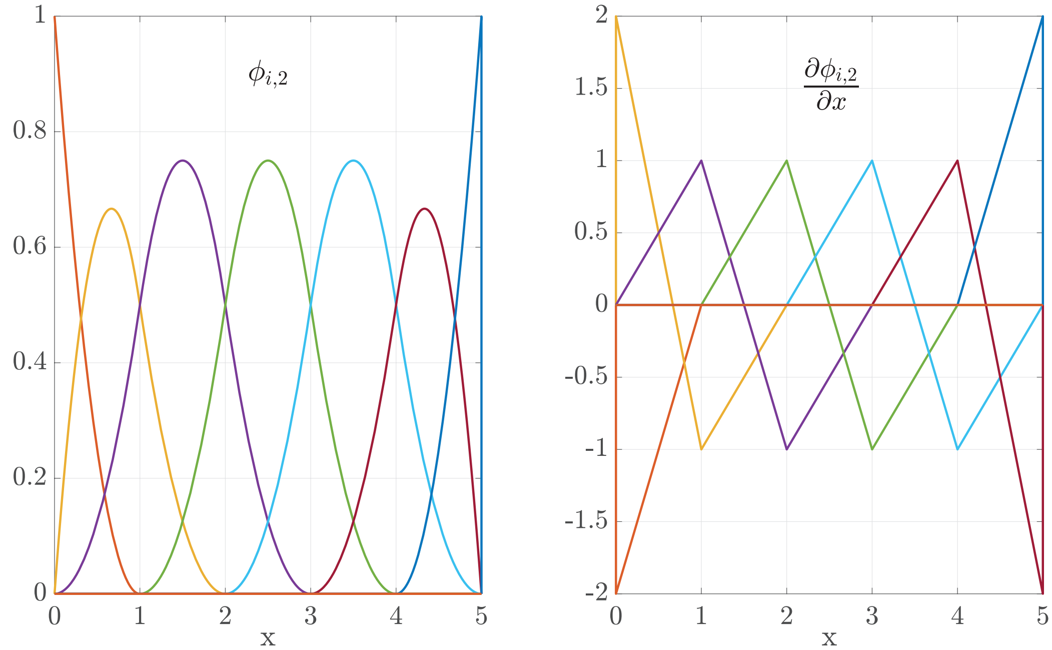


Figure 5.6: Quadratic B-Splines and their derivatives for k=2 on the interval [0,5] with knots at 0, 1, 2, 3, 4 and 5.

As shown in Fig. 5.6 and 5.7 the B-Splines defined by Eqn. 5.25 are not all centred over the nodes $\{0;1;2;3;4;5\}$ and are not equal one at the nodes despite at the boundaries. Nevertheless, they form a partition of unity and sum up to one everywhere. The compact support changes from length one at the boundaries to the maximum of three for the quadratic and four for the cubic case inside the considered interval. The derivatives of the quadratic and cubic B-Splines in Fig. 5.6 and 5.7 are continuous and therefore suitable to mitigate the grid crossing noise. Increasing the polynomial order of the B-Splines, increases the compact support and the smoothness of their derivatives. At the same time,

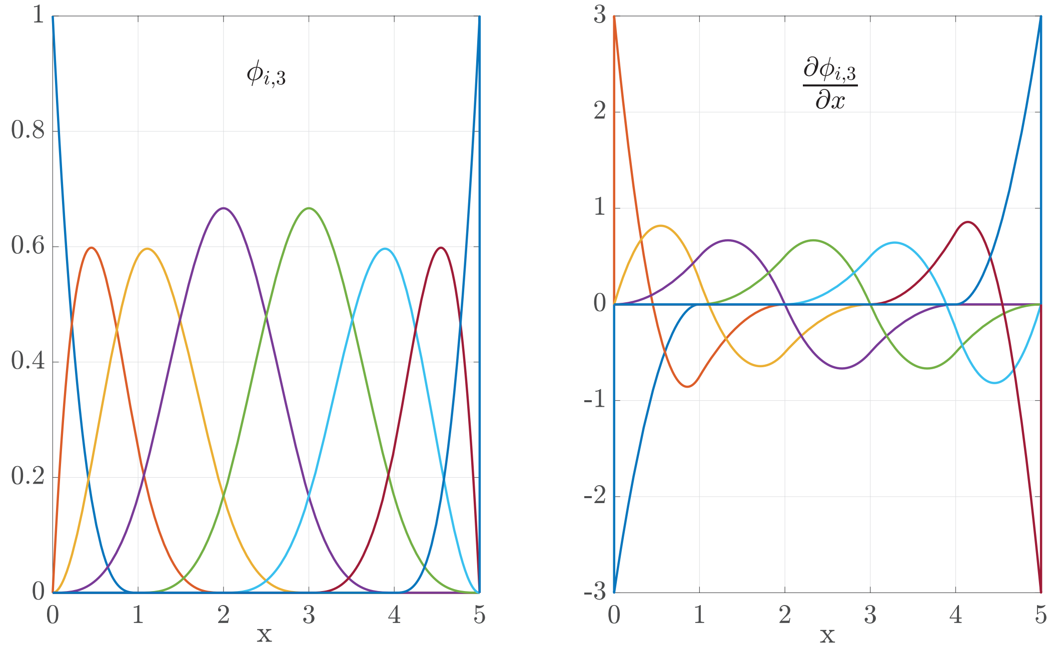


Figure 5.7: Cubic B-Spline and their derivatives for k=3 on the interval [0,5] with knots at 0, 1, 2, 3, 4 and 5.

the number of functions is increased, which leads to a higher computational effort. The B-Splines in Fig. 5.6 and 5.7 sum up to one over the interval [0,5] but are not one at the knots. Furthermore, their maxima are not located at the knots.

Centred B-Splines The B-Spline functions can also be obtained by convolution of rectangular functions similar to the derivation of GIMP. This is helpful to obtain centred B-Splines, which can be defined at grid nodes.
Fig. 5.8 shows the multiple convolution of the rectangular function, which leads to a linear B-Spline after the first, a quadratic B-Spline after the second and a cubic B-Spline after the third convolution. The resulting zero centred B-Splines can be defined at the grid nodes by shifting the centre. The resulting quadratic and cubic B-Splines with their derivatives for the one-dimensional case are shown below.

$$\phi_{i,2}(x) = \begin{cases} \frac{1}{2h^2}\left(x-x_i\right)^2 + \frac{3}{2h}\left(x-x_i\right) + \frac{9}{8} & \text{if } \; x \in \left[x_i - \frac{3}{2}h \,, x_i - \frac{1}{2}h\right] \\ \frac{3}{4} - \frac{1}{h^2}(x-x_i)^2 & \text{if } \; x \in \left[x_i - \frac{1}{2}h \,, x_i + \frac{1}{2}h\right] \\ \frac{1}{2h^2}\left(x-x_i\right)^2 - \frac{3}{2h}\left(x-x_i\right) + \frac{9}{8} & \text{if } \; x \in \left[x_i + \frac{1}{2}h \,, x_i + \frac{3}{2}h\right] \\ 0 & \text{otherwise} \end{cases} \tag{5.26}$$

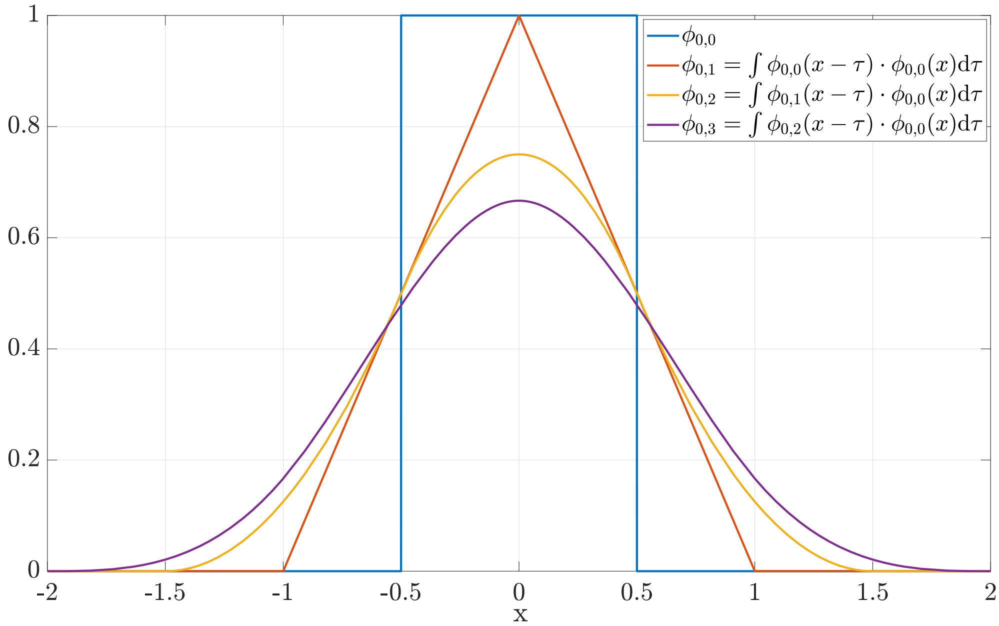


Figure 5.8: Derivation of B-Splines by convolution of rectangular functions (blue): First convolution results in a linear shape function corresponding to a first-order or linear B-Spline. The second convolution results in a quadratic B-Spline. The third convolution results in a cubic B-Spline. The knot index $i = 0$ corresponds to $x = 0$.

$$\frac{\partial \phi_{i,2}(x)}{\partial x} = \begin{cases} \frac{1}{h^2}\left(x - x_i\right) + \frac{3}{2h} & \text{if } \; x \in \left(x_i - \frac{3}{2}h\,, x_i - \frac{1}{2}h\right) \\ -\frac{2}{h^2}\left(x - x_i\right) & \text{if } \; x \in \left(x_i - \frac{1}{2}h\,, x_i + \frac{1}{2}h\right) \\ \frac{1}{h^2}\left(x - x_i\right) - \frac{3}{2h} & \text{if } \; x \in \left(x_i + \frac{1}{2}h\,, x_i + \frac{3}{2}h\right) \\ 0 & \text{otherwise} \end{cases} \tag{5.27}$$

The quadratic B-spline in Eqn. 5.26 and its derivative in Eqn. 5.27 represent the uGIMP weighting functions shown in Eqn. 5.18 and 5.19 for the case $l_p = h$. This is not a coincidence, as the GIMP weighting functions were also derived as the convolution of rectangular and tent functions.

$$\phi_{i,3}(x) = \begin{cases} \frac{(x-x_i)^3}{6h^3} + \frac{1}{h^2}(x-x_i)^2 + \frac{2}{h}(x-x_i) + \frac{4}{3} & \text{if } \; x \in [x_i - 2h\,, x_i - h] \\ -\frac{(x-x_i)^3}{2h^3} - \frac{1}{h^2}(x-x_i)^2 + \frac{2}{3} & \text{if } \; x \in [x_i - h\,, x_i] \\ \frac{(x-x_i)^3}{2h^3} - \frac{1}{h^2}(x-x_i)^2 + \frac{2}{3} & \text{if } \; x \in [x_i\,, x_i + h] \\ -\frac{(x-x_i)^3}{6h^3} + \frac{1}{h^2}(x-x_i)^2 - \frac{2}{h}(x-x_i) + \frac{4}{3} & \text{if } \; x \in [x_i + h\,, x_i + 2h] \\ 0 & \text{otherwise} \end{cases} \tag{5.28}$$

$$\frac{\partial \phi_{i,3}}{\partial x} = \begin{cases} \frac{1}{2h^2}\left(x - x_i\right)^2 + \frac{2}{h^2}\left(x - x_i\right) + \frac{2}{h} & \text{if } \ x \in (x_i - 2h\,, x_i - h) \\ -\frac{3}{2h^3}\left(x - x_i\right)^2 - \frac{2}{h^2}\left(x - x_i\right) & \text{if } \ x \in (x_i - h\,, x_i) \\ \frac{3}{2h^3}\left(x - x_i\right)^2 - \frac{2}{h^2}\left(x - x_i\right) & \text{if } \ x \in (x_i\,, x_i + h) \\ -\frac{1}{2h^2}\left(x - x_i\right)^2 + \frac{2}{h^2}\left(x - x_i\right) - \frac{2}{h} & \text{if } \ x \in (x_i + 2h\,, x_i + h) \\ 0 & \text{otherwise} \end{cases} \tag{5.29}$$

The derived centred B-Splines correspond to inner B-Splines shown in Fig. 5.6 and 5.7, which are different from the B-Splines in the vicinity of the boundaries (de Vaucorbeil *et al.*, 2020). The centred cubic B-Spline and its derivate are presented in Eqn. 5.28 and 5.29 respectively.

B-Splines in 2D We calculate the product of two one-dimensional centred B-splines to obtain the two-dimensional centred B-spline.

$$\phi(x{,}y) = \phi(x)\phi(y) \tag{5.30}$$

The two-dimensional B-Spline in Eqn. 5.30 has a rectangular compact support and can be used on rectangular grids as a nodal shape function. The two-dimensional quadratic B-spline, centred at the point (1,1) with its derivatives in x- and y-direction, is presented below.

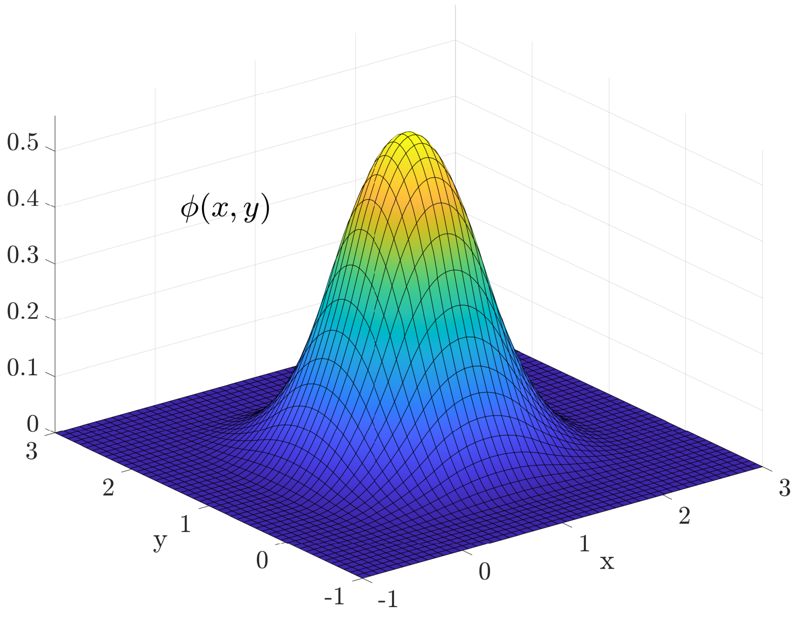


Figure 5.9: Two-dimensional quadratic B-Spline centred at (1,1).

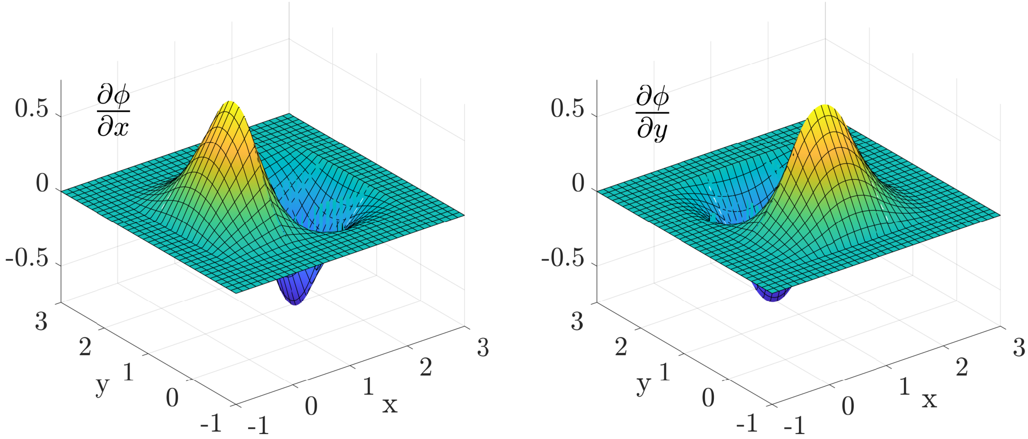


Figure 5.10: Derivative of quadratic B-Spline centred at (1,1) with respect to x.

The B-Spline in Fig. 5.9 and its derivatives in Fig. 5.10 are continuous. The presented two-dimensional B-Spline corresponds to the three inner B-Splines centred at the knots 1.5, 2.5 and 3.5 shown in Fig. 5.6. Modified centred B-Splines need to be used to maintain the partition of unity property in the vicinity of the boundary. This can be achieved by dividing each B-Spline by the sum of all B-Splines defined on the grid. This method will be applied later for the newly introduced Wendland function.

5.3.4 Null Space Filter

A numerical problem in the Particle in Cell Method (PIC), the predecessor of MPM, was described by Brackbill (1988) and Gritton *et al.* (2015) as the so-called ringing instability or, more generally, null space problem. Gritton *et al.* (2015) examined the mapping matrix between MP and nodes for its null space, leading to the loss of information. In particular, the calculation of derivatives using the mapping matrix was investigated. To explain this in more detail, we consider the one-dimensional case with the mapping between MP and nodes as matrix multiplication. For the mapping from MP to nodes, we consider the matrix $\mathbf{\Phi} : \mathbb{R}^{n_p} \to \mathbb{R}^{n_n}$ and the vectors $\mathbf{Y}$ and $\mathbf{y}$, where the first vector consists of values at MP and the second of values at nodes. As a reminder, n_p denotes the total number of MP and n_n the total number of nodes. The mapping from MP to nodes can be written as a linear system of equations,

$$\mathbf{\Phi}\,\mathbf{Y} = \mathbf{y}\,. \tag{5.31}$$

The entries of the matrix $\mathbf{\Phi}$ according to Eqn. 5.6 are the nodal shape functions evaluated at MP. The null space of the matrix $\mathbf{\Phi}$ is defined as the set $\mathcal{N} := \{\mathbf{X} \in \mathbb{R}^{n_p} | \mathbf{\Phi}\mathbf{X} = \mathbf{0}\}$. If there are non-zero vectors in $\mathcal{N}$, they can be added to $\mathbf{Y}$ in Eqn. 5.31 without affecting the resulting nodal values, i.e. $\mathbf{\Phi}\left(\mathbf{Y} + \mathbf{X}\right) = \mathbf{\Phi}\mathbf{Y} + \mathbf{\Phi}\mathbf{X} = \mathbf{y}$. The mapping of derivatives is performed by the matrix d$\mathbf{\Phi}$, which consists of the derivatives of nodal shape functions evaluated at MP. Gritton *et al.* (2015) proposed to use a Null Space Filter derived from the singular value decomposition (SVD) of $\mathbf{\Phi}$ to improve the mapping of derivatives, calculated by $\mathrm{d}\boldsymbol{\Phi}$. The idea behind the Null Space Filter of Gritton *et al.* (2015) is to project the calculated derivatives at MP onto the column space of $\boldsymbol{\Phi}$. Null space filtering smoothes the derivatives but does not necessary increase the accuracy. Let us consider a one-dimensional example of a given function and its derivative:

$$y(x) = \frac{\sin(10\,\pi\,x)}{x+\frac{1}{10}} \tag{5.32a}$$

$$y'(x) = \frac{10\,\pi\cos(10\,\pi\,x)}{x+\frac{1}{10}} - \frac{\sin(10\,\pi\,x)}{(x+\frac{1}{10})^2}\,. \tag{5.32b}$$

We want to obtain the derivative in Eqn. 5.32b at MP from exact values at the nodes. Therefore, we consider the interval [0,1] divided into 20 elements of length 0.05 with two MP per element, each located at 1/4 and 3/4 of the element's length. Mapping is carried out by employing linear shape functions. The unfiltered mapping is performed by the matrix $\mathrm{d}\boldsymbol{\Phi}$, which contains the piecewise constant derivatives of the linear shape functions. The filtered mapping is calculated by projecting the unfiltered values on the column space of $\boldsymbol{\Phi}$. The basis for the column space has the length $l = \mathrm{rank}(\boldsymbol{\Phi})$ and is obtained by the SVD of $\boldsymbol{\Phi}$. For a detailed explanation of the Null Space Filter in MPM the reader is referred to Gritton *et al.* (2015). The comparison of filtered and unfiltered mappings is shown in Fig. 5.11. The unfiltered mapping forms steps as pairs of MP with equal values, while the filtered approximation is smoother between these steps.

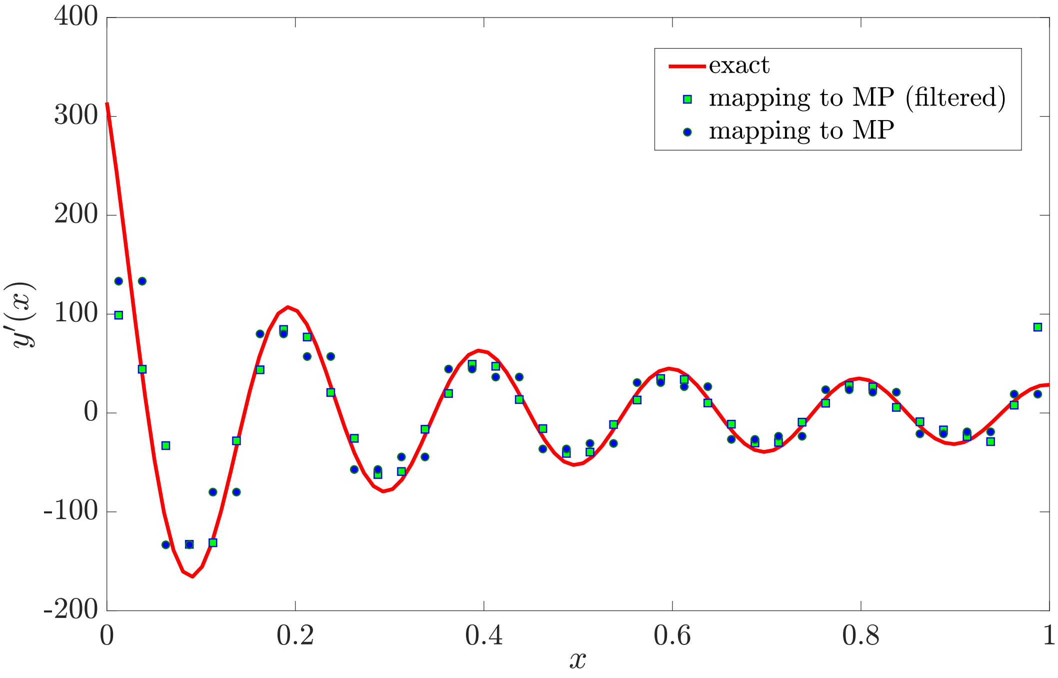


Figure 5.11: Comparison of standard(blue) and filtered(green) approximation of the derivative $y'(x)$ on MP.

The filtered approximation deviates more from the exact solution than the unfiltered at the boundaries.
Eqn. 5.5 shows that in the Standard MPM the mapping from nodes to MP uses the same shape function values as Eqn. 5.6 for the mapping from MP to nodes. Therefore, the resulting matrix for the inverse mapping in the Standard MPM is the transpose of $\mathbf{\Phi}$. Mathematically, an inverse of a matrix is only defined for square matrices with full rank. The null space of invertible matrices contains only the zero vector, making the mapping one-to-one (bijection) in both directions. $\mathbf{\Phi}$ is a $n_n \times n_p-$matrix, which might be not square or rank deficient and therefore not invertible. This also means that, in general, the MP values in $\mathbf{Y}$ can not be exactly recovered after being mapped to the nodes. This leads to the question of improved mapping approaches, which is the main focus of this thesis. It is important to notice that the properties of $\mathbf{\Phi}$ depend not only on the choice of shape functions but also the locations of MP relatively to the nodes.

5.4 Least Squares Approximation

The null space problem described in Sec. 5.3.4 leads to a general question about the mapping between nodes and material points in MPM. It is crucial to investigate which errors are made in mapping, how they depend on the arrangement of MP and how these errors can be reduced. This question can be investigated within the framework of scattered data approximation (Wendland, 2004). Two cases can be considered. On the one hand, the function values at the material points are known and should be reconstructed as accurately as possible at the nodes. On the other hand, the values are known at the nodes and should be reconstructed at the material points. More generally, we have two sets of points where we try to recover functions at the location of the first set from known values at the location of the second set. One direct approach to reconstructing a function at arbitrary points from scattered data is the use of polynomials. This means that for a polynomial of desired order, the coefficients of a polynomial basis in three dimensions, i.e. $\mathbf{p}(\mathbf{x}) = (1 \ x \ y \ \dots \ xyz \ \dots)^T$ with $\mathbf{x} = (x, y, z)$, must be determined so that the resulting error is as small as possible. Let us consider a vector of n given values $\mathbf{u} = (u_1 \ u_2 \ \dots \ u_n)^T$ at the locations $\tilde{\mathbf{x}}_1$, $\tilde{\mathbf{x}}_2$,..., $\tilde{\mathbf{x}}_n$, which leads to the Least Squares Approximation (LS)

$$\mathrm{u_h}(\mathbf{x}) = \sum_{i=1}^{m} p_i(\mathbf{x}) a_i = \mathbf{p}^T(\mathbf{x})\mathbf{a}\,, \tag{5.33}$$

with the unknown coefficients a_i and m being the number of entries in $\mathbf{p}(\mathbf{x})$. To find the coefficients a_i the following system of equations needs to be solved:

$$\mathbf{Pa} = \mathbf{u} \tag{5.34}$$

with $\mathbf{a} = (a_1 \; a_2 \; \dots \; a_m)^T$ and

$$\mathbf{P} = \begin{pmatrix} p_1(\tilde{\mathbf{x}}_1) & p_2(\tilde{\mathbf{x}}_1) & \cdots & p_m(\tilde{\mathbf{x}}_1) \\ p_1(\tilde{\mathbf{x}}_2) & \cdots & p_k(\tilde{\mathbf{x}}_2) & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ p_1(\tilde{\mathbf{x}}_n) & p_2(\tilde{\mathbf{x}}_n) & \cdots & p_m(\tilde{\mathbf{x}}_n) \end{pmatrix}. \tag{5.35}$$

The dimension of $\mathbf{P}$ depends on the number of known values and the order of the polynomial approximation. The entries p_i denote the $i-$th component of $\mathbf{p}(\mathbf{x})$. To solve for the unknown coefficients a_i, the matrix $\mathbf{P}$ needs to be inverted, which in general is not possible as $\mathbf{P}$ is not necessarily a square matrix. In this case, the aim is to minimize the expression $J = \|\mathbf{Pa} - \mathbf{u}\|_2^2$, which measures the error of the approximation. It is helpful to distinguish the two cases in this minimization problem:

- Case 1: If $\mathbf{u}$ is in the column space of $\mathbf{P}$, i.e. $\mathbf{u}$ can be expressed as a linear combination of the columns of $\mathbf{P}$, there is at least one solution $\mathbf{a}$ for Eqn. 5.34. The sum of the solution $\mathbf{a}$ with any vector from the null space of $\mathbf{P}$ is again a solution for which $J = 0$.

- Case 2: If $\mathbf{u}$ is not in the column space of $\mathbf{P}$, i.e. $\mathbf{u}$ can not be expressed as a linear combination of the columns of $\mathbf{P}$, there is no solution $\mathbf{a}$ for Eqn. 5.34. The minimum of $J = \|\mathbf{Pa} - \mathbf{u}\|_2^2$ is obtained through the orthogonal projection of $\mathbf{u}$ onto the column space of $\mathbf{P}$.

The LS solution can be calculated by setting the derivative of J with respect to each a_i equal to zero, forming a system of m linear equations. Hammerquist and Nairn (2017) presented the XPIC method, which uses approximated orthogonal projectors to solve the scenario described in the second case.

5.4.1 Moving Least Squares Approximation

The calculation of the function $\mathrm{u_h}$ in LS uses all known values and applies to the entire domain. When applied in MPM, it is useful to involve only MP near a node or vice versa for reconstruction at the node and neglect distant MP. For this purpose, functions with compact support are often used to weight the influence of the data. This approach is called weighted least squares approximation (WLS). A very similar approach to WLS which is widely used in meshfree methods (Belytschko, Krongauz, *et al.*, 1996), is the Moving Least Squares Approximation (MLS). The MLS approach leads to coefficients $\mathbf{a}$ which, in contrast to the LS, depend on $\mathbf{x}$,

$$\mathrm{u_h}(\mathbf{x}) = \sum_{i=1}^{m} p_i(\mathbf{x}) a_i(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x})\,. \tag{5.36}$$

The solution for the variable coefficients $a_i(\mathbf{x})$ is obtained by considering the modified minimization problem with $J = (\mathbf{P}(\mathbf{x})\mathbf{a}(\mathbf{x}) - \mathbf{u})^T\mathbf{W}(\mathbf{x})(\mathbf{P}(\mathbf{x})\mathbf{a}(\mathbf{x}) - \mathbf{u})$ and the weighting function matrix

$$\mathbf{W}(\mathbf{x}) = \begin{pmatrix} w\left(\mathbf{x} - \tilde{\mathbf{x}}_1\right) & 0 & \cdots & 0 \\ 0 & w\left(\mathbf{x} - \tilde{\mathbf{x}}_2\right) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w\left(\mathbf{x} - \tilde{\mathbf{x}}_n\right) \end{pmatrix}. \tag{5.37}$$

Belytschko, Krongaus, *et al.* (1996) presented an overview of various common weighting functions $w\left(\mathbf{x} - \tilde{\mathbf{x}}_i\right)$. The solution for minimizing J is obtained by setting the derivatives of J with respect to each a_i equal to zero. The resulting system of m equations is (Wang *et al.*, 2018),

$$\mathbf{P}^T\mathbf{W}(\mathbf{x})\mathbf{P}\mathbf{a}(\mathbf{x}) - \mathbf{P}^T\mathbf{W}(\mathbf{x})\mathbf{u} = \mathbf{0}\,. \tag{5.38}$$

For the computation of the coefficient vector $\mathbf{a}$, we introduce some useful abbreviations. We define the matrix $\mathbf{M}(\mathbf{x}) := \mathbf{P}^T\mathbf{W}(\mathbf{x})\mathbf{P}$, which is sometimes referred to as moment matrix and $\mathbf{B}(\mathbf{x}) := \mathbf{P}^T\mathbf{W}(\mathbf{x})$. Now we can write the solution of Eqn. 5.38 as

$$\mathbf{a}(\mathbf{x}) = \mathbf{M}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{u}\,. \tag{5.39}$$

The matrix $\mathbf{M}(\mathbf{x})$ is a weighted sum of matrices of the form $\mathbf{p}(\tilde{\mathbf{x}}_i)\mathbf{p}(\tilde{\mathbf{x}}_i)^T$. $\mathbf{M}(\mathbf{x})$ is for instance a scalar if $\mathbf{p}(\mathbf{x}) = (1)$ or a $3 \times 3-$matrix if $\mathbf{p}(\mathbf{x}) = (1\ x\ y)^T$. The inversion of $\mathbf{M}(\mathbf{x})$ can therefore be carried out efficiently if, $\mathbf{M}(\mathbf{x})$ is invertible and $\mathbf{p}(\mathbf{x})$ of low order and dimension. Inserting Eqn. 5.39 in Eqn. 5.36 leads to the new representation of the approximation,

$$u_h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{M}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{u} = \mathbf{\Phi}^{\mathbf{MLS}}(\mathbf{x})\mathbf{u}\,, \tag{5.40}$$

where, we introduced the MLS shape function $\mathbf{\Phi}^{\mathbf{MLS}}(\mathbf{x})$. For a given vector of data points $\mathbf{u}$, we can obtain the approximated value at an arbitrary point $\overline{\mathbf{x}}$ simply by plugging $\overline{\mathbf{x}}$ in Eqn. 5.40. As the weighting functions in Eqn. 5.37 often have a compact support, $\mathbf{\Phi}^{\mathbf{MLS}}(\mathbf{x})$ is non-zero only for data points in the vicinity of $\overline{\mathbf{x}}$. The accuracy of MLS depends on the distribution of the given data and is generally not easy to assess. Reproducibility is an essential property for approximation methods, as it describes which functions can be exactly reproduced from given data. As mentioned earlier, we use polynomial bases for our approximation. So an obvious question would be whether MLS can reproduce these polynomials. Indeed MLS reproduces all polynomials used in $\mathbf{p}$ by construction. The proof of this fact can be found in Belytschko, Krongauz, *et al.* (1996).

Derivatives of MLS weighting functions Next, we want to obtain the derivatives of $\mathbf{\Phi}^{\mathbf{MLS}}(\mathbf{x})$, which will provide an approximation of the gradients used in MPM. Therefore we rewrite Eqn. 5.40 as a sum over n data points,

$$\mathbf{\Phi}^{\mathbf{MLS}}(\mathbf{x})\mathbf{u} = \sum_{i=1}^{n} \Phi_i^{\mathbf{MLS}}(\mathbf{x})u_i \quad \text{with} \quad \Phi_i^{\mathbf{MLS}}(\mathbf{x}) = w(\mathbf{x} - \tilde{\mathbf{x}}_i)\,\underbrace{\mathbf{p}^T(\mathbf{x})\mathbf{M}^{-1}(\mathbf{x})}_{:=\mathbf{c}^T(\mathbf{x})}\,\mathbf{p}(\tilde{\mathbf{x}}_i)\,. \tag{5.41}$$

Following the work of V. P. Nguyen, Rabczuk, *et al.* (2008) and Huerta *et al.* (2017) we differentiate $\Phi_i^{MLS}(\mathbf{x})$ in Eqn. 5.41 with respect to x,

$$\frac{\partial \Phi_i^{\mathbf{MLS}}(\mathbf{x})}{\partial x} = \frac{\partial \mathbf{c}^T(\mathbf{x})}{\partial x}\mathbf{p}(\tilde{\mathbf{x}}_i)w(\mathbf{x}-\tilde{\mathbf{x}}_i) + \mathbf{c}^T(\mathbf{x})\mathbf{p}(\tilde{\mathbf{x}}_i)\frac{\partial w(\mathbf{x}-\tilde{\mathbf{x}}_i)}{\partial x}\,. \tag{5.42}$$

The derivative of $w(\mathbf{x}-\tilde{\mathbf{x}}_i)$ on the right hand side of Eqn. 5.42 is easily obtained analytically whereas the derivative of $\mathbf{c}^T(\mathbf{x})$ is not straightforward. For the sake of simplicity the derivative of $\mathbf{c}^T(\mathbf{x})$ is sometimes neglected resulting in the so-called diffuse derivative of $\Phi_i^{\mathbf{MLS}}(\mathbf{x})$ (Huerta *et al.*, 2017). The derivative of $\mathbf{c}^T(\mathbf{x})$ as shown in Eqn. 5.42 can be derived following V. P. Nguyen, Rabczuk, *et al.* (2008) and Huerta *et al.* (2017),

$$\frac{\partial \mathbf{c}(\mathbf{x})}{\partial x} = \mathbf{M}^{-1}(\mathbf{x})\left[\frac{\partial \mathbf{p}(\mathbf{x})}{\partial x} - \frac{\partial \mathbf{M}(\mathbf{x})}{\partial x}\mathbf{c}(\mathbf{x})\right]\,, \tag{5.43}$$

where $\partial \mathbf{p}(\mathbf{x})/\partial x = (\partial p_1/\partial x \;\; \partial p_2/\partial x \; \ldots \; \partial p_m/\partial x)^T$ is the derivative of each component of the polynomial basis and $\partial \mathbf{M}(\mathbf{x})/\partial x = \sum_{i=1}^{n}[\partial w(\mathbf{x}-\tilde{\mathbf{x}}_i)/\partial x]\mathbf{p}(\tilde{\mathbf{x}}_i)\mathbf{p}^T(\tilde{\mathbf{x}}_i)$ contains only the derivatives of weighting functions. The derivative with respect to x of an unknown function given by the data set $\mathbf{u}$, can now be approximated as

$$\frac{\partial u_h(\mathbf{x})}{\partial x} = \sum_{i=1}^{n} \frac{\partial \Phi_i^{\mathbf{MLS}}(\mathbf{x})}{\partial x} u_i\,. \tag{5.44}$$

Similarly, derivatives for y and z can be obtained. With Eqn. 5.41 and Eqn. 5.44 the necessary approximations are derived for the inclusion in the MPM algorithm.

MLS in MPM The successful application of MLS in MPM has already been shown in several publications (Sulsky and Gong, 2016; Wobbes *et al.*, 2019; Tran *et al.*, 2019). Sulsky and Gong (2016) introduced the improved MPM (iMPM) using MLS to recover nodal and Gauss Point (GP) values from MP. It was shown that the Standard MPM can be interpreted as MLS with $\mathbf{p}(\mathbf{x}) = (1)$ and $w(\mathbf{x}_i - \mathbf{x}_p) = m_p N_i(\mathbf{x}_p)$, where $N_i(\mathbf{x}_p)$ is the nodal linear shape function described in Eqn. 5.3. In the framework of MLS also the BSMPM can be obtained by using again $\mathbf{p}(\mathbf{x}) = (1)$ and replacing the weighting functions by B-Splines. In the following, we will investigate the application of higher order polynomial bases $\mathbf{p}(\mathbf{x})$ and different weighting functions. Let us therefore consider the two-dimensional case with $\mathbf{p}(\mathbf{x}) = (1 \; x \; y)^T$. We assume the weighting function to be defined as the product of two one-dimensional functions, i.e. $w(\mathbf{x}-\tilde{\mathbf{x}}_i) = w(x-\tilde{x}_i)w(y-\tilde{y}_i)$. The matrix $\mathbf{M}(\mathbf{x})$ can now be written as

$$\mathbf{M}(\mathbf{x}) = \sum_{i=1}^{n} w(x-\tilde{x}_i)w(y-\tilde{y}_i)\begin{pmatrix} 1 & \tilde{x}_i & \tilde{y}_i \\ \tilde{x}_i & \tilde{x}_i^2 & \tilde{x}_i\tilde{y}_i \\ \tilde{y}_i & \tilde{y}_i\tilde{x}_i & \tilde{y}_i^2 \end{pmatrix}\,. \tag{5.45}$$

The matrices on the right side of Eqn. 5.45 are obtained from the dyadic products $(1 \; \tilde{x}_i \; \tilde{y}_i)^T(1 \; \tilde{x}_i \; \tilde{y}_i)$, which are rank one. In order to obtain an invertible matrix $\mathbf{M}(\mathbf{x})$

at least three data points are needed as the sum of less than three such matrices would lead to a matrix with rank lower than three. Nevertheless, three data points might also be too less if the dyadic vectors $(1\ \tilde{x}_i\ \tilde{y}_i)$ are linearly dependent. To shed more light on the problem let us consider four data points $\tilde{\mathbf{x}}_1 = (3\,,2)$, $\tilde{\mathbf{x}}_2 = (2\,,1)$, $\tilde{\mathbf{x}}_3 = (1\,,2)$ and $\tilde{\mathbf{x}}_4 = (1\,,0)$. The corresponding vectors $\mathbf{p}(\tilde{\mathbf{x}}_i)$ are three-dimensional with the entry one in the first coordinate. In Fig. 5.12 the vectors $\mathbf{p}(\tilde{\mathbf{x}}_i)$, for the given data points, are shown in three dimensions.

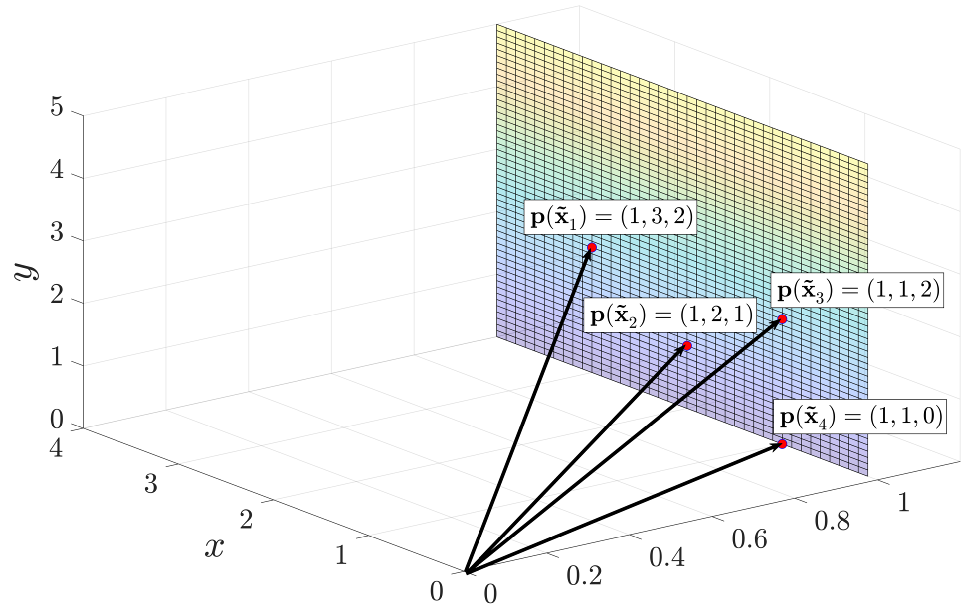


Figure 5.12: Configuration of four different material points $(1\ x_p\ y_p)$ for the basis $\mathbf{p}(\mathbf{x}) = (1\ x\ y)^T$.

By definition all of them are located on the plane $(1\ x\ y)$. Let us assume that the weighting functions of all these points are non-zero for an arbitrary point $\overline{\mathbf{x}}$. Using the vectors $\tilde{\mathbf{x}}_1 = (3\,,2)$, $\tilde{\mathbf{x}}_2 = (2\,,1)$ and $\tilde{\mathbf{x}}_4 = (1\,,2)$ will result in a rank two moment matrix as $\mathbf{p}(\tilde{\mathbf{x}}_1) + \mathbf{p}(\tilde{\mathbf{x}}_4) = 2\mathbf{p}(\tilde{\mathbf{x}}_2)$. In contrast, using $\tilde{\mathbf{x}}_1$, $\tilde{\mathbf{x}}_2$, $\tilde{\mathbf{x}}_3$ would lead to an invertible $\mathbf{M}(\mathbf{x})$. Obviously the same is true, if all four points are used. If $\mathbf{M}(\mathbf{x})$ is not invertible or ill-conditioned the approximation in Eqn. 5.41 is not accurate. In higher dimensions and with higher order polynomial bases, the requirements on the number and location of data points become more demanding as the dimension of $\mathbf{M}(\mathbf{x})$ increases. To meet this requirements, the compact support of the weighting functions can be increased leading to not only more data points being involved in the calculation but also increasing the computational costs. This can be done adaptively by checking the condition number of the matrix and gradually increasing the compact support until a predefined threshold value is reached. Another approach is to use the Pseudo Inverse of $\mathbf{M}(\mathbf{x})$, which can be obtained from the SVD of $\mathbf{M}(\mathbf{x})$ by inverting the non-zero entries of the diagonal matrix of the decomposition. In addition to both the approaches, a third method is introduced which will be explained in detail in Ch. 6.

Chapter 6

Improvements of MPM

In this chapter a regularization method is proposed to improve the robustness and accuracy of MLS in MPM. Further a new weighting function is introduced for MPM calculations. Both novelties are benchmarked against other methods for small and large deformations to highlight their improvements.

6.1 Regularized MLS

Instead of increasing the compact supports of the weighting functions, we propose a different approach to make $\mathbf{M}(\mathbf{x})$ invertible. Following Wang *et al.* (2018), a regularized MLS, which results in the modified form of Eqn. 5.38

$$[\mathbf{P}^T\mathbf{W}(\mathbf{x})\mathbf{P}+\mathbf{I}]\mathbf{a}-\mathbf{P}^T\mathbf{W}(\mathbf{x})\mathbf{u}-\lambda\mathbf{I}=\mathbf{0}\,, \tag{6.1}$$

is proposed here to improve the robustness of iMPM. The regularized iMPM (riMPM) introduces the regularization parameter λ, which prevents $\mathbf{M}(\mathbf{x})$ from becoming singular. The modified coefficients $\mathbf{a}(\mathbf{x})$ corresponding to Eqn. 6.1 are

$$\mathbf{a}(\mathbf{x})=\left(\mathbf{M}(\mathbf{x})+\lambda\mathbf{I}\right)^{-1}\mathbf{B}(\mathbf{x})\mathbf{u}\,. \tag{6.2}$$

The choice of λ is not obvious and can affect the results in a wrong way (Brunton and Kutz, 2019). Regarding the computational efficiency, the regularization method is much faster than the computation of the Pseudo Inverse or adaptive increase of compact supports. For the previously shown case of $\mathbf{M}(\mathbf{x})$ being a $3\times3-$matrix, the inverse can easily be computed by the Rule of Sarrus.

6.2 Wendland weighting functions

In addition to the widely used weighting functions in MPM a different function is proposed here. The Wendland function family derived in Wendland (1995) is a suitable candidate for a weighting function in MPM. Only one kind of Wendland functions is used in the following for comparison with the well established B-Spline function in the MPM.

$$w(r)=\begin{cases}\left(\frac{|r|}{|h|}-1\right)^4\left(\frac{4\,|r|}{|h|}+1\right) & \text{if} \quad |r/h|<1\\ 0 & \text{otherwise}\end{cases} \tag{6.3}$$

The function in Eqn. 6.3 is non-zero on the interval $(-h\,,h)$ and equals one at $r = 0$. We will use this function on rectangular grids and replace the radius r by x and y, similar to what was done for B-Splines. In the following, we will compare the implementations of all presented methods and see how well they are performing on different data sets. It should be noted that the use of the Wendland function in Eqn. 6.3 on a rectangular grid does not satisfy the Partition of Unity property. To overcome this issue each weighting function is divided by the sum of all weighting functions.

Partition of Unity In order to satisfy the Partition of Unity property for a weighting function and in particular for the presented function in Eqn. 6.3, we follow Shepard (1968) and Nealen (2004) and define the new normalized weighting function

$$W_i(x) = \frac{w_i(x)}{\sum_{i=1}^{n} w_i(x)}\,. \tag{6.4}$$

In Eqn. 6.4 the index i denotes the knot around which the function is centred. It is easy to check that the function in Eqn. 6.4 satisfies the Partition of Unity property, i.e. $\sum_{i=1}^{6} W_i(x) = 1$. Let us consider the one-dimensional knot vector $\{0; 1; 2; 3; 4; 5\}$ of length $n = 6$ with $h = 1$ and construct the six resulting functions for Eqn. 6.4 as well as their sum to check the desired properties. In Fig. 6.1 the resulting normalized Wendland functions are shown. Their sum gives the desired property and therefore is one over the whole interval $[0\,,5]$.

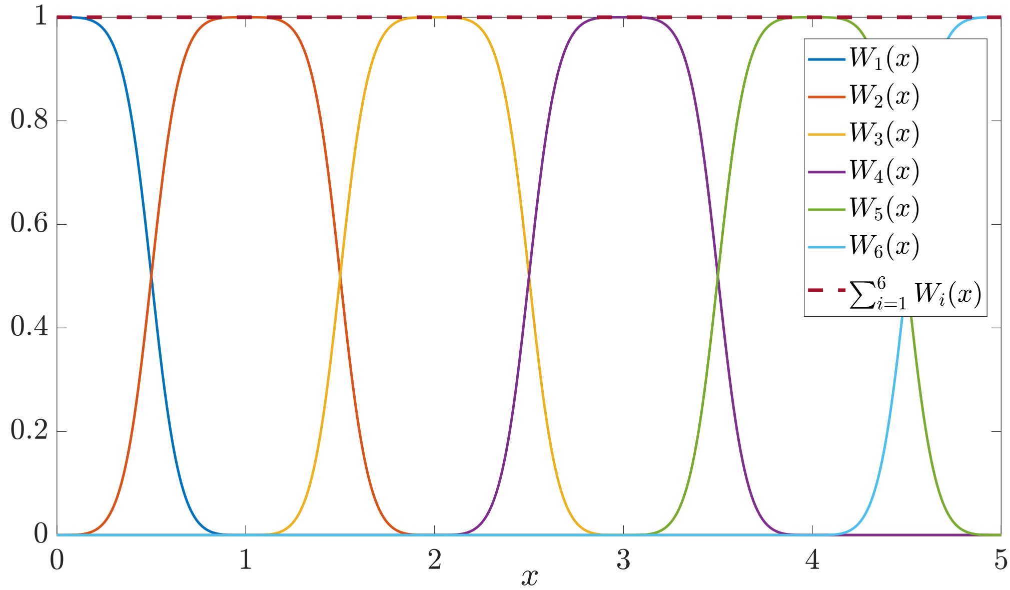


Figure 6.1: Normalized Wendland functions for the knot vector $\{0; 1; 2; 3; 4; 5\}$ and their sum over the interval $[0\,,5]$.

6.3 Comparison of methods on structured data

In the following we compare the mentioned MPM approximations on different point sets. We consider the following function,

$$z(x,y) = 3\,\mathrm{e}^{-(y+1)^2-x^2}\,(x-1)^2 - \frac{\mathrm{e}^{-(x+1)^2-y^2}}{3} + \mathrm{e}^{-x^2-y^2}\left(10\,x^3 - 2\,x + 10\,y^5\right), \tag{6.5}$$

which can be called in MATLAB by the command *peaks*, on the interval $\Omega = [-2\,,2] \times [-2\,,2]$. We focus on the approximation of MP values to nodes, which is performed in the beginning of each MPM computation cycle. Therefore, we assume a structured nodal grid with $21 \times 21-$nodes, which will be the set where the approximation is evaluated. We are considering two different scenarios for the MP data. In the first scenario the MP are arranged in a structured manner on a $41 \times 41-$grid as shown in Fig. 6.2. In the second scenario, the MP data is randomly distributed at 100 locations. First we compare the case of a constant polynomial basis $\mathbf{p}(\mathbf{x}) = (1)$. In this case, the inverse of $\mathbf{M}(\mathbf{x})$ is trivial as $\mathbf{M}(\mathbf{x})$ is a scalar. Therefore we compare different weighting functions.

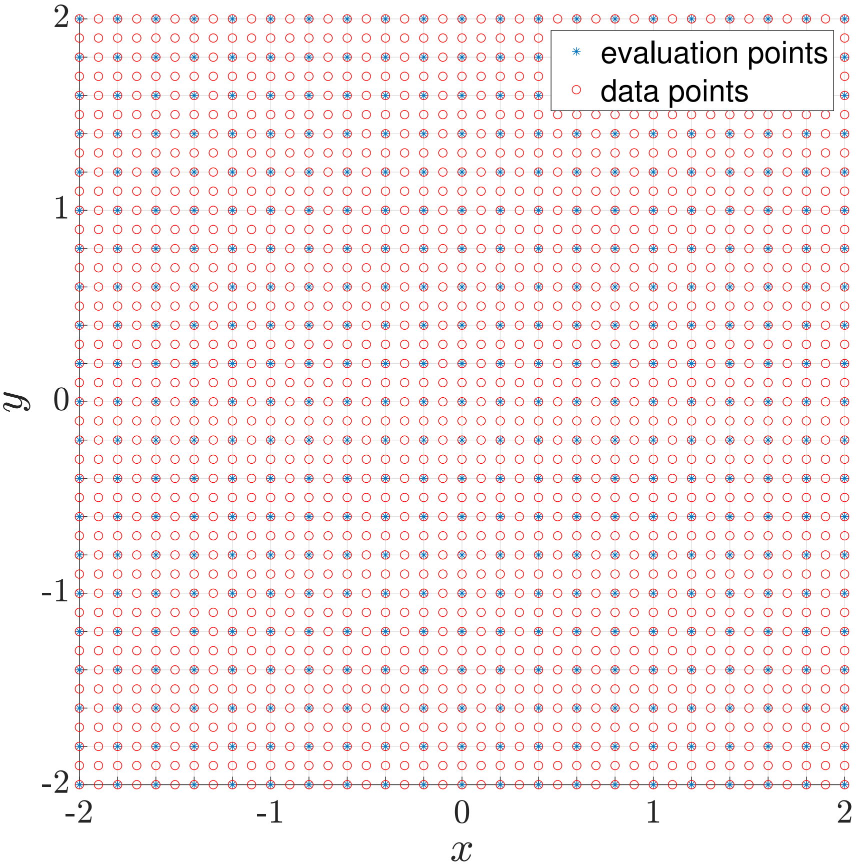


Figure 6.2: Case 1: Structured arrangement of MP (red) and nodes (blue). The function in Eqn. 6.5 is given at the data points and approximated at the evaluation points.

In Fig. 6.3, the MLS solutions for different weighting functions are shown. The relative error using the L_2−norm is calculated by $||\mathbf{u}-\mathbf{u}_h||_2/||\mathbf{u}||_2$, where $\mathbf{u}$ is the vector containing the exact and $\mathbf{u}_h$ the vector containing the approximated values at the nodes. The compact supports for the B-Splines are defined according to the nodal grid size $h = 0.2$ so that they form a partition of unity inside Ω (linear:$2h$, quadratic:$3h$, cubic:$4h$). At the boundary, no modified Splines were used, which leads to inaccuracies in the approximation of the boundary values. For the Wendland function the compact support was chosen to be $2.6h$.

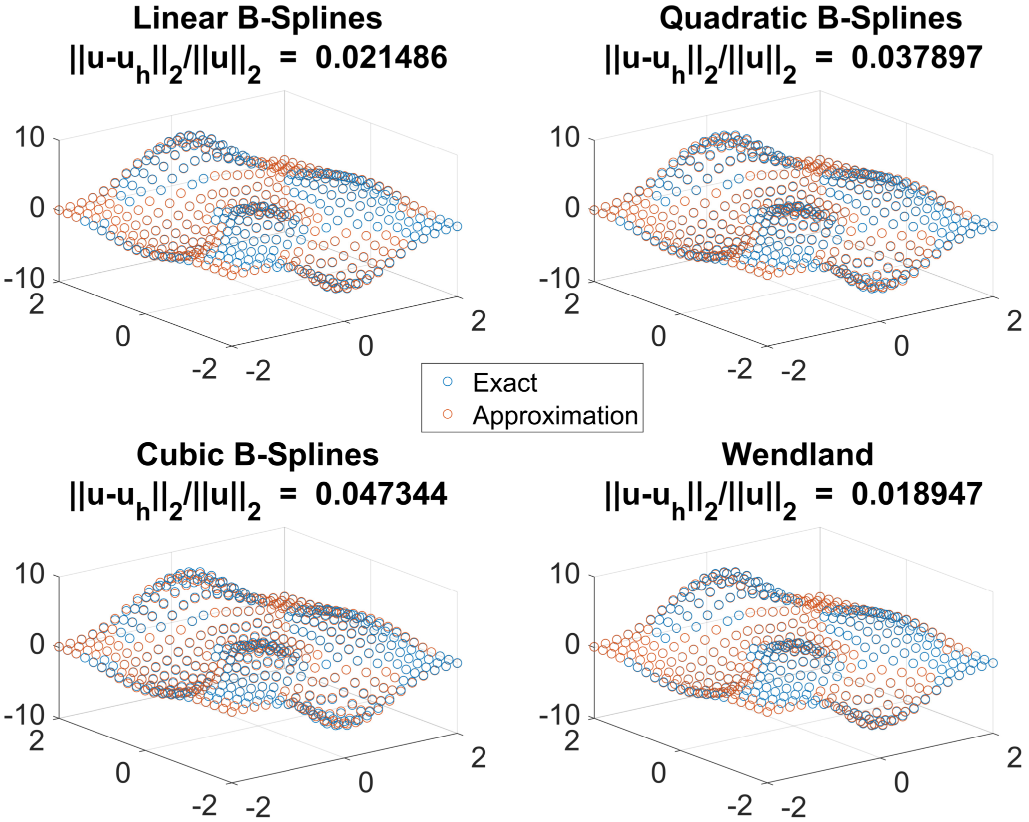


Figure 6.3: Comparison of approximations using different weighting functions for $\mathbf{p}(\mathbf{x}) = (1)$. The approximations are compared with the exact solution in Eqn. 6.5 on the structured arranged nodes.

In Fig. 6.4 the absolute error of exact and approximated values $|u(x_i,y_j) - u_h(x_i,y_j)|$ is shown. Dark blue areas indicate small errors, while yellow areas indicate large deviations. It can be seen that the highest errors occur at the boundary.

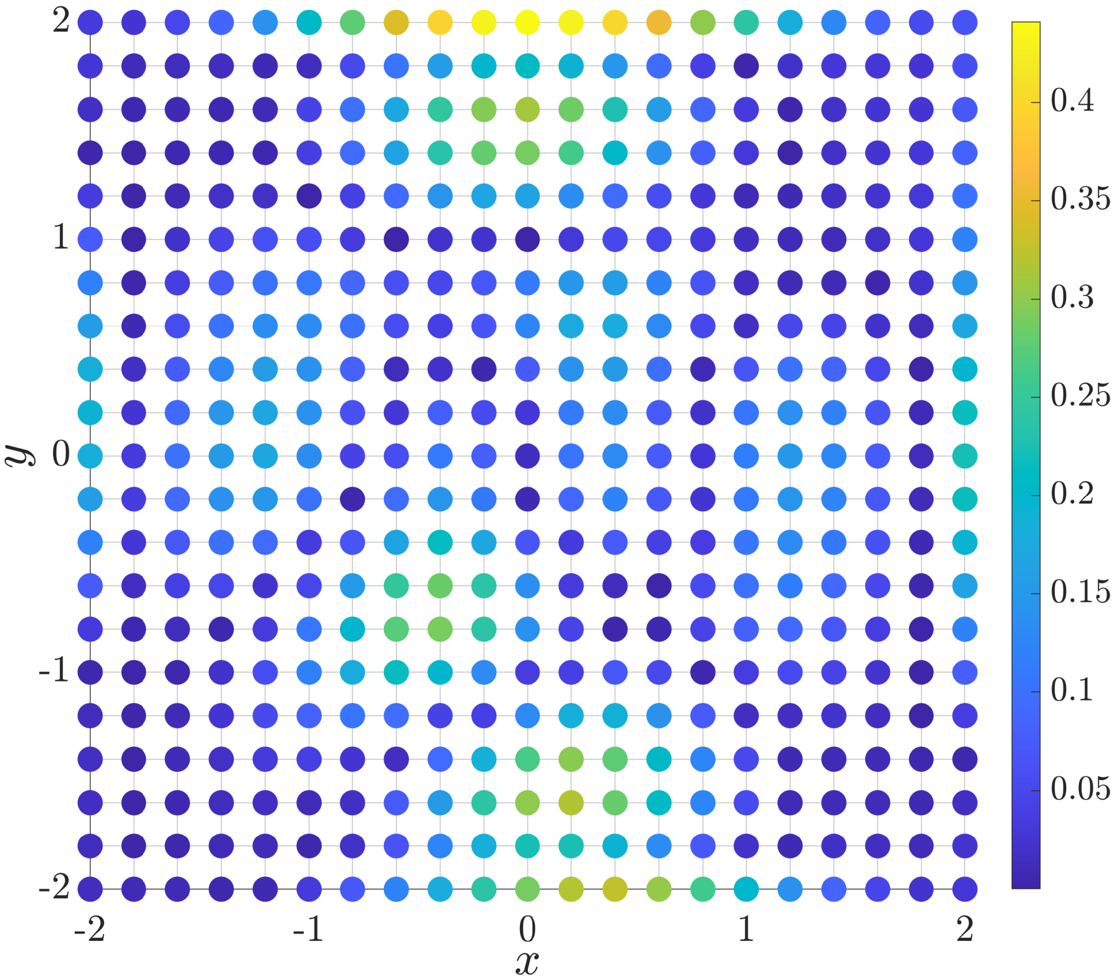


Figure 6.4: The distribution of the absolute error $|u(x_i,y_j) - u_h(x_i,y_j)|$ at each node for the cubic B-Spline weighting function and the polynomial basis $\mathbf{p}(\mathbf{x}) = (1)$.

In Fig. 6.5 the polynomial basis $\mathbf{p}(\mathbf{x}) = (1\ x\ y)^T$ was used for the MLS approximation. The L_2-error is significantly smaller for all weighting functions compared to the results obtained for $\mathbf{p}(\mathbf{x}) = (1)$ in Fig. 6.3. The inversion of $\mathbf{M}(\mathbf{x})$ is performed using the pinv function in MATLAB.

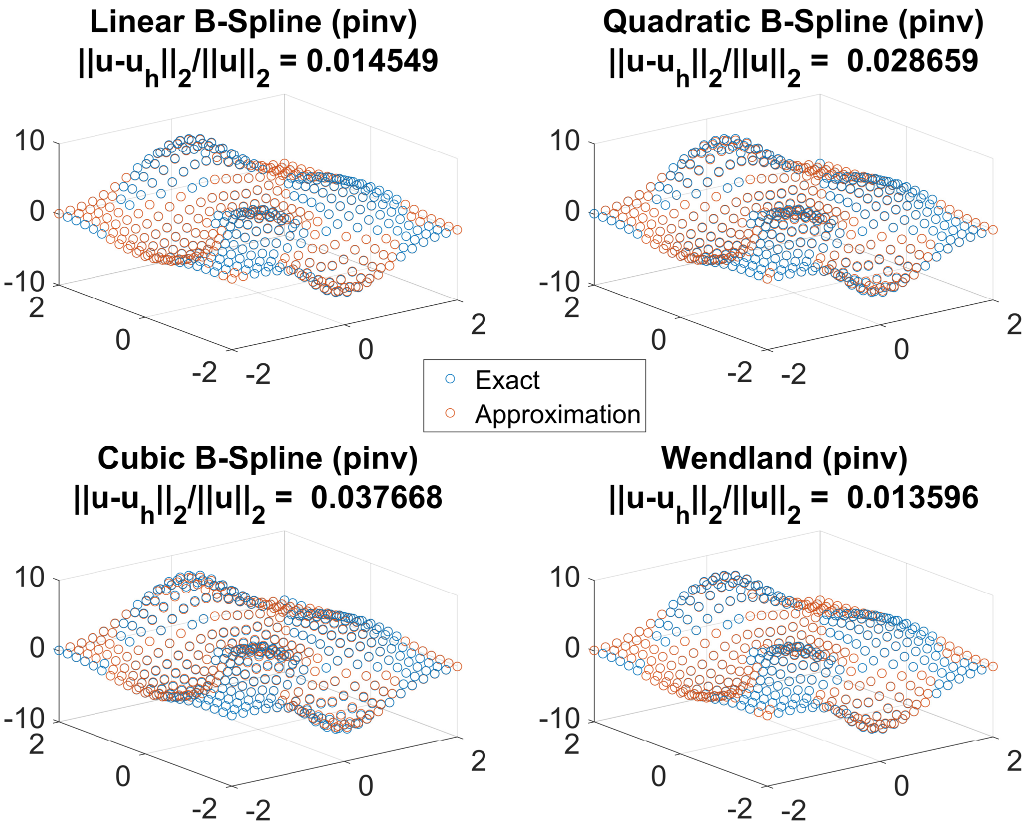


Figure 6.5: Comparison of approximations using different weighting functions with the polynomial basis $\mathbf{p}(\mathbf{x}) = (1\ x\ y)^T$. The abbreviation 'pinv' is referred to the Pseudo Inverse of $\mathbf{M}(\mathbf{x})$, which is computed by the MATLAB function pinv.

In Fig. 6.6, the regularized matrix $\mathbf{M}(\mathbf{x})$ with $\lambda = 0.00001$ was inverted instead of using the *pinv* function. The results show errors similar to Fig. 6.5. Nevertheless the regularization is computationally cheaper as the *pinv* function, which requires the SVD of $\mathbf{M}(\mathbf{x})$.

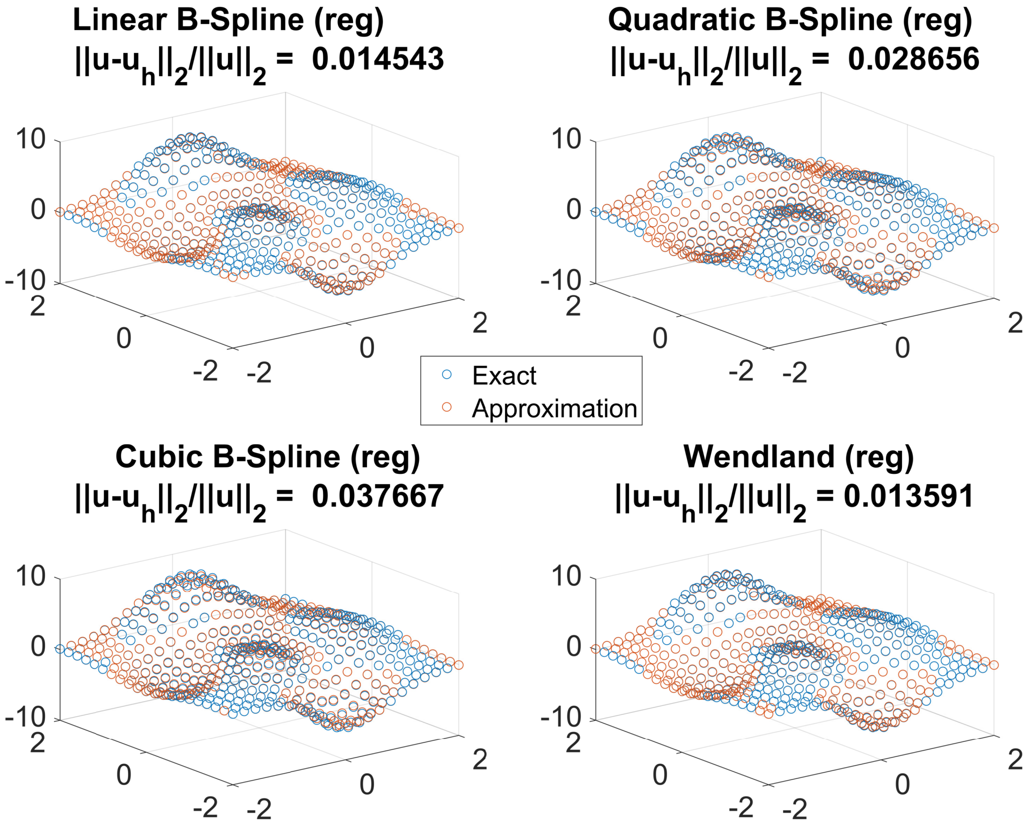


Figure 6.6: Comparison of approximations using different weighting functions with the polynomial basis $\mathbf{p}(\mathbf{x}) = (1\ x\ y)^T$. The abbreviation 'reg' is referred to the regularization of $\mathbf{M}(\mathbf{x})$ for $\lambda = 0.00001$.

The influence of the regularization parameter λ on the L_2-error for $\mathbf{p}(\mathbf{x}) = (1\ x\ y)^T$ in combination with the Wendland weighting function is shown in Fig. 6.7. On structured data, as shown in Fig. 6.2, the matrix $\mathbf{M}(\mathbf{x})$ is invertible and does not need to be regularized. The *pinv* function in this case becomes an expensive way of computing the inverse. What should be noted here is that with increasing λ, the error increases. Whereas for values lower than 10^{-4}, the regularization is not affecting the inversion of $\mathbf{M}(\mathbf{x})$. Therefore, the challenge is to determine a suitable value for λ without introducing new errors.

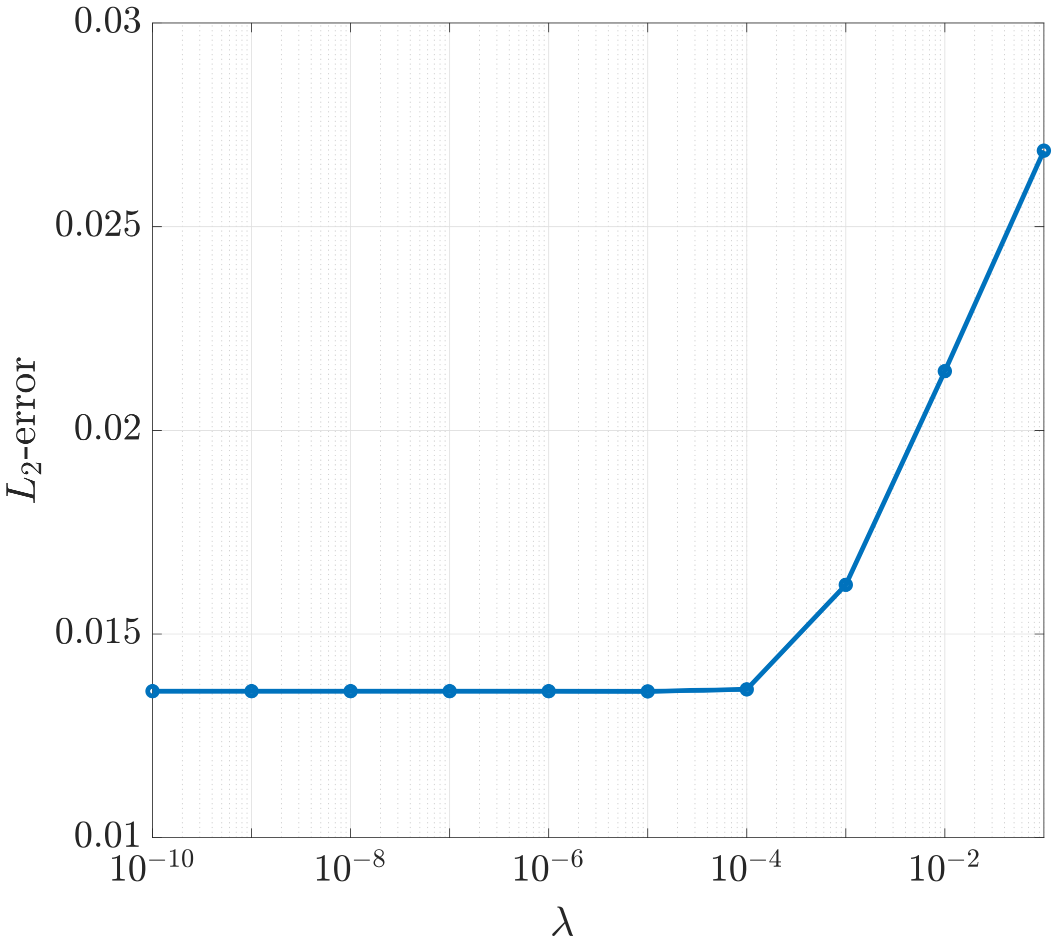


Figure 6.7: Influence of regularization parameter λ on the L_2-error using the Wendland weighting function for the polynomial basis $\mathbf{p}(\mathbf{x}) = (1\ x\ y)^T$.

Comparison of methods on unstructured data On structured and dense data as shown in Fig. 6.2, quite accurate approximations are obtained for all methods. The challenging case is to obtain an acceptable approximation from random and sparse arrangements of data points. In particular, simulations performed in MPM mainly consider large deformation processes, where the arrangements of MP can become arbitrary. To put it differently, an accurate approximation method of arbitrary scattered data is essential for MPM to simulate large deformations. In Fig. 6.8, the random distribution of 100 data points is shown. We now apply the MLS to reconstruct the function at the structured nodes or evaluation points.

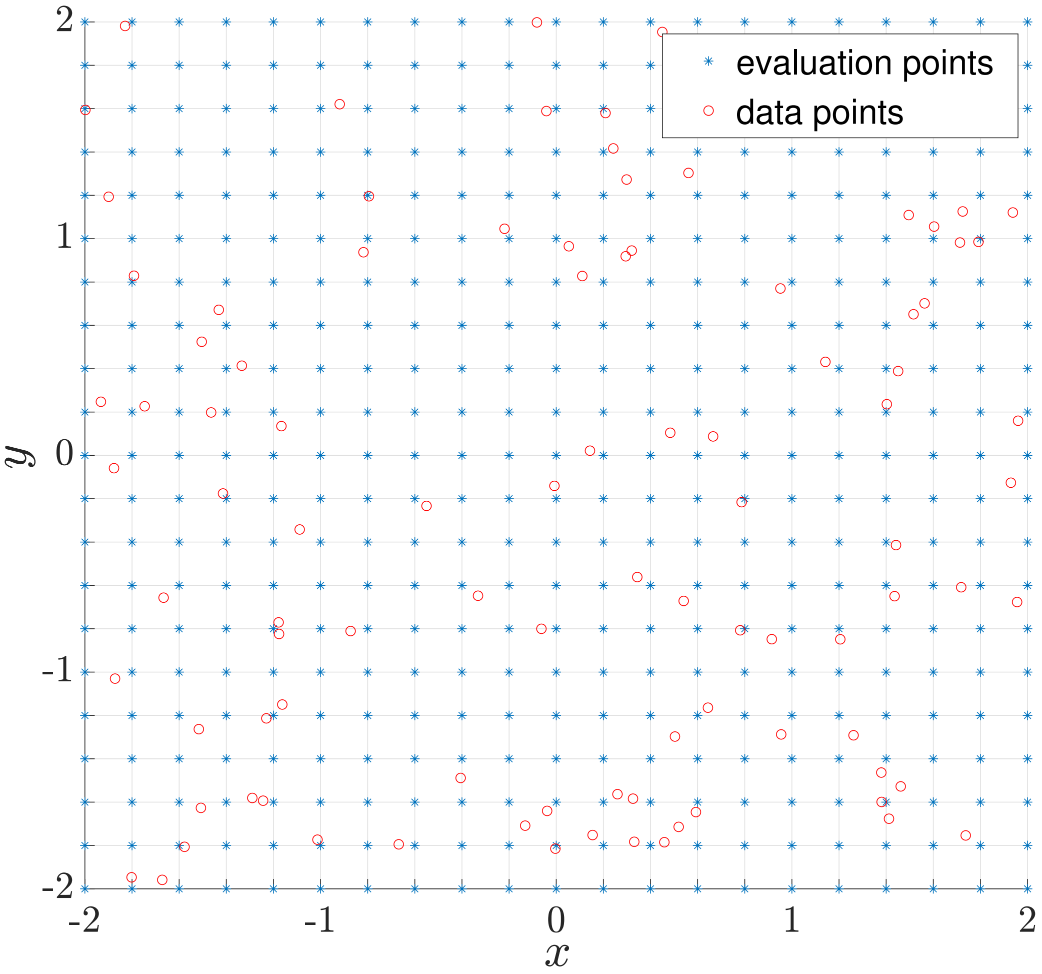
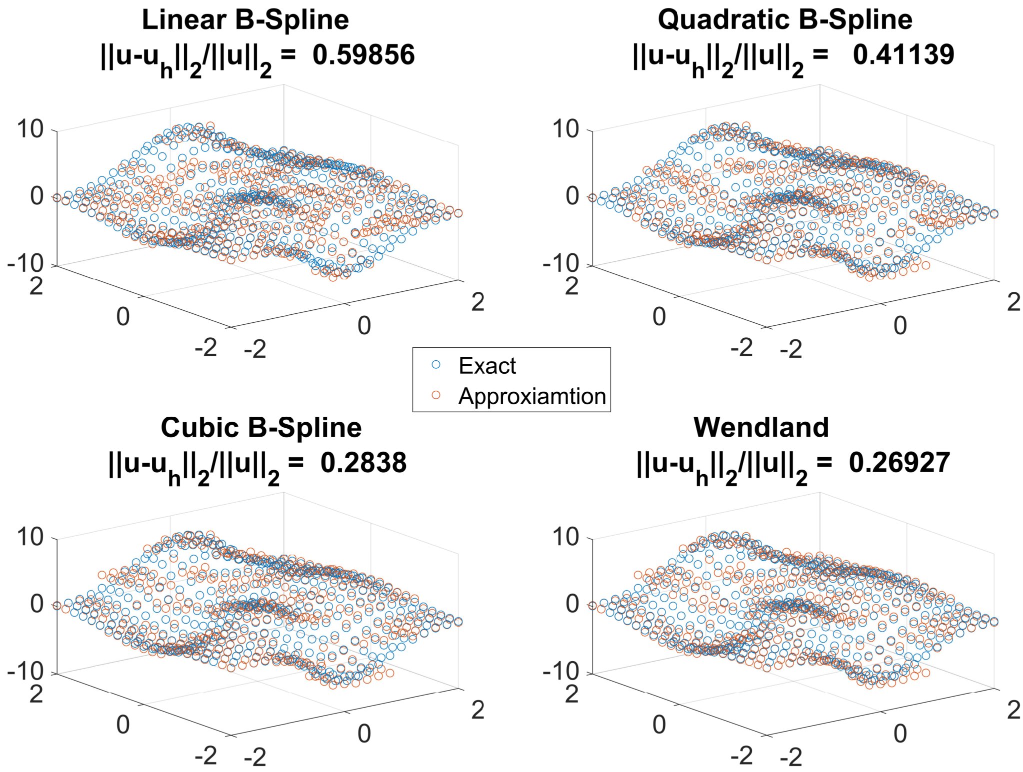


Figure 6.8: Case 2: Unstructured arrangement of 100 MP(red) and nodes(blue). The function in Eqn. 6.5 is given at the data points and approximated at the evaluation points.

In Fig. 6.9, the MLS solutions for different weighting functions from unstructured scatter data are shown. The compact supports for the B-Splines are defined according to the nodal grid size $h = 0.2$, so that they form a partition of unity inside Ω (linear:$2h$, quadratic:$3h$, cubic:$4h$). For the Wendland function the compact support was chosen to be $4h$.


Linear B-Spline
$||u-u_h||_2/||u||_2 =$ **0.59856**

Quadratic B-Spline
$||u-u_h||_2/||u||_2 =$ **0.41139**

Cubic B-Spline
$||u-u_h||_2/||u||_2 =$ **0.2838**

Wendland
$||u-u_h||_2/||u||_2 =$ **0.26927**

Figure 6.9: Comparison of approximations using different weighting functions and $\mathbf{p}(\mathbf{x}) = (1)$ for the point set shown in Fig. 6.8. The approximations are compared with the exact solution in Eqn. 6.5 on the structured arranged nodes.

The errors in Fig. 6.10 increase compared to the case of $\mathbf{p}(\mathbf{x}) = (1)$ in Fig. 6.9. This is due to the unfavourable arrangement of points, which causes a ill-conditioned matrix $\mathbf{M}(\mathbf{x})$. In particular, for the case of quadratic B-Splines with a compact support of $3h$, the arrangements of data points, as shown in Fig. 6.8, leads to disadvantageous conditions for the approximation. To improve the condition of $\mathbf{M}(\mathbf{x})$ and therefore the accuracy of the approximation, the regularization in Eqn. 6.1 is used.

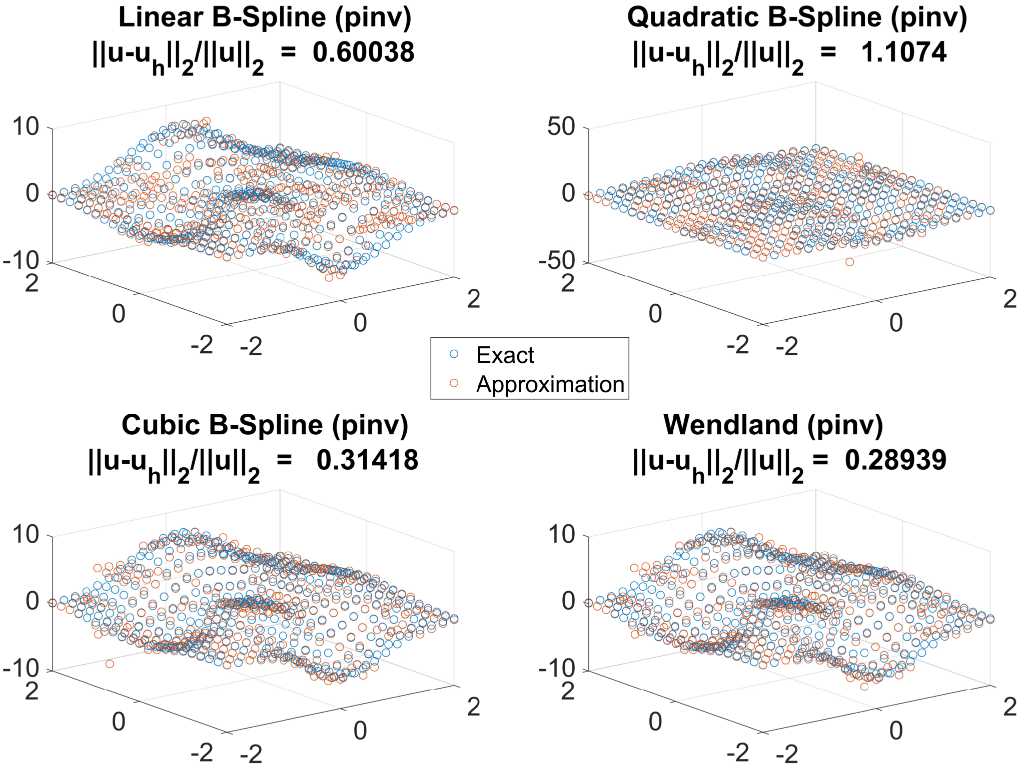


Figure 6.10: Comparison of approximations from unstructured data points using different weighting functions with the polynomial basis $\mathbf{p}(\mathbf{x}) = (1\ x\ y)^T$. The abbreviation 'pinv' is referred to the Pseudo Inverse of $\mathbf{M}(\mathbf{x})$, which is computed by the MATLAB function *pinv*.

The regularized approximation in Fig. 6.11 shows an improvement compared to the results obtained by the Pseudo-Inverse in Fig. 6.10. In particular, the case of the quadratic B-Spline weighting function shows a significant improvement.

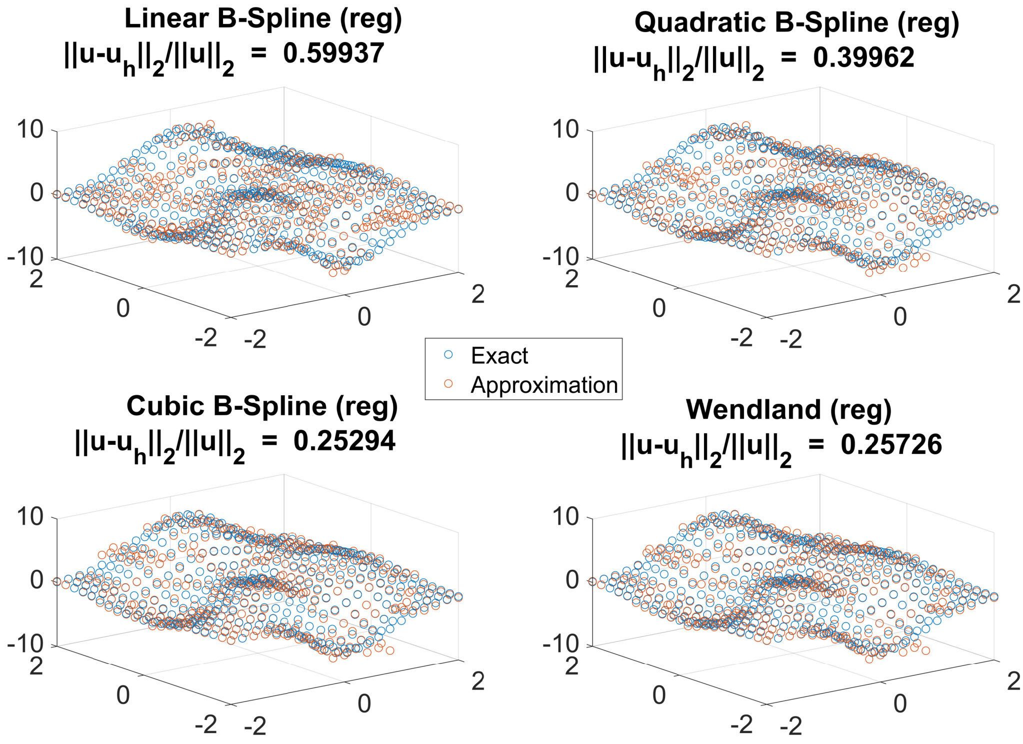


Figure 6.11: Comparison of approximations from unstructured data points using different weighting functions with the polynomial basis $\mathbf{p}(\mathbf{x}) = (1\ x\ y)^T$. The abbreviation 'reg' is referred to the regularization of $\mathbf{M}(\mathbf{x})$ for $\lambda = 0.00001$.

The choice of a suitable regularization parameter λ in the case of scattered data points is more complicated than in the case shown in Fig. 6.7. In Fig. 6.12 the influence of the regularization parameter λ on the L_2-error is shown. As in the case shown in Fig. 6.8, the matrix $\mathbf{M}(\mathbf{x})$ becomes ill-conditioned for some evaluation points, and therefore the L_2-error increases if λ is too small. On the other hand, the error increases also if λ is too big. Therefore, an interval can be identified in Fig. 6.12, where λ causes the smallest L_2-error. If $\mathbf{M}(\mathbf{x})$ becomes ill-conditioned, depends also on the compact support of the weighting functions. Either there are too few data points inside the compact support or too many linearly dependent ones.

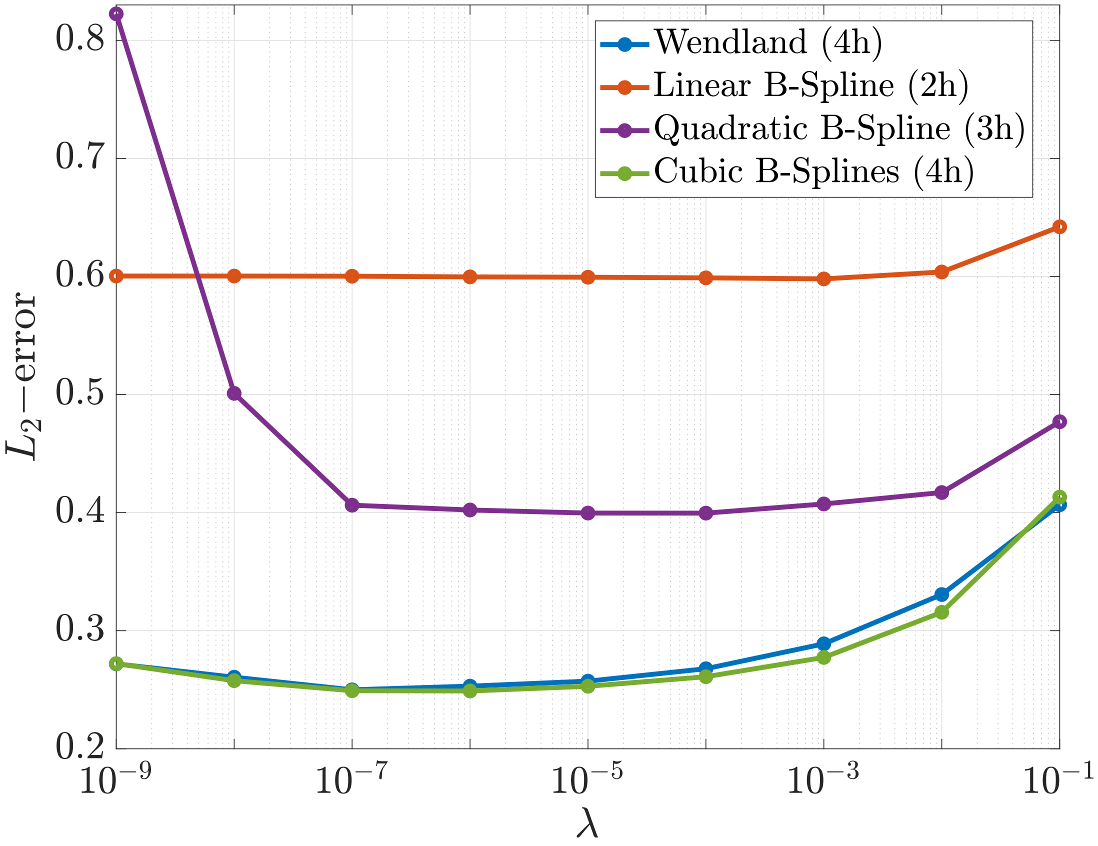


Figure 6.12: Influence of regularization parameter λ on the L_2-error using different weighting function for the polynomial basis $\mathbf{p} = (1 \ x \ y)$.

6.3.1 Adaptive Regularization

In the previous examples of regularization we have used the same λ for all evaluation points. Two issues arise with this approach. The first is that we do not know how to choose λ and if the moment matrix needs to be regularized. The second is that some matrices might be zero and will be strongly affected by λ even though their Pseudo-Inverse is zero. Therefore, an adaptive regularization is proposed here.

To verify if regularization should be applied, two checks are made:

- Check if all entries of $\mathbf{M}(\mathbf{x})$ are zero, e.g. $||\mathbf{M}(\mathbf{x})|| = 0$, if yes set $\mathbf{M^{-1}}(\mathbf{x}) = \mathbf{0}$
- Check if the condition of $\mathbf{M}(\mathbf{x})$ is large, if yes start regularization with very small λ until $\mathbf{M}(\mathbf{x})$ is not ill-conditioned

The second point still requires the definition of a very small λ and of an acceptable matrix condition number. For instance in MATLAB regularization could be applied if the condition number exceeds the value 10^{15}. The reason is that the MATLAB function *inv* is giving warnings of inaccuracy for higher values. If the problem permits, decreasing this value might speed up the calculation. The regularization parameter λ could be estimated by the smallest (non-zero) singular values of $\mathbf{M}(\mathbf{x})$, which is computationally expensive. If the problem permits, increasing the initial $\lambda-$value might speed up the calculation.

6.4 Method of Manufactured Solutions(MMS)

In recent years, the method of manufactured solutions (MMS) has become a common benchmark for the convergence rate of different MPM variants. The method was introduced by Knupp (2002) to verify non-linear codes when a classical analytical solution is missing. The idea is to assume, for example, a simple displacement field and calculate backwards the required body forces. The application of MMS for MPM was first shown by Wallstedt and Guilkey (2008). Since then, most of the convergence investigations in MPM have been carried out using MMS. In the following, a brief explanation of the method for the two-dimensional case is given. For more details the reader is referred to Sulsky and Gong (2016). We assume a two-dimensional displacement field of the form

$$\mathbf{u}(X,Y,t) = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} B\sin(\pi X)\sin(c\pi t) \\ B\sin(\pi Y)\sin(c\pi t) \end{pmatrix} . \tag{6.6}$$

We used the Lagrangian coordinates in Eqn. 6.6. The assumed displacement field satisfies the boundary conditions $\mathbf{u}(0,Y,t) = \mathbf{u}(X,0,t) = \mathbf{u}(1,Y,t) = \mathbf{u}(X,1,t) = 0$ and the initial conditions $\mathbf{u}(X,Y,0) = 0$. For the initial velocity, the assumed displacement in Eqn. 6.6 results in

$$\mathbf{v}(X,Y,0) = c\pi B \begin{pmatrix} \sin(\pi X) \\ \sin(\pi Y) \end{pmatrix} . \tag{6.7}$$

The amplitude B can be chosen to mimic large and small deformation problems. The coefficient c is the wave propagation speed. In the considered case it is defined as $c = \sqrt{E/\rho}$. We now can compute the resulting deformation gradient as

$$\begin{aligned} \mathbf{F}(X,Y,t) = \mathbf{I} + \mathbf{grad}(\mathbf{u}) &= \begin{pmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{pmatrix} \\ &= \begin{pmatrix} 1 + \pi B\cos(\pi X)\sin(c\pi t) & 0 \\ 0 & 1 + \pi B\cos(\pi Y)\sin(c\pi t) \end{pmatrix} . \end{aligned} \tag{6.8}$$

The deformation gradient is now used with the Neo-Hookean material model in Eqn. 5.24 to determine the body forces. In the Lagrangian coordinates we use the first Piola-Kirchhoff stress tensor $\mathbf{P}$, which is connected to the Cauchy stress tensor through $\mathbf{P}\mathbf{F}^T = J\boldsymbol{\sigma}$. The resulting body forces are calculated as

$$\mathbf{f}_{ext} = \pi^2 \begin{pmatrix} u_1(X,Y,t)\left[\lambda_N/F_{11}^2(1-\ln(F_{11}F_{22})) + \mu(1+1/F_{11}^2) - E\right] \\ u_2(X,Y,t)\left[\lambda_N/F_{22}^2(1-\ln(F_{11}F_{22})) + \mu(1+1/F_{22}^2) - E\right] \end{pmatrix}. \tag{6.9}$$

Now the initial boundary value problem defined by Eqn. 6.7 and 6.9 together with the zero boundary conditions can be solved numerically. The resulting fields, e.g. displacement, velocity, stress etc. can be compared with the presented analytical expressions. In the following, we will compare in particular the velocity field as the velocities are the prime variables in our formulation. We show the resulting velocity for the assumed displacement field in Eqn. 6.6 and the amplitude $B = 0.2$ at different time instants. In Fig. 6.13 the solution for the velocities derived by MMS is shown. We will use this solution for verification and comparison in the following.

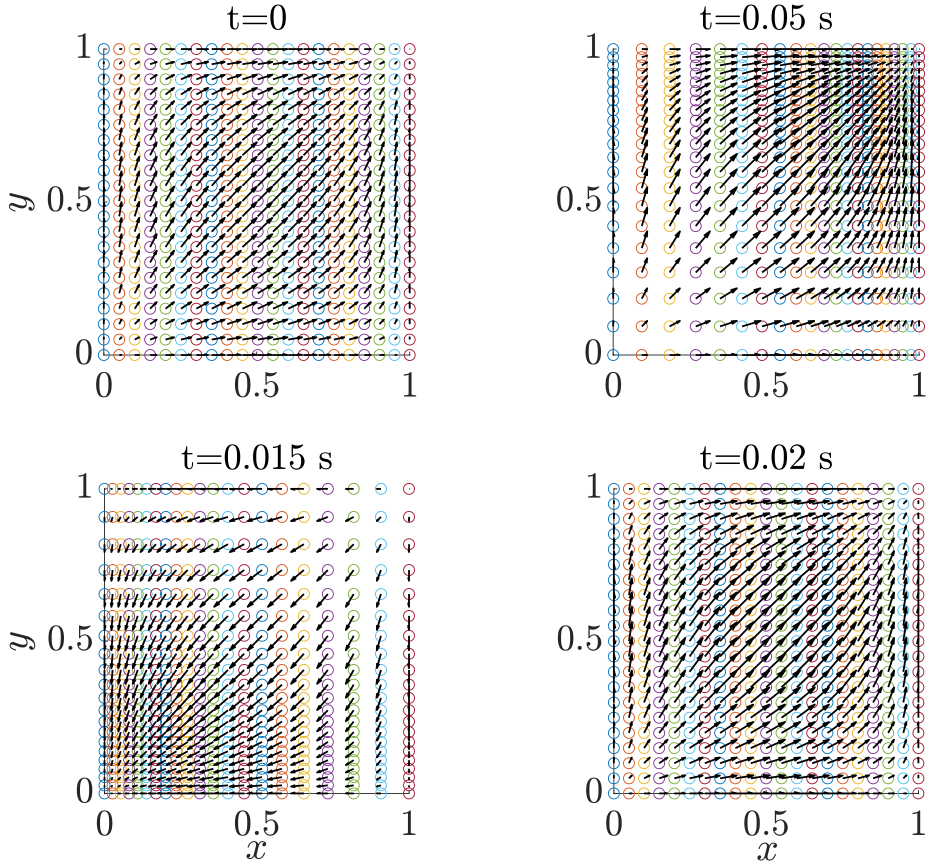


Figure 6.13: The velocity field (black vectors) obtained by MMS for the amplitude $B = 0.2$ at different time instants.

6.4.1 Rate of Convergence for MMS

We use the MMS to verify and compare the presented methods. The iMPM introduced by Sulsky and Gong (2016) is using the MLS approximation, not for all field variables. MLS approximation with cubic B-Splines is applied to approximate nodal velocities from MP. MLS with quadratic B-Splines is applied to approximate MP densities and stresses in the element's centre. The remaining approximations are performed with linear B-Splines as in the Standard MPM. In the following, we also apply the MLS approximation with different polynomial bases and weighting functions to nodal velocities. The rest of the variables are approximated with linear B-Splines and polynomial basis $\mathbf{p}(\mathbf{x}) = (1)$, if not stated differently. The adaptive regularization is applied in those cases where an initial $\lambda-$value is given. The compact supports of the one-dimensional weighting functions are, $2h$ for linear, $3h$ for quadratic, $4h$ for cubic B-Splines and $2.6h$ for the Wendland function. To speed up the calculation, not too small λ was chosen. We consider two cases with different amplitudes B for the MMS. The first case represents small deformations, while the second case reproduces large deformations (Wobbes *et al.*, 2019). In both cases one MP per element was used, which initially was placed in the element's centre. The total time of the simulation is $T = 0.02\,\mathrm{s}$, which corresponds to one period of the solution in Eqn. 6.6. The time increment for the forward Euler integration is chosen as $\Delta t = 0.00005\,\mathrm{s}$. The following parameter for the MMS were used:

ρ	1000 kg/m^3
E	10^7 Pa
ν	0.3
λ_N	$E\nu/[(1+\nu)(1-2\nu)]$
μ	$E/[2(1+\nu)]$
B	0.0001(1st case) and 0.05(2nd case) .

Table 6.1: Material properties and amplitudes for MMS

For the comparison we consider the MP velocities at their current location. The error between the calculated and MMS solution is obtained as the time-averaged discrete L_2-norm,

$$error = \frac{1}{T_N}\sum_{n=1}^{T_N}\left(\frac{1}{n_p}\sum_{p=1}^{n_p}\|\mathbf{v}_p^n - \mathbf{v}_{p,MMS}^n\|_2\right) . \tag{6.10}$$

In Eqn. 6.10 the total number of time increments is T_N, the number of MP is n_p and MP and MMS velocities at $(x_p\ y_p)$ for time step n are denoted by $\mathbf{v}_p^n$ and $\mathbf{v}_{p,MMS}^n$.

In Fig. 6.14 the comparison of different methods for the case of $B = 0.0001$ is shown. The error is plotted logarithmically against the element size h. Therefore, the rate of convergence is directly related to the slope between the x- and y-axis. The application of Wendland weighting function results in a convergence rate slightly over 2. The Standard MPM as well as the two cases of linear polynomial bases gives a similar convergence rate of approximately 1.5 . For the cases using $\mathbf{p}(\mathbf{x}) = (1\, x\, y)^T$, a solution can only be obtained by the pseudo inverse or regularization, as discretization with one MP per element violates the need of at least three MP at the boundary to obtain an invertible moment matrix. It should be noted that by additional approximation modifications, usage of more MP per element and Gauss quadrature smaller errors but with similar convergence rate can be obtained with MLS (Sulsky and Gong, 2016; Wobbes *et al.*, 2019).

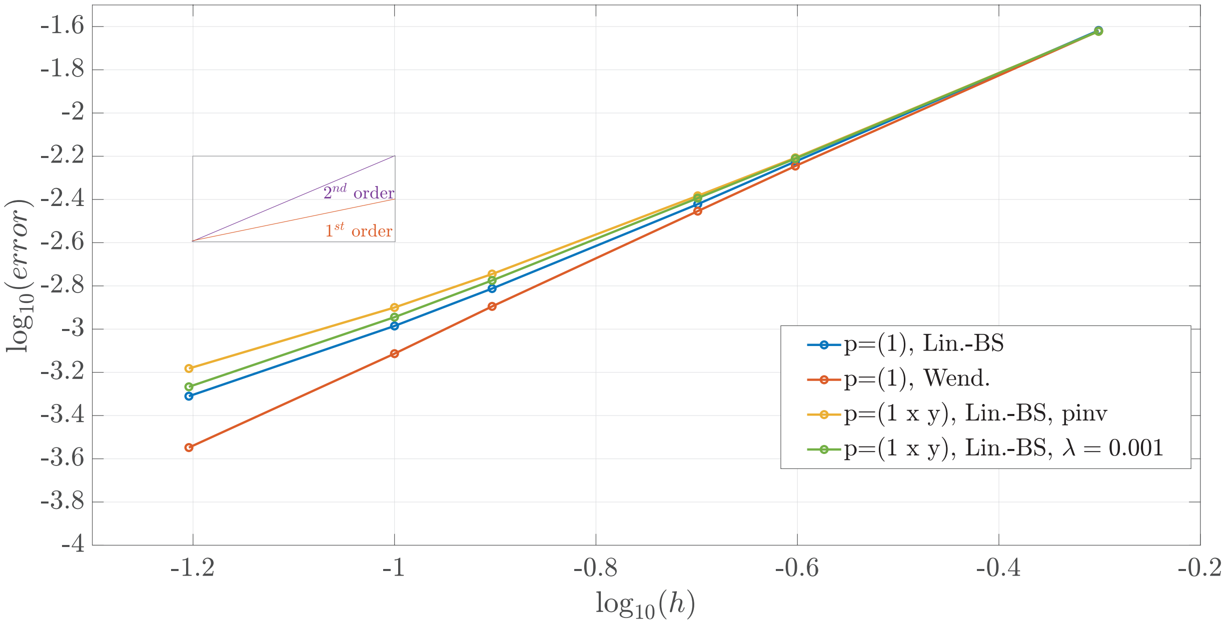


Figure 6.14: Comparison of MMS and numerical solutions for the case $B = 0.0001$. **blue**: Standard MPM; **red**: Wendland weighting function for nodal velocities, $\mathbf{p}(\mathbf{x}) = (1)$; **yellow**: $\mathbf{p}(\mathbf{x}) = (1\, x\, y)^T$ with linear B-Splines for nodal velocities solved with *pinv*; **green**: $\mathbf{p}(\mathbf{x}) = (1\, x\, y)^T$ with linear B-Splines for nodal velocities solved with initial $\lambda = 0.001$.

In Fig. 6.15 the comparison of different methods for the case of $B = 0.05$ is shown. The large amplitude B leads to divergence of the Standard MPM. The reason to this is the loss of contact issue explained in Sec. 5.2. The large deformation leads to empty elements between MP in particular when the element size decreases.Therefore divergence appears with decreasing element size as also mentioned in Sulsky and Gong (2016). In Fig. 6.16 the loss of contact for the Standard MPM case is shown. We chose an element size of $h = 0.0625$, which corresponds to 256 elements for the unit square shown in Fig. 6.13 and used one MP per element. The resulting crack divides the square in four fragments, which due to the elastic material model, tend to return to their corners. The combination of linear B-Splines with the polynomial basis $\mathbf{p}(\mathbf{x}) = (1\, x\, y)^T$ leads to an earlier divergence, as the moment matrix becomes ill-conditioned. The application of the pseudo inverse also

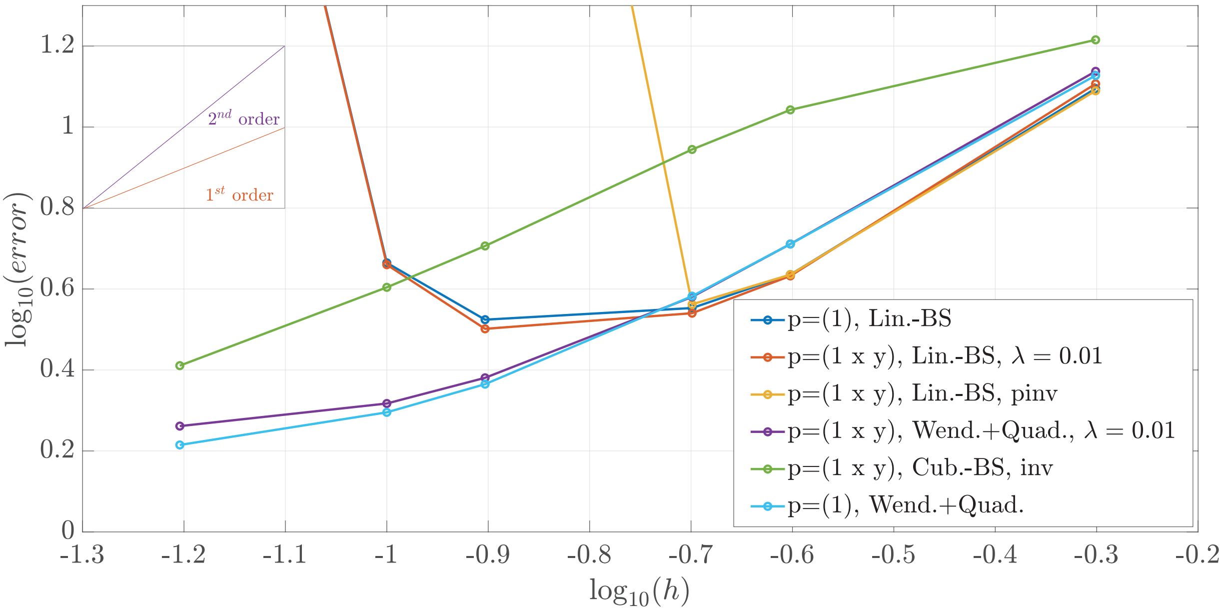


Figure 6.15: Comparison of MMS and numerical solutions for the case $B = 0.05$. **blue**: Standard MPM; **red**: $\mathbf{p}(\mathbf{x}) = (1\,x\,y)^T$ with linear B-Splines for nodal velocities solved with initial $\lambda = 0.01$; **yellow**: $\mathbf{p}(\mathbf{x}) = (1\,x\,y)^T$ with linear B-Splines for nodal velocities solved with *pinv*; **purple**: Wendland weighting functions with $\mathbf{p}(\mathbf{x}) = (1\,x\,y)^T$ for nodal velocities + quadratic B-Spline with $\mathbf{p}(\mathbf{x}) = (1)$ for all other variables solved with initial $\lambda = 0.01$; **green**: Cubic B-Spline with $\mathbf{p}(\mathbf{x}) = (1\,x\,y)^T$ for nodal velocities + cubic B-Spline with $\mathbf{p}(\mathbf{x}) = (1)$ for all other variables solved with *inv*; **light blue**: Wendland weighting functions with $\mathbf{p}(\mathbf{x}) = (1)$ for nodal velocities + quadratic B-Spline with $\mathbf{p}(\mathbf{x}) = (1)$ for all other variable.

leads to divergence. Regularization can fix this issue and delay the divergence to a similar time as for the Standard MPM. Only the application of weighting functions with larger compact support maintains convergence. The smallest error was achieved by combing Wendland weighting functions for nodal velocities and quadratic B-Splines for the remaining variables. Nevertheless for the constant $\mathbf{p}(\mathbf{x}) = (1)$ and linear basis $\mathbf{p}(\mathbf{x}) = (1\,x\,y)^T$, the convergence rate dropped below one for decreasing element size. The combination of cubic B-Splines with $\mathbf{p}(\mathbf{x}) = (1\,x\,y)^T$ for nodal velocities and $\mathbf{p}(\mathbf{x}) = (1)$ for the rest, maintains a convergence rate of approx. one even with smaller elements. The large compact support of cubic B-Splines involves enough MP in the MLS approximation of nodal values, that the moment matrix is invertible all the time. Therefore, no pseudo inverse or regularization is required and the standard inverse can be used. The large compact support of cubic B-Splines implies higher computational costs, as more MP are involved in the calculation. The compact support of the one-dimensiponal Wendland weighting function was chosen to be $h = 2.6$ in all calculations. It is possible to increase it without violating the partition of unity property as mentioned in Sec. 6.2. The presented results of the MMS analysis confirmed the findings for rate of convergence of other authors (Sulsky and Gong, 2016; Wobbes *et al.*, 2019). The introduced regularization and Wendland weighting functions improved the robustness and accuracy of existing methods for small and large deformations.

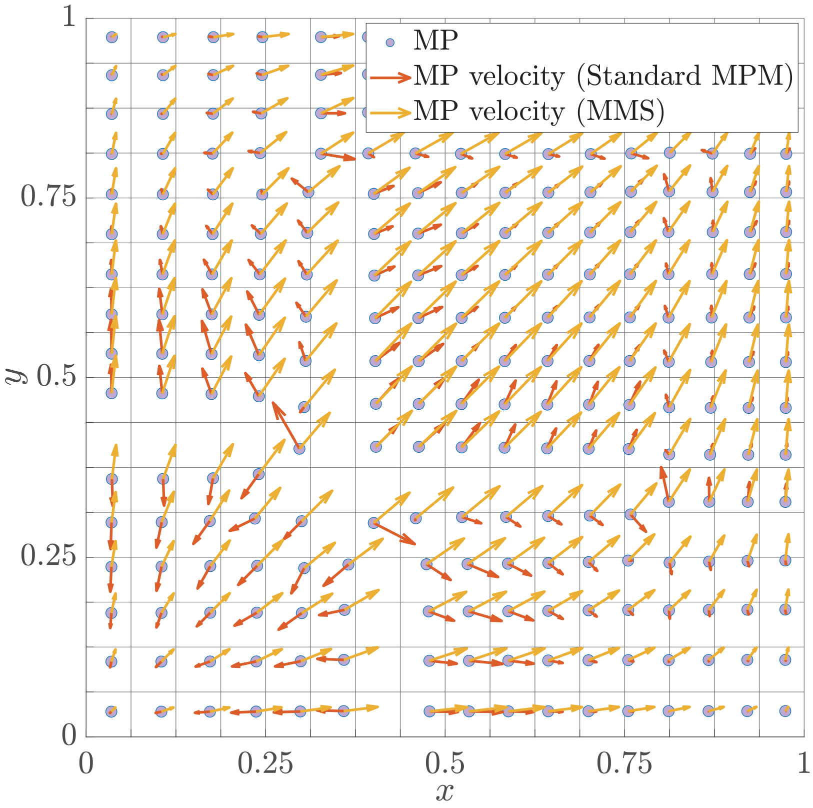


Figure 6.16: Standard MPM simulation for $B = 0.05$ and the loss contact of MP. The resulting crack leads to an elastic behaviour of each fragment. The calculated MP velocities (red) and the analytical MMS solution are partly pointing in different directions.

Chapter 7

MPM for Coupled Problems

In this chapter the MPM method is extended to two phases. The derived equations from Ch. 4 are used for this purpose. The Double-Point MPM (Więckowski, 2013) is used to discretize the equations on two different MP sets. Finally, the results obtained are verified by means of simple examples.

7.1 Singe- and Double-Point MPM

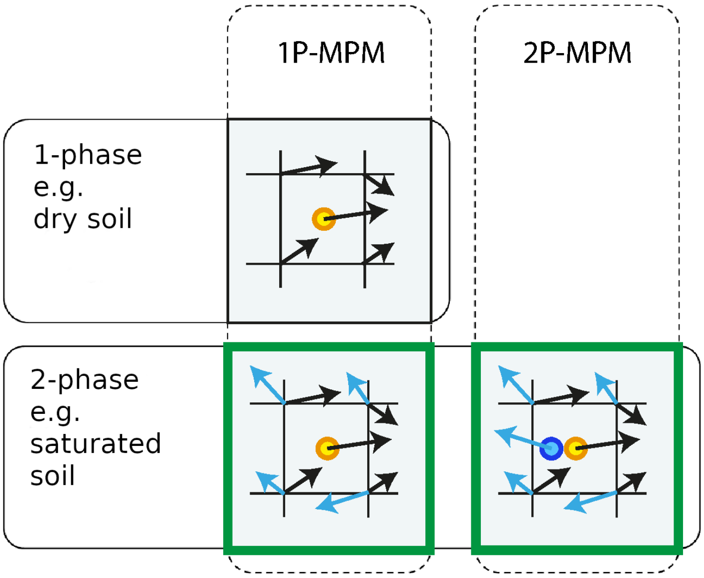


Figure 7.1: Different MPM discretization approaches: In the 1-phase single point MPM (1P-MPM), the nodal velocities of the solid skeleton are used to calculate the new MP position. The 2-phase 1P-MPM is the same, although additional velocities of the fluid phase are calculated. For the 2-phase double point (2P-MPM), two different MP types are used so that for one, a new position can be calculated based on the fluid velocities and for the other based on the solid skeleton velocities.

In Więckowski (2013) two different possibilities of MP discretization were presented to solve Eqn. 4.37a and 4.37b. Two velocity fields must be calculated for the coupled equations, one for the soil and one for the pore water. It was suggested that the calculated nodal velocities could be transferred to either one or two sets of MP. Fig. 7.1 shows both possibilities in addition to the usual one-phase MPM. In the single point MPM (1P-MPM) for two phases, both velocity fields are transferred to one set of MP, resulting in an ambiguous calculation of the MP displacement. Therefore, the 1P-MPM uses only the solid velocity to calculate the new MP position. This method is only acceptable for applications where both velocity fields have a similar direction and amplitude.

Weak formulation We derive the weak form of Eqn. 4.39a and 4.39b with gravity as the only body force, i.e. $\mathbf{b} = (0\,,-g)^T = (0\,,-9.81\mathrm{m/s^2})^T$ in two dimensions. Therefore we multiply both equations by the vector-valued test function $\delta\boldsymbol{u}$ and integrate over Ω^s and Ω^f with $\Omega = \Omega^s \cup \Omega^f$. The domains Ω^s and Ω^f are occupied by soil and the pore fluid, respectively. It should be noted that in general the test function $\delta\boldsymbol{u}$ might be different for each equation. Furthermore, we apply the divergence theorem and integration by parts to shift the spatial derivatives to the test function and obtain the weak form as

$$\begin{aligned}\int_{\Omega^s}\left[n_s\rho_s\frac{\mathrm{D}^s\mathbf{v}_s}{\mathrm{D}t}\cdot\delta\boldsymbol{u}\right]\mathrm{d}\Omega &= \int_{\Omega^s}\left[(-\boldsymbol{\sigma}' + n_s p_f\mathbf{I}) : \nabla\delta\boldsymbol{u}\right]\mathrm{d}\Omega \\ &+ \int_{\Omega^s}\left[n_s\rho_s\mathbf{b}\cdot\delta\boldsymbol{u}\right]\mathrm{d}\Omega - \int_{\Omega^s\cap\Omega^f}\left[\overline{\mathbf{f}}_d\cdot\delta\boldsymbol{u}\right]\mathrm{d}\Omega \\ &+ \int_{\partial\Omega^s}\left[(\boldsymbol{\sigma}' - n_s p_f\mathbf{I})\,\mathbf{n}\cdot\delta\boldsymbol{u}\right]\mathrm{d}S\,,\end{aligned} \tag{7.1a}$$

$$\begin{aligned}\int_{\Omega^f}\left[n_f\rho_f\frac{\mathrm{D}^f\mathbf{v}_f}{\mathrm{D}t}\cdot\delta\boldsymbol{u}\right]\mathrm{d}\Omega &= \int_{\Omega^f}\left[n_f p_f\mathbf{I} : \nabla\delta\boldsymbol{u}\right]\mathrm{d}\Omega \\ &+ \int_{\Omega^f}\left[n_f\rho_f\mathbf{b}\cdot\delta\boldsymbol{u}\right]\mathrm{d}\Omega + \int_{\Omega^s\cap\Omega^f}\left[\overline{\mathbf{f}}_d\cdot\delta\boldsymbol{u}\right]\mathrm{d}\Omega \\ &+ \int_{\partial\Omega^f}\left[(-n_f p_f\mathbf{I})\,\mathbf{n}\cdot\delta\boldsymbol{u}\right]\mathrm{d}S\,.\end{aligned} \tag{7.1b}$$

Following the Standard MPM approach as in Więckowski (2013) and Bandara (2013) Eqn. 7.1a and 7.1b can be written for the nodal acceleration vectors at time t as

$$\left(\mathbf{M}_s^{lump}\right)_i^t\left(\hat{\mathbf{a}}_s\right)_i^t = \left(\mathbf{f}_s^{ext}\right)_i^t - \left(\mathbf{f}_s^{int}\right)_i^t - \left(\mathbf{F}_d\right)_i^t = \left(\mathbf{f}_s\right)_i^t\,, \tag{7.2a}$$

$$\left(\mathbf{M}_f^{lump}\right)_i^t\left(\hat{\mathbf{a}}_f\right)_i^t = \left(\mathbf{f}_f^{ext}\right)_i^t - \left(\mathbf{f}_f^{int}\right)_i^t + \left(\mathbf{F}_d\right)_i^t = \left(\mathbf{f}_f\right)_i^t\,, \tag{7.2b}$$

with the nodal masses and forces

$$(\mathbf{M}_s^{lump})_i = \sum_{sp=1}^{N_{sp}} m_{sp} N_i(\mathbf{x}_{sp}) \;\;,\;\; (\mathbf{M}_f^{lump})_i = \sum_{fp=1}^{N_{fp}} m_{fp} N_i(\mathbf{x}_{fp})\,,$$

$$(\mathbf{f}_s^{ext})_i = \sum_{sp=1}^{N_{sp}} m_{sp} N_i(\mathbf{x}_{sp})\mathbf{b}(\mathbf{x}_{sp}) + \int_{\partial\Omega_s} N_i(\mathbf{x})\,(\boldsymbol{\sigma}' - n_s p_f \mathbf{I})\,\mathbf{n}\,\mathrm{d}S\,,$$

$$(\mathbf{f}_f^{ext})_i = \sum_{fp=1}^{N_{fp}} m_{fp} N_i(\mathbf{x}_{fp})\mathbf{b}(\mathbf{x}_{fp}) + \int_{\partial\Omega_f} N_i(\mathbf{x})(-n_f p_f \mathbf{I})\mathbf{n}\,\mathrm{d}S\,,$$

$$(\mathbf{f}_s^{int})_i = \sum_{sp=1}^{N_{sp}} V_{sp}\boldsymbol{\sigma}'_{sp}\,\mathbf{grad}(N_i)(\mathbf{x}_{sp}) - \sum_{fp=1}^{N_{fp}} (1-n_{fp})p_{fp}V_{fp}\,\mathbf{grad}(N_i)(\mathbf{x}_{fp})\,,$$

$$(\mathbf{f}_f^{int})_i = -\sum_{fp=1}^{N_{fp}} n_{fp}V_{fp}p_f\mathbf{I}\;\mathbf{grad}(N_i)(\mathbf{x}_{fp})\,,$$

$$(\mathbf{F}_d)_i = -\sum_{j=1}^{N_n}\sum_{fp=1}^{N_{fp}} \frac{m_{fp}n_{fp}g}{\kappa_{fp}} N_i(\mathbf{x}_{fp})N_j(\mathbf{x}_{fp})(\mathbf{v}_{fj} - \mathbf{v}_{sj})\,.$$

In the two-dimensional case, the entries of Eqn. 7.2a and 7.2b have always two components for each node i corresponding to the x- and y-direction, e.g. $\mathbf{v}_{fi} = (v_{fi}^x\;,\;v_{fi}^y)^T$. The indices sp and fp refer to solid and fluid MP, where N_{sp} and N_{fp} is the total number of the corresponding MP. The total number of nodes is denoted by N_n. The vector $\mathbf{n}$ is the normal unit vector on the corresponding surface and $\mathbf{I}$ is the identity tensor, which is represented by a $2 \times 2-$matrix in the considered case. The Drag-Force $(\mathbf{F}_d)_i$ is only non-zero for nodes, where both fluid and solid MP have a non-zero contribution. Conversely, if there are only fluid or only solid MP in the vicinity of node i, the resulting Drag-Force is set to zero. For simplicity only the first summand (Dracy flow) in Eqn. 4.40 is considered. The discretization of the whole Eqn. 4.40 can be found in Więckowski (2013). Eqn. 7.2a and 7.2b is integrated in time using the explicit forward Euler scheme to obtain the updated nodal momentum as

$$(\hat{\mathbf{mv}})_s^{t+\Delta t} = (\hat{\mathbf{mv}})_s^t + \Delta t\mathbf{f}_s^t\,, \tag{7.4a}$$

$$(\hat{\mathbf{mv}})_f^{t+\Delta t} = (\hat{\mathbf{mv}})_f^t + \Delta t\mathbf{f}_f^t\,. \tag{7.4b}$$

The updated nodal momenta $(\hat{\mathbf{mv}})_s^{t+\Delta t}$ and $(\hat{\mathbf{mv}})_f^{t+\Delta t}$ in Eqn. 7.4a and 7.4b are used to update the nodal velocities and positions of MP analogue to the steps described in Tab. 5.1. Division of $(\hat{\mathbf{mv}})_s^{t+\Delta t}$ by the nodal mass provides the nodal velocities $\hat{\mathbf{v}}_s^{t+\Delta t}$ and $\hat{\mathbf{v}}_f^{t+\Delta t}$, which are used in the calculation of pore pressures. The pore pressure is calculated by explicit time-integration of Eqn. 4.34 as

$$
\begin{aligned}
p_f^{t+\Delta t} &= p_f^t \\
&- \frac{\Delta t}{n_f C_f}\Big(\mathbf{grad}\,(n_{fp}) \cdot (\mathbf{v}_f - \mathbf{v}_s) + (1 - n_f)\,\mathbf{div}\,(\mathbf{v}_s) + n_f \mathbf{div}\,(\mathbf{v}_f)\Big)\,.
\end{aligned}
\tag{7.5}
$$

The divergence operator in Eqn. 7.5 at the $p-$th MP is calculated as

$$
\mathbf{div}(\mathbf{v}_p) \approx \sum_{i=1}^{N_n} \mathbf{grad}(N_i)(\mathbf{x}_p) \cdot \mathbf{v}_i\,. \tag{7.6}
$$

A detailed description of the calculation algorithm is given in Appx. C or can be found in Bandara (2013) and Więckowski (2013).

Updated porosity Eqn. 7.2a and 7.2b contain terms with porosities n_{sp} and n_{fp} evaluated at the corresponding MP. In the 2P-MPM as introduced by Więckowski (2013), Bandara (2013), Abe *et al.* (2014), and Martinelli and Rohe (2015) the change of porosity is calculated using the discretized model at the MP. Therefore, the solid MP are assumed to be compressible and the volume change of V_{sp} is calculated analogue to Eqn. 4.10 as $V_{sp}^{t+\Delta t} = V_{sp}^t J_t^{t+\Delta t}$. As mentioned in Sec. 4.2.1, this contradicts the assumption of incompressible soil grains. In addition, it contradicts the incompressibility assumptions made in Sec. 4.2.2. In the 2P-MPM, these violations are accepted in order to simulate large deformations, erosion and sedimentation. To the author's knowledge, a continuum formulation that contains all the mentioned features without violations has not yet been published. The volume change of a solid MP leads to a change of the porosity n_s, which is used to calculate n_f using the saturation condition $n_s + n_f = 1$. Based on these values, new states of the mixture can be introduced. For example in the case of increasing pore space and vanishing effective stresses, fluidization of solid MP is identified as

$$
\sigma'_{kk} \to 0^- \quad \text{and} \quad \frac{\partial(1-n_s)}{\partial t} > 0 \quad \text{with} \quad \sigma'_{kk} = \sigma'_{xx} + \sigma'_{yy} + \sigma'_{zz}\,. \tag{7.7}
$$

For the sedimentation process, the time derivative of the porosity i.e. $\partial(1-n_s)/\partial t < 0$ and the volumetric stain rate $\dot{\boldsymbol{\epsilon}}_v$ must be negative. The contact of soil grains is identified if the porosity falls below a defined limit value n_{min}. For more details the reader is referred to Więckowski (2013), Bandara (2013), Abe *et al.* (2014), and Martinelli and Rohe (2015).

7.2 Local Damping

In order to use the dynamic formulation in Eqn. 7.2a and 7.2b also for quasi-static and static calculations, the contribution of the inertial forces must be minimised. For this purpose, the so-called local damping is used, which was presented in Al-Kafaji (2013) for a slightly different coupled formulation than the one used here. Therefore, an adapted approach is presented below. The application of local damping leads to modified Eqn. 7.2a and 7.2b, which can be expressed as

$$\mathbf{M}_s^{lump}\hat{\mathbf{a}}_s = \mathbf{f}_s^{ext} - \mathbf{f}_s^{int} - \mathbf{F}_d - L_D \cdot \mathbf{sign}(\hat{\mathbf{v}}_s) \cdot |\mathbf{f}_s^{ext} - \mathbf{f}_s^{int}| \, , \tag{7.8a}$$

$$\mathbf{M}_f^{lump}\hat{\mathbf{a}}_f = \mathbf{f}_f^{ext} - \mathbf{f}_f^{int} + \mathbf{F}_d - L_D \cdot \mathbf{sign}(\hat{\mathbf{v}}_f) \cdot |\mathbf{f}_f^{ext} - \mathbf{f}_f^{int}| \, , \tag{7.8b}$$

where L_D is the local damping coefficient. The sign function in Eqn. 7.8a and 7.8b causes the damping forces to act in the opposite direction of the nodal velocities. The results in the following sections simulating gravity loading and consolidation were obtained with $L_D = 0.95$.

7.3 Gravity Loading

The dynamic formulation from Eqn. 7.2a and 7.2b leads to solutions including wave propagation when a load is applied. For most calculations, however, it is first necessary to establish a static or quasi-static equilibrium, which for example describes the hydrostatic pressure in the pore water and static earth pressure. In order to obtain a static or quasi-static solution, the influence of the inertia terms must be reduced. Therefore, we apply the damped formulation showed in Eqn. 7.2a and 7.2b.

In the following, we consider two cases for the initialisation of gravity. In the first case, a 2 m water column with a submerged 1 m soil column is initialised. In the second example, a 2 m soil column and a 1m pore water column are considered. Linear B-Splines as weighting functions, square elements with an edge length of 0.02m, one solid and one fluid MP per element were used for the calculation. To check the accuracy of the results, the following formulas for saturated soil can be used,

$$\begin{aligned} &\text{vertical total stress:} && [(1-n)\rho_s + n\rho_w]\, gz \\ &\text{hydro-static pressure:} && \rho_w gz \\ &\text{vertical effective stress:} && [(1-n)\rho_s + n\rho_w - \rho_w]\, gz \, , \end{aligned} \tag{7.9}$$

where z denotes the height of the considered column. The densities and porosity were chosen as follows

ρ_s	2650 kg/m^3
ρ_f	1000 kg/m^3
n_f	0.4
g	10 m/s^2 .

Table 7.1: Material properties used for gravity loading

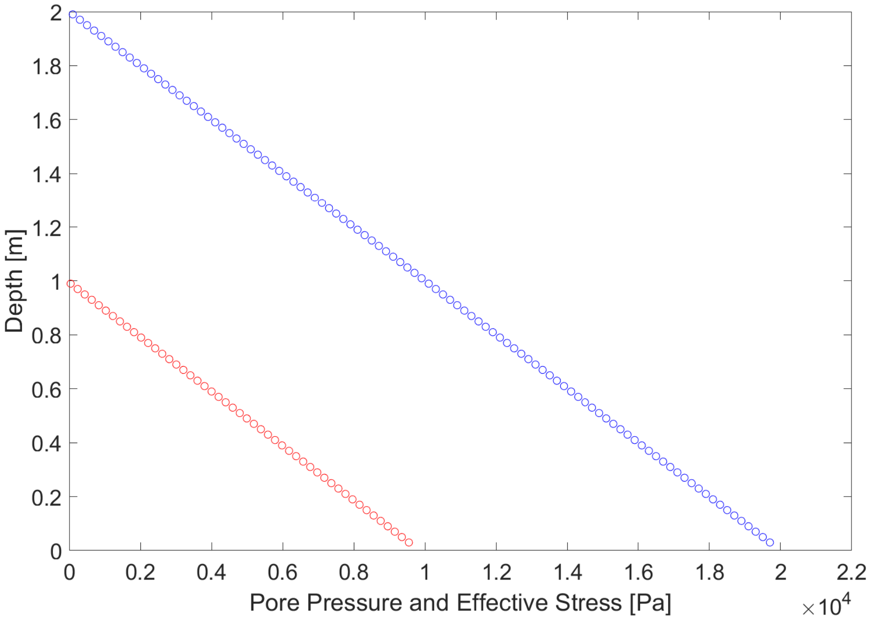


Figure 7.2: Gravity initialization (submerged): Pore pressure at fluid MP(blue circles) and vertical effective stresses at solid MP(red circles). Eqn. 7.9 results in $p(z=0) = 20000\,\text{Pa}$ and $\sigma'(z=0) = 9900\,\text{Pa}$.

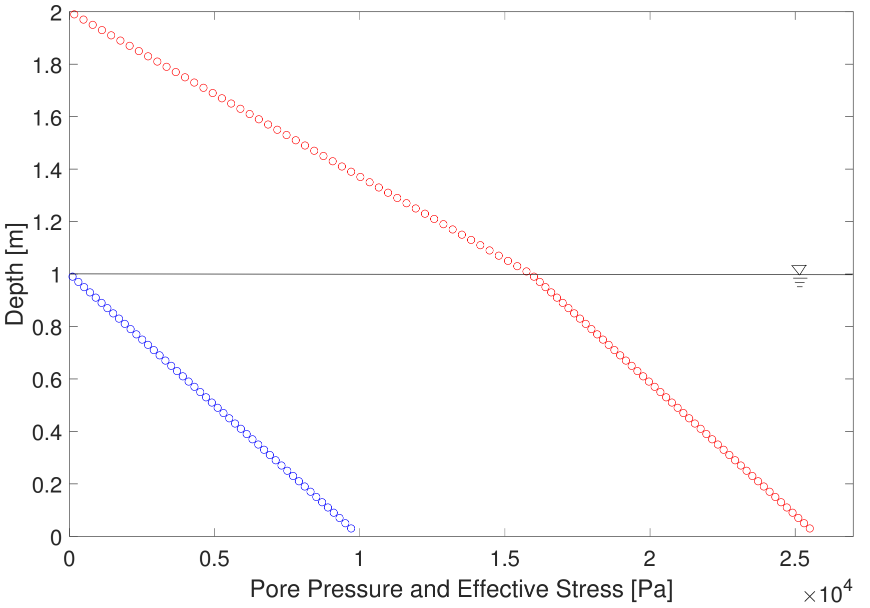


Figure 7.3: Gravity initialization (saturated): Pore pressure at fluid MP(blue circles) and vertical effective stresses at solid MP(red circles). Eqn. 7.9 gives $p(z=0) = 10000\,\text{Pa}$ and $\sigma'(z=0) = 25800\,\text{Pa}$.

The shown results in Fig. 7.2 and 7.3 are in good agreement with the analytical solutions.

7.4 Consolidation

While the body forces are mainly decisive in the initialisation of gravity, the Drag-Force is of great importance in the consolidation, as it describes the flow of the pore fluid through the pore space. Again, local damping must be used to simulate the quasi-static process using our dynamic formulation. A linear-elastic material behaviour is assumed for the soil as well as a slightly compressible fluid. The additional material properties to those in Tab. 7.1 are shown in Tab. 7.2. Linear B-Splines as weighting functions, square elements with an edge length of 0.05 m, one solid and one fluid MP per element were used for the calculation. A vertical effective stress of −1 Pa and zero pore pressure are applied at the top of the column as boundary conditions. The analytical solution of Terzaghi's one dimensional consolidation theory (Craig, 2004) is used to compare the numerical results,

E	10^7 Pa
ν	0
k	10^{-3} m/s
$K_w = C_f^{-1}$	$2 \cdot 10^9$ Pa

Table 7.2: Material properties used for consolidation

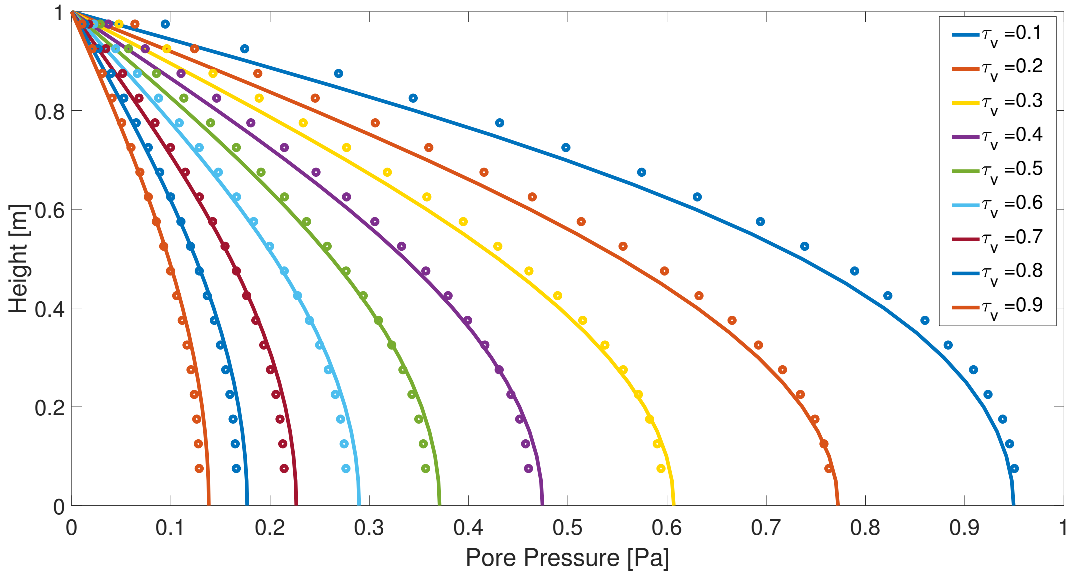


Figure 7.4: Comparison of numerical(circles) and analytical(lines) solutions of pore pressure at fluid MP for different time factor τ_v.

$$p\left(z\right) = \sum_{i=1}^{\infty} \frac{p_0}{\pi\left(2i+1\right)} \cdot \sin\left(\frac{\pi\left(2i+1\right)}{2 \cdot H}\right) \cdot \exp\left(-\left[\pi\left(m+\frac{1}{2}\right)\right]^2 \cdot \tau_v\right) . \qquad (7.10)$$

In Eqn. 7.10 the coefficient of consolidation c_v and the time factor τ_v are defined as

$$c_v = \frac{k \cdot E}{g \cdot \rho_w} \quad \text{and} \quad \tau_v = \frac{C_v \cdot t}{H^2} . \qquad (7.11)$$

The infinite sum in Eqn. 7.10 is calculated as a finite sum with 100 summands. Fig. 7.4 shows the comparison between numerical and analytical solutions for a height of 1m. The results are in good agreement, in particular for larger values of τ_v.

7.5 Bulk Viscosity

Propagating discontinuities can be caused by a suddenly applied boundary condition, discontinuous initial conditions or a non-linear material behaviour. This, so called shock propagation causes instabilities in form of oscillations in front and behind the discontinuity. Richtyer (1950) and Landshoff (1955) proposed a method to overcome these oscillations by adding artificial viscosity in case of compression, i.e. if the trace of the strain rate for soil is negative, $\mathrm{tr}(\dot{\boldsymbol{\varepsilon}}_{\boldsymbol{s}}) < 0$. In Zhang *et al.* (2011) this method was applied in the framework of MPM. In Chmelnizkij *et al.* (2019) it was adopted for the 2P-MPM as

$$q = \begin{cases} \rho_s \left(c_2 h \cdot \mathrm{tr}(\dot{\boldsymbol{\varepsilon}}_{\boldsymbol{s}})\right)^2 - c_1 \rho_s h c \, \mathrm{tr}(\dot{\boldsymbol{\varepsilon}}_{\boldsymbol{s}}) & \text{if } \mathrm{tr}(\dot{\boldsymbol{\varepsilon}}_{\boldsymbol{s}}) < 0 \\ 0 & \text{otherwise} \end{cases} . \tag{7.12}$$

In Eqn. 7.12 the coefficients c_1 and c_2 define the amount of viscosity applied to the model. Too high values might lead too smooth results. Further h is the element size, c the wave propagation speed and ρ_s the soil density. The quantity q in Eqn. 7.12 is added to the effective stress tensor during the calculation of internal forces. The calculations in the following Sec. 7.6 were performed with the coefficients $c_1 = 0.42$ and $c_2 = 1.2$.

7.6 Wave propagation in saturated soil

While in the previous examples the influence of the inertial forces by means of damping was kept as small as possible, it is crucial to take them into account in the case of wave propagation. Therefore, in the following examples no local damping is used. We apply linear B-Splines as weighting functions, square elements with an edge length of 0.0001 m, one solid and one fluid MP per element for the calculations. We consider the following initial boundary value problem:

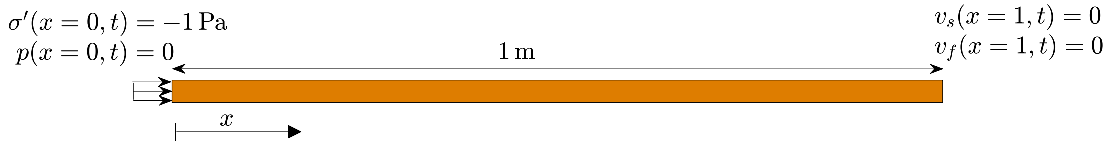


Figure 7.5: One-dimensional boundary value problem with natural boundary conditions on the left and essential on the right side.

As the boundary condition on the left side in Fig. 7.5 prescribes a non-zero effective stress, a discontinuity begins to propagate to the right side. The numerical solution leads to oscillations (Chmelnizkij *et al.*, 2019), which are minimised here by using bulk viscosity as mentioned in the previous section. For the simulation the following material properties were used

ρ_s	$2650\,\mathrm{kg/m^3}$
ρ_f	$1000\,\mathrm{kg/m^3}$
E	5 GPa
$K_w = C_f^{-1}$	2 GPa
ν	0
k	$10^{-3}\ \ \mathrm{m/s}\,.$

Table 7.3: Material properties used for wave propagation in saturated soil.

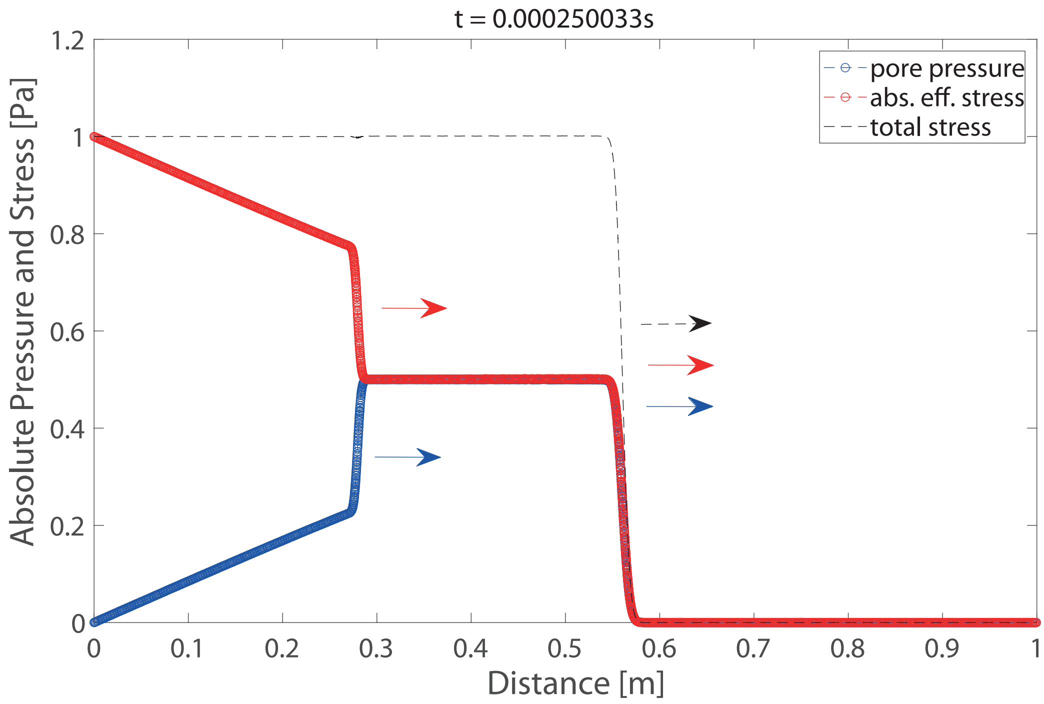


Figure 7.6: Pore pressure and absolute effective stress of the propagating undrained wave at 0.5605 m and drained wave at 0.2795 m.

The resulting pore pressure and effective stresses after 0.000250033 s are shown in Fig. 7.6. To verify the results, we consider Eqn. 4.58 and 4.57. The distribution of pore pressure and effective stress of the undrained wave from Eqn. 4.58 is 0.5, which is in very good agreement with the numerical results in Fig. 7.6. The propagating wave speeds from Eqn. 4.57 are $C_1 = 2241.7\mathrm{m/s}$ and $C_2 = 1118\,\mathrm{m/s}$, which results in travelling distances of 0.5605 m for the undrained and 0.2795 m fro the drained wave. These speeds agree well with the numerical results shown in Fig. 4.58. Biot (1956a) and Jong (1956) characterized the undrained wave as being in phase and the drained out of phase. While in the undrained wave, the particles move in the same direction with the same velocity. In the drained wave, the liquid and solid particles move in opposite directions with different velocities. This is also in agreement with the results in Fig. 7.6. While the soil exhibits a second compression through the faster moving solid MP to the right, the pore fluid undergoes a relaxation after compression by fluid MP moving to the left. To obtain results as accurate as shown in Fig. 7.6, a very fine discretization is required. Besides the small element size used for the calculation, a time increment of $\Delta t = 10^{-8}$ was chosen. The challenge is to achieve acceptable on sparse and scattered data.

7.7 Wave propagation on scattered data

The wave propagation in the previous section was calculated with one solid and one fluid MP per element located in the centre of each element. As the deformation in the wave propagation example is negligible, the MP barely change their position throughout the whole simulation. In case of simulations where large deformations take place, the positions of MP can become arbitrary. It is desirable to solve the wave propagation accurately for any arrangements of MP. Let us consider the case where liquid and solid MP are randomly arranged inside the elements. Therefore, we randomly vary the horizontal coordinate of MP within each element, while the vertical coordinate is held constant, which correspond to the one-dimensional case we consider here. We use a coarser grid compared to the previous example, with square elements and an edge length of 0.001 m.
Fig. 7.7 shows the random distribution of 100 solid and 100 liquid MP inside a 0.1 m column. We consider a similar problem as in the previous section and compare the results of different methods after $t = 4 \cdot 10^{-5}$ s. The corresponding travelling distances obtained by Eqn. 4.57 for both waves are $d_2 = 0.0897$ m and $d_1 = 0.0447$ m. The random arrangement of MP in Fig. 7.7 is chosen in such a way, that still each element contain one solid and one fluid MP. Therefore, each element is not empty and no gaps are present.

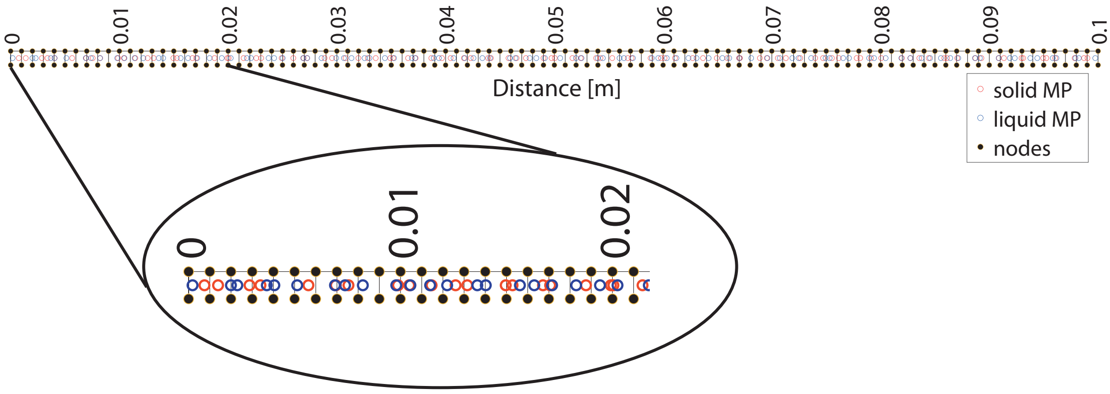


Figure 7.7: Discretization of a saturated column with randomly arranged liquid and solid MP.

In Fig. 7.8, the results for different weighting functions and constant polynomial basis $\mathbf{p}(\mathbf{x}) = (1)$ are shown. The arbitrary distribution of MP causes strong oscillations at the boundary in case the of linear B-Splines, which corresponds to the Standard MPM. In general, oscillations are present in all calculations leading to less accurate results, as shown in Fig. 7.6. The discontinuous shock fronts are smeared over several elements, partly due to the random arrangement of MP and larger elements. Additionally, weighting functions with larger compact supports lead to smeared shock fronts and slightly inaccurate wave propagation, as the information is transferred to more distant nodes in each increment. In the presence of empty elements, unwanted reflections occur, which the increase of compact supports can mitigate. Let us consider a structured distribution of MP but with a gap in the middle of the considered interval.

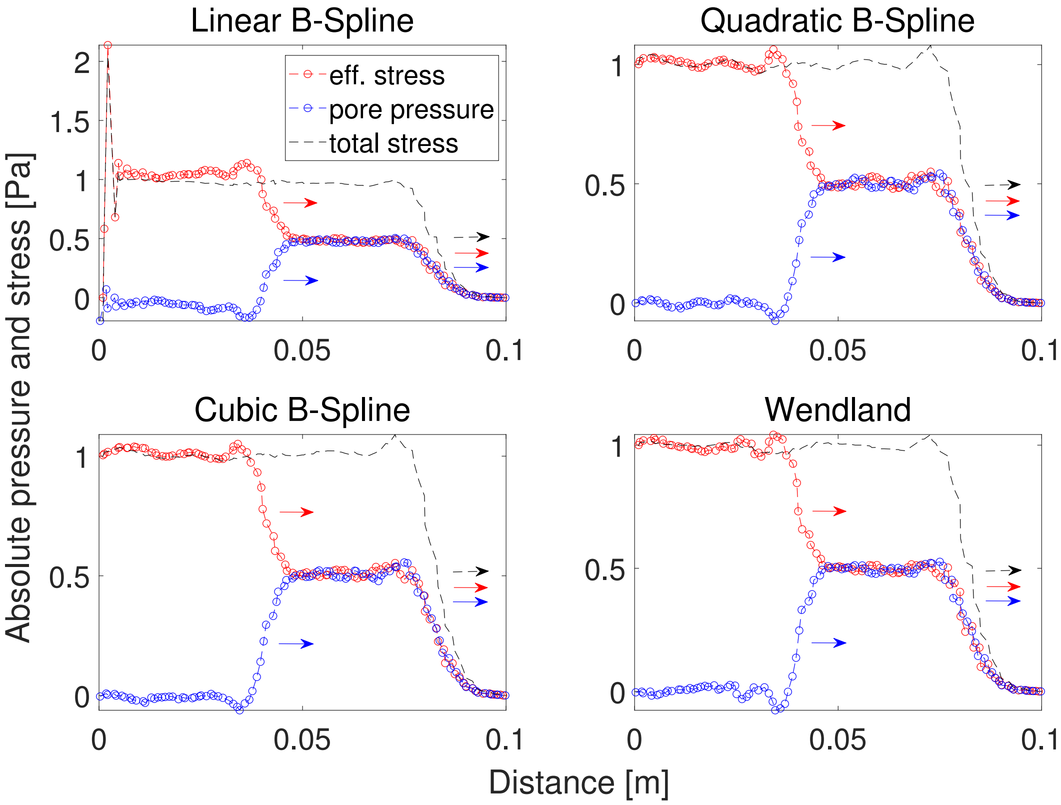


Figure 7.8: Comparison of different methods with $\mathbf{p}(\mathbf{x}) = (1)$ for a set of scattered MP (Linear: $2h$, Quadratic: $3h$, Cubic: $4h$, Wendland: $3.1h$).

As shown in Fig. 7.9, a gap of one empty element leads a full reflection of the propagating wave in the case of Standard MPM. The total stress in Fig. 7.9 indicates a reflection at a free boundary by changing from a compressional to a tensile wave. Therefore, in problems where large deformation and wave propagation are considered, unwanted reflections and interrupted wave propagation can occur if linear B-Splines are used. To overcome this issue, which is closely related to the loss of contact problem, we increase the compact support of the weighting functions. To mitigate the reflections significantly we propose to use a compact support of at least $10h$. In the following the application of the Wendland function with such a large compact support is shown.

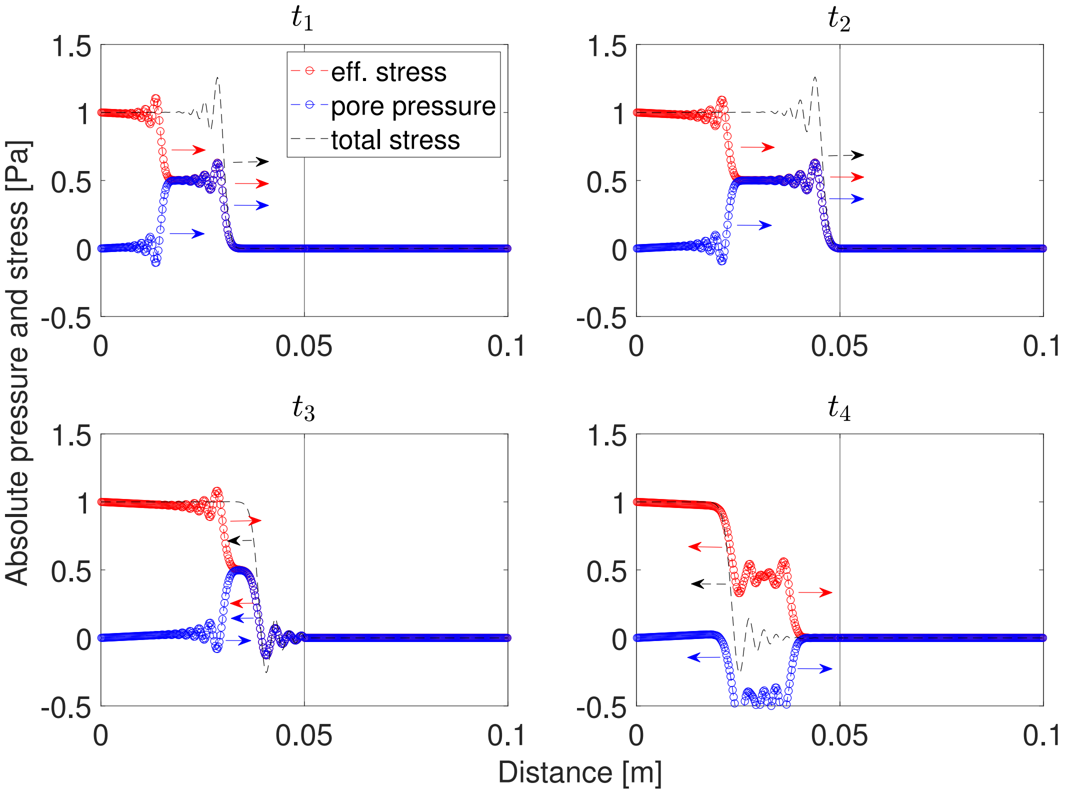


Figure 7.9: Wave propagation at different time instants, from t_1 to t_4, using linear B-Spline weighting functions. One empty element is placed in the middle at 0.05m. The arrows indicate the propagation direction of the corresponding shock front.

In Fig. 7.10 the results for the Wendland weighting function are shown. The MLS was applied to obtain the nodal masses, velocities and internal forces. Linear B-Splines were used to approximate the remaining variables. The propagating shock fronts cause oscillations in the vicinity of the empty element at 0.05 m when they surpass the gap. The reflections at 0.05 m are mitigated significantly and are small compared to the amplitudes of the propagating shock fronts. The amplitude of the total stress is reduced to approx. 0.77 Pa after surpassing the gap and is, therefore, lower than the expected value of 1 Pa. The propagating waves are reflected at the fixed boundary at 0.1 m. As expected, the total stress doubles its amplitude and travels backwards from right to left.
The application of the linear basis $\mathbf{p}(\mathbf{x}) = (1\, x\, y)^T$ leads to ill-conditioned moment matrices, as all MP are aligned horizontally forming a line. Therefore, we apply the regularization method discussed in Ch. 6 with the regularization parameter $\lambda = 0.001$.

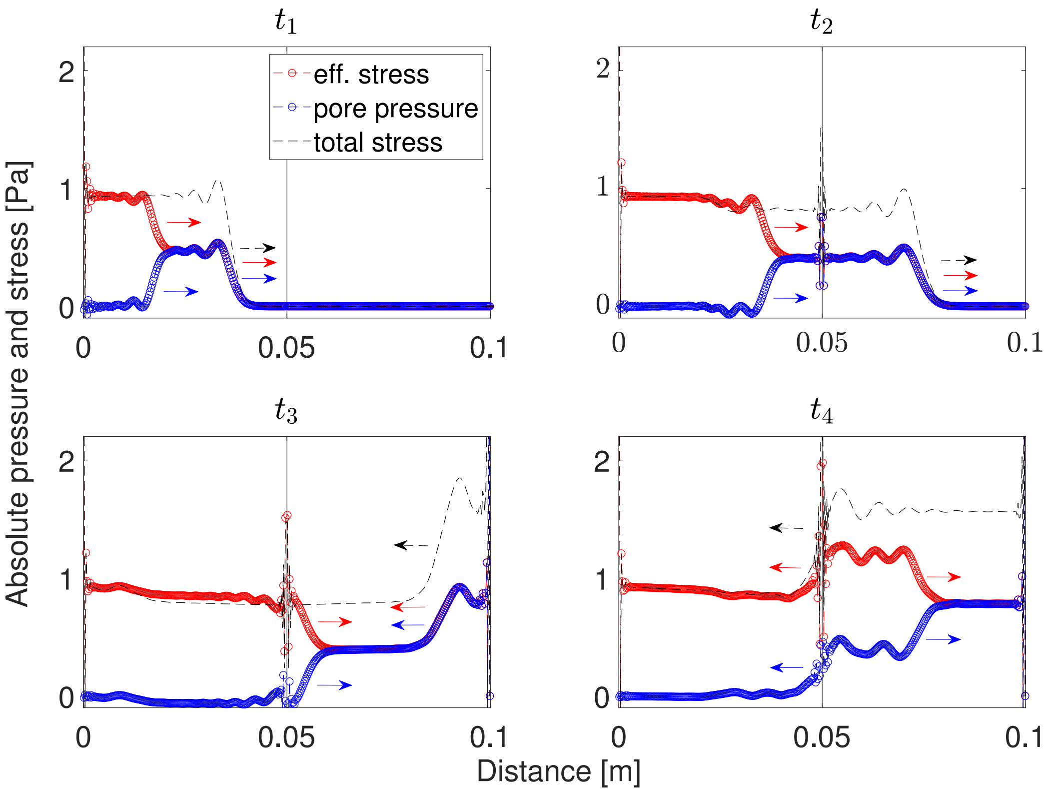


Figure 7.10: Wave propagation at different time instants, from t_1 to t_4, using the Wendland weighting functions with a compact support of $10h$ and the constant basis $\mathbf{p}(\mathbf{x}) = (1)$. One empty element is placed in the middle at 0.05m. The arrows indicate the propagation direction of the corresponding shock front.

The results of riMPM in combination with the Wendland weighting function are shown in Fig. 7.11. The obtained results are very similar to the ones shown in Fig. 7.10. The regularization successfully improves the condition of the moment matrices making it possible to apply iMPM to a problem where all MP are arranged in a line.

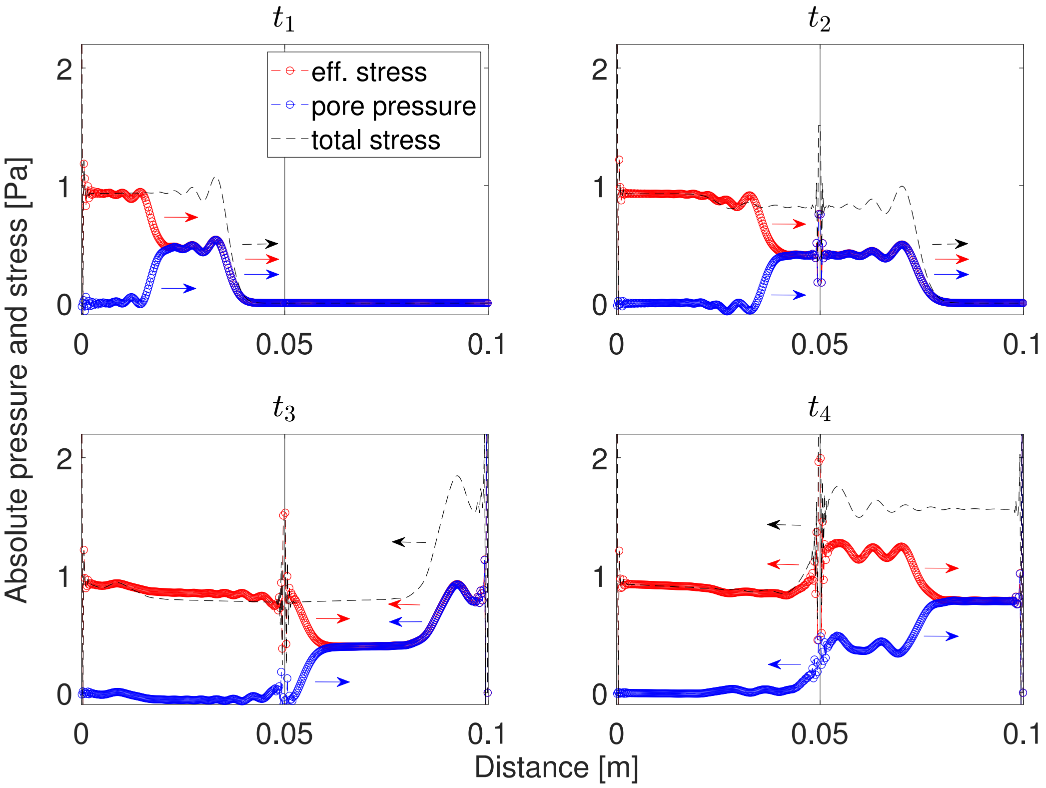


Figure 7.11: Wave propagation at different time instants, from t_1 to t_4, using the Wendland weighting functions with a compact support of $10h$, the linear basis $\mathbf{p}(\mathbf{x}) = (1\,x\,y)^T$ and $\lambda = 0.001$. One empty element is placed in the middle at 0.05m. The arrows indicate the propagation direction of the corresponding shock front.

Chapter 8

Implementation of 2P-MPM for parallel computing

In this chapter the implementation of a two-dimensional 2P-MPM code is discussed. Beginning with a sequential code to calculate required quantities explained in the previous chapters, a parallelized approach is introduced. First, the parallelization is shown for the case of linear B-Splines. Later, the idea is extended to quadratic B-Splines. In the same manner the parallelization can also be carried out for higher order B-Splines and other weighting functions. The key point for the parallelization is the rectangular grid, which facilitates fast neighbour searching. Utilizing GPU computations, the sequential and parallelized code are benchmarked for different setups and the speed-ups are presented.

8.1 2D Implementation

For the implementation, 4-noded quadrilateral elements are considered. As presented in Ch. 5, different shape functions for the nodes and MP can be chosen. For the sake of simplicity, the Standard MPM approach is applied to the fluid and solid phase here. Extension of the implementation to other MPM approaches will be discussed afterwards. The parent quadrilateral element is defined on the interval $[-1,1] \times [-1,1]$ with the four corner nodes n_1, n_2, n_3 and n_4 located at $(-1,-1)$, $(1,-1)$, $(1,1)$ and $(-1,1)$ respectively. The indices 1,...,4 denote the local numbering of nodes inside each element. The linear shape functions in the parent coordinate system (ξ,η) can be defined as

$$\begin{aligned} N_1(\xi,\eta) &= 0.25(1-\xi)(1-\eta) \ , \ N_2(\xi,\eta) = 0.25(1+\xi)(1-\eta)\,, \\ N_3(\xi,\eta) &= 0.25(1+\xi)(1+\eta) \ , \ N_4(\xi,\eta) = 0.25(1-\xi)(1+\eta)\,. \end{aligned} \tag{8.1}$$

The evaluation of these shape functions at the location of MP requires the global coordinates (x_p,y_p) to be transformed in the parent coordinate system. For this purpose the global coordinates of the nodes of the considered element, (x_1^n, y_1^n),...,(x_4^n, y_4^n), can be used as follows

$$\xi_p = 2(x_p - x_1^n)/\Delta x - 1\,, \qquad \eta_p = 2(y_p - y_4^n)/\Delta y - 1\,. \tag{8.2}$$

Here Δx and Δy in Eqn. 8.2 denote the element size in x- and y-direction. Also, the derivatives of the nodal shape functions are required to compute differential operators,

$$\begin{aligned}
\frac{\partial N_1(\xi,\eta)}{\partial \xi}&, = -0.25(1-\eta) \qquad & \frac{\partial N_1(\xi,\eta)}{\partial \eta} &= -0.25(1-\xi)\,, \\
\frac{\partial N_2(\xi,\eta)}{\partial \xi}&, = 0.25(1-\eta) \qquad & \frac{\partial N_2(\xi,\eta)}{\partial \eta} &= -0.25(1+\xi)\,, \\
\frac{\partial N_3(\xi,\eta)}{\partial \xi}&, = 0.25(1+\eta) \qquad & \frac{\partial N_3(\xi,\eta)}{\partial \eta} &= 0.25(1+\xi)\,, \\
\frac{\partial N_4(\xi,\eta)}{\partial \xi}&, = -0.25(1+\eta) \qquad & \frac{\partial N_4(\xi,\eta)}{\partial \eta} &= 0.25(1-\xi)\,.
\end{aligned} \tag{8.3}$$

To obtain the derivatives of Eqn. 8.3 with respect to the global coordinates, they need to be multiplied by the derivatives of the parent coordinates in Eqn. 8.2 with respect to the global coordinates, which are

$$\frac{\partial \xi_p}{\partial x_p} = \frac{2}{\Delta x} \quad \text{and} \quad \frac{\partial \eta_p}{\partial y_p} = \frac{2}{\Delta y}\,. \tag{8.4}$$

Material point to node interpolation (P2N) As in the one-phase case, in the beginning quantities must be interpolated from MP to nodes. Which MP contribute to which nodes depend on the MPM formulation. In the case of Standard MPM one MP is always contributing only to the four nodes of the element where it is located. The weights of its contribution are determined by the nodal shape functions at its location. Therefore, the information

$$\mathbf{N}(x_p,y_p) = [N_1(x_p,y_p), N_2(x_p,y_p), N_3(x_p,y_p), N_4(x_p,y_p)] \tag{8.5}$$

is needed for each MP. It is useful to think of this data as an $1 \times 4-$array, which is associated to a particular MP. In the same manner the $2-$array **dN** of the shape function derivatives is defined as

$$\mathbf{dN}(x_p,y_p) = \begin{bmatrix} \frac{\partial N_1}{\partial \xi}(x_p,y_p) & \frac{\partial N_2}{\partial \xi}(x_p,y_p) & \frac{\partial N_3}{\partial \xi}(x_p,y_p) & \frac{\partial N_4}{\partial \xi}(x_p,y_p) \\ \frac{\partial N_1}{\partial \eta}(x_p,y_p) & \frac{\partial N_2}{\partial \eta}(x_p,y_p) & \frac{\partial N_3}{\partial \eta}(x_p,y_p) & \frac{\partial N_4}{\partial \eta}(x_p,y_p) \end{bmatrix}\,. \tag{8.6}$$

To obtain the arrays **N** and **dN** on a uniform rectangular grid it is necessary to find first the element, where the considered MP is located and identify the nodes and their shape functions for Eqn. 8.5. Then, the parent coordinates need to be determined. Finally, the shape functions and their derivatives can be evaluated in the parent coordinate system.

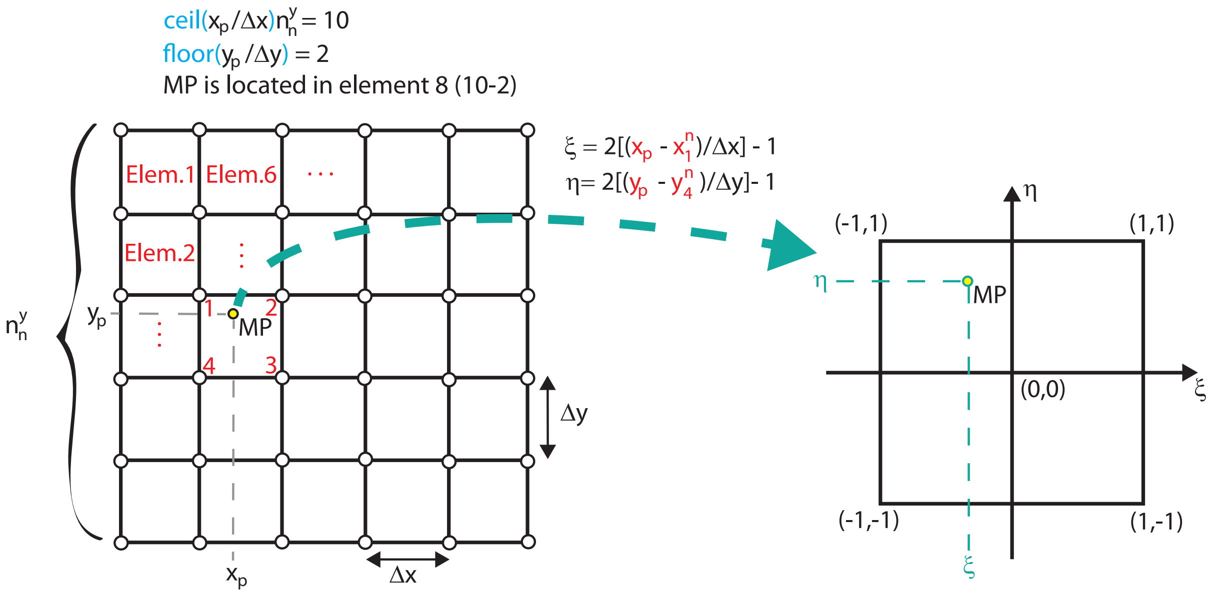


Figure 8.1: Illustrative example of determining **N** and **dN** for a MP:
1. Division of MP coordinates x_p and y_p by the grid sizes Δx and Δy and evaluation of the *floor*− and *ceiling*−functions to find the element, where the MP is located.
2. Transformation to parent coordinate system.
3. Evaluation of shape functions and their derivatives.

Fig. 8.1 shows the described procedure fo one MP. This procedure is independent for each MP and can be performed in parallel.

Force calculation The array **N** is needed to transfer mass, momentum and external forces at MP to the nodes by simply summing all contributions of the shape function N_i at each node i. The internal forces for the solid $\mathbf{f}_s^{int}$ and the fluid $\mathbf{f}_f^{int}$ at each node i can be obtained by utilizing the entries of **dN** as follows

$$
\begin{aligned}
(\mathbf{f}_s^{int})_i &= \sum_{k=1}^{N_{sp}} V_k^s \begin{bmatrix} \frac{\partial N_i}{\partial \xi}(x_k^s,y_k^s)(\sigma_{xx}(x_k^s,y_k^s) + \sigma_{xy}(x_k^s,y_k^s)) \\ \frac{\partial N_i}{\partial \eta}(x_k^s,y_k^s)(\sigma_{xy}(x_k^s,y_k^s) + \sigma_{xy}(x_k^s,y_k^s)) \end{bmatrix}, \\
(\mathbf{f}_f^{int})_i &= \sum_{k=1}^{N_{fp}} V_k^f \begin{bmatrix} \frac{\partial N_i}{\partial \xi}(x_k^f,y_k^f)p(x_k^f,y_k^f) \\ \frac{\partial N_i}{\partial \eta}(x_k^f,y_k^f)p(x_k^f,y_k^f) \end{bmatrix}.
\end{aligned}
\tag{8.7}
$$

In Eqn. 8.7, N_{sp} and N_{fp} denote the number of fluid and solid MP contributing to node i and $V_k^{s/f}$ the volume of the corresponding MP.

Drag-Force The Drag-Force is determined by subtracting the known velocities for fluid and solid at the nodes. It is not necessary to interpolate between nodes and MP here even though extended approaches do so (Mackenzie-Helnwein *et al.*, 2010). Following the

work of Yamaguchi *et al.* (2020), the shape function product in the Drag-Force expression shown in Eqn. 7.2a and 7.2b is replaced by the nodal volume $V_i = \int_\Omega N_i(\mathbf{x})dV$.
Once all required forces $(\mathbf{f}_s)_i$ and $(\mathbf{f}_f)_i$ are known at every node i, the new momentum can be calculated by adding the increments $\Delta t(\mathbf{f}_s)_i$ and $\Delta t(\mathbf{f}_f)_i$. The new nodal velocities $\hat{\mathbf{v}}_s$ and $\hat{\mathbf{v}}_f$ are obtained by dividing the new solid momentum by the nodal solid mass m_i^s and new fluid momentum by the nodal fluid mass m_i^f.

Node to MP interpolation N2P The new velocities at each node are now transferred to those MP, which are located inside the compact support of N_i. In the case of quadrilateral 4-noded elements, the support of N_i covers the 4 adjacent elements. The weights for the transfer are defined by $\mathbf{N}(x_p,y_p)$. While $\hat{\mathbf{v}}_f$ is only transferred to fluid MP, $\hat{\mathbf{v}}_s$ is only transferred to solid MP. In the case of quadrilateral 4-noded elements, always 4 nodes contribute to each MP. The transferred velocities at the MP can be used to update their positions by the forward Euler integration scheme,

$$\begin{aligned} \begin{pmatrix} x_p^s(t_{n+1}) \\ y_p^s(t_{n+1}) \end{pmatrix} &= \begin{pmatrix} x_p^s(t) \\ y_p^s(t) \end{pmatrix} + \Delta t \begin{pmatrix} v_{xp}^s(t) \\ v_{yp}^s(t) \end{pmatrix} , \\ \begin{pmatrix} x_p^f(t_{n+1}) \\ y_p^f(t_{n+1}) \end{pmatrix} &= \begin{pmatrix} x_p^f(t) \\ y_p^f(t) \end{pmatrix} + \Delta t \begin{pmatrix} v_{xp}^f(t) \\ v_{yp}^f(t) \end{pmatrix} . \end{aligned} \tag{8.8}$$

In Eqn. 8.8 the superscripts s and f refer to solid and fluid MP, while the indices x and y denote the Cartesian components.

Deformation gradient The deformation gradient describes the deformation of the element with respect to a reference configuration. In the case of the Standard MPM, an updated-Lagrangian approach is used, which refers the current configuration to the previous one. In a total Lagrangian approach, the reference configuration is the initial one for all calculation steps.
The use of isoparametric elements allows to compute the incremental deformation gradients utilizing $\mathbf{dN}$. Therefore the nodal displacement increments Δu_I are calculated by multiplying the nodal velocities with Δt. The incremental deformation gradient for a solid MP can then be calculated as

$$\mathbf{F}_s^{incr}(x_p^s,y_p^s) = \mathbf{I}_2 + \begin{pmatrix} \sum_{i=1}^4 u_{xi}^s \frac{\partial N_i}{\partial \xi}(x_p^s,y_p^s) & \sum_{i=1}^4 u_{xi}^s \frac{\partial N_i}{\partial \eta}(x_p^s,y_p^s) \\ \sum_{i=1}^4 u_{yi}^s \frac{\partial N_i}{\partial \xi}(x_p^s,y_p^s) & \sum_{i=1}^4 u_{yi}^s \frac{\partial N_i}{\partial \eta}(x_p^s,y_p^s) \end{pmatrix} . \tag{8.9}$$

In Eqn. 8.9 the symbol $\mathbf{I}_2$ denote the 2×2 identity matrix. The nodal displacements u_{xi}^s, u_{yi}^s, u_{xi}^f and u_{yi}^f are obtained by multiplication of the corresponding nodal velocities and Δt. The summation in Eqn. 8.9 is performed over the 4 nodes of the element where the considered MP is located. The deformation gradient for a solid MP at the time step $n+1$ can be calculated as

$$\mathbf{F}_s^{n+1} = \mathbf{F}_s^{incr}\mathbf{F}_s^n . \tag{8.10}$$

The deformation gradient is used to define a strain measure. For example the small strain tensor

$$\mathbf{E} = \frac{1}{2}\left(\mathbf{F}_s + \mathbf{F}_s^T\right) - \mathbf{I}\,, \tag{8.11}$$

where the superscript t denotes the transpose of the deformation gradient. The rate of the deformation gradient $\dot{\mathbf{F}}$ can be calculated as

$$\dot{\mathbf{F}} = \left(\frac{\mathbf{F}^{n+1} - \mathbf{F}^{n}}{\Delta t}\right)\,. \tag{8.12}$$

The same procedure can be applied to fluid MP. Alternatively, the pore pressure which is calculated from the divergence of both velocity fields can be used. In the considered case, the latter is used to update the volume change of the fluid MP. To obtain the divergence at the location of a fluid MP again $\mathbf{dN}$ is used as

$$\begin{aligned}\mathbf{div}(\mathbf{v}^s)(x_p^f,y_p^f) &= \sum_{i=1}^{4} v_{xi}^s \frac{\partial N_i}{\partial \xi}(x_p^f,y_p^f) + \sum_{i=1}^{4} v_{yi}^s \frac{\partial N_i}{\partial \eta}(x_p^f,y_p^f)\,,\\ \mathbf{div}(\mathbf{v}^f)(x_p^f,y_p^f) &= \sum_{i=1}^{4} v_{xi}^f \frac{\partial N_i}{\partial \xi}(x_p^f,y_p^f) + \sum_{i=1}^{4} v_{yi}^f \frac{\partial N_i}{\partial \eta}(x_p^f,y_p^f)\,.\end{aligned} \tag{8.13}$$

In Eqn.8.13, the summation is performed over the 4 nodes of the element where the fluid MP is located. The volumetric strain increment of the fluid MP can now be calculated as

$$\Delta\boldsymbol{\epsilon}_v^f(x_p^f,y_p^f) = \left(n\mathbf{div}(\mathbf{v}^f)(x_p^f,y_p^f) - (1-n)\mathbf{div}(\mathbf{v}^s)(x_p^f,y_p^f)\right)/\Delta t\,. \tag{8.14}$$

The strain quantities at the fluid and solid MP can be used together with constitutive models to calculate the stresses or pressures, which are needed to obtain the internal forces for the next computational step.

8.2 Parallel-Computing

The presented implementation in Sec. 8.1 can avoid loops (except the time integration loop) and implement everything as element-wise array operations. The resulting code then can be executed on a multi-threaded central processing unit (CPU) or a graphics processing unit (GPU) with CUDA cores. The parallel implementation of MPM was shown X. Li and Sulsky (2000), Parker (2002), and Huang *et al.* (2008). In the present work, we are focussing on the shared memory approach and extend the parallelization to the 2P-MPM formulation. In the following, benchmarking of the wave propagation example from Sec. 7.6 is performed. Therefore, a loop-based code on a CPU (Intel i-9) is compared against a vectorized code on a GPU (Titan Z). The parallel GPU-based code for the wave propagation benchmark is shown in Appx. C.

In the first benchmark, the column is discretised by 3002 nodes, 4400 fluid and 4400 solid MP. Three variations are considered, in particular 10, 20 and 100 time step. The computational times for the sequential code executed on a CPU and the parallel code executed on a GPU are summarized in Fig. 8.2.

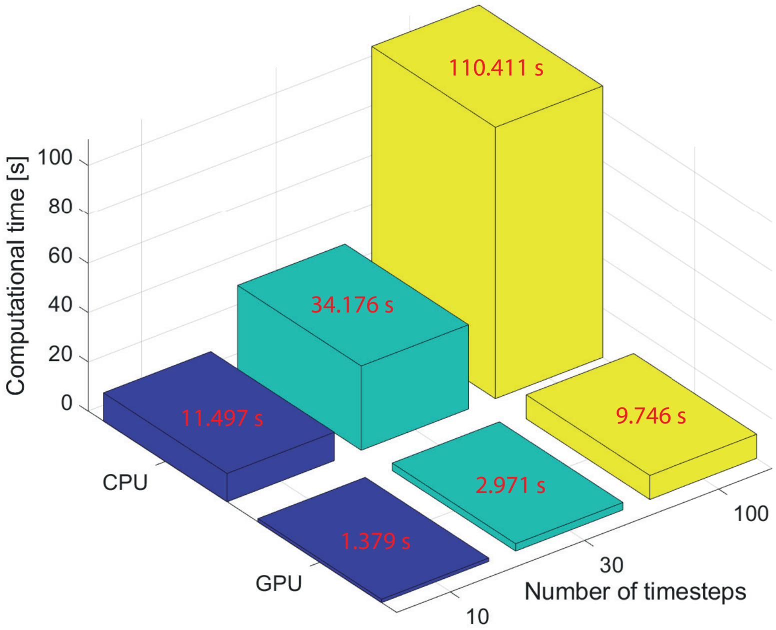


Figure 8.2: Benchmark 1: Computational time of CPU and GPU calculations for different number of time steps utilizing 3002 nodes, 4400 fluid and 4400 solid MP.

In the second benchmark, the number of nodes and MP is increased by a factor of 10 to 30002 nodes and 44000 fluid and solid MP. The results and computational times for the second benchmark are summarized in Fig. 8.3. The computational time of the sequential code increases proportionally with the degree of discretization by a factor of approx. 10. The increase of the computational time for the parallel code depends on the number of time steps and is approx. 4 in the case of 100 steps.
The speed-up factor for each calculation is presented in Fig. 8.4. From the results, it can be concluded that besides the significant acceleration of the GPU calculations, the speed-up increases for larger problems and more time steps. This is due to the structure of a loop-based MPM code where the size of the major loops is equal to the number of MP and nodes. While this size is growing with larger models, a GPU code is treating these operations simultaneously.

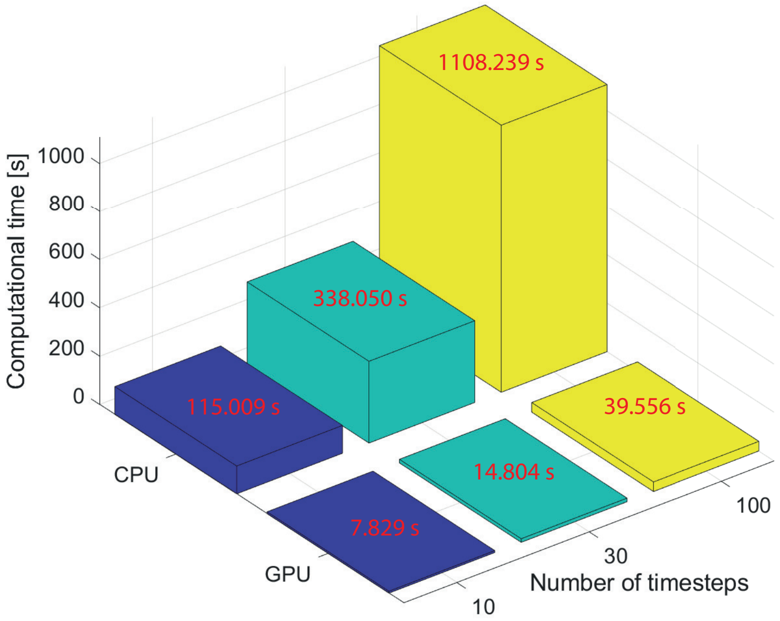


Figure 8.3: Benchmark 2: Computational time of CPU and GPU calculations for different number of time steps utilizing 30002 nodes, 44000 fluid and 44000 solid MP.

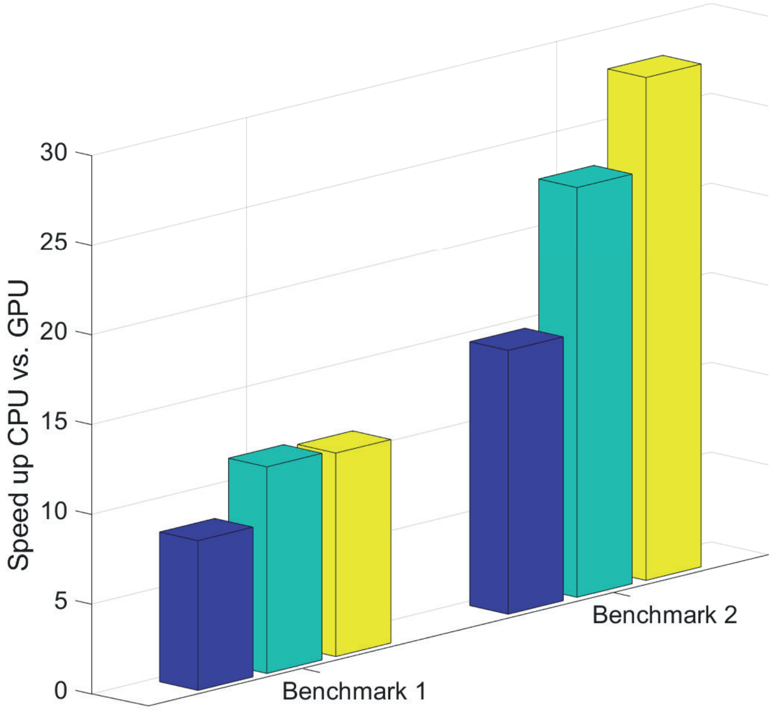


Figure 8.4: Speed-up of computational times between the sequential code on a CPU and parallel code on a GPU for both benchmarks. The speed-ups are shown for 10 (blue), 30 (green) and 100 (yellow) timesteps.

8.3 Extension to B-Splines

While for linear shape functions a MP always contributes to 4 nodes (for 2D-quadrilateral elements), this is no longer true for higher order B-splines, due to the larger compact support. In the following we consider the two-dimensional quadratic B-spline from Sec. 5.3.3 described by Eqn. 5.26 and 5.27.
In Fig. 8.5, the area in which the centred quadratic B-spline from Eqn. 5.3.3 at the i-th node is non-zero, is coloured. The red and green squares show the piecewise defined domains corresponding to the one-dimensional expression in Eqn. 5.3.3. While in the case of linear shape functions it was enough to know the element, where a MP is located, to determine the 4 nodes, to which it contributes. In case of a quadratic B-Spline, an additional distinction must be made.

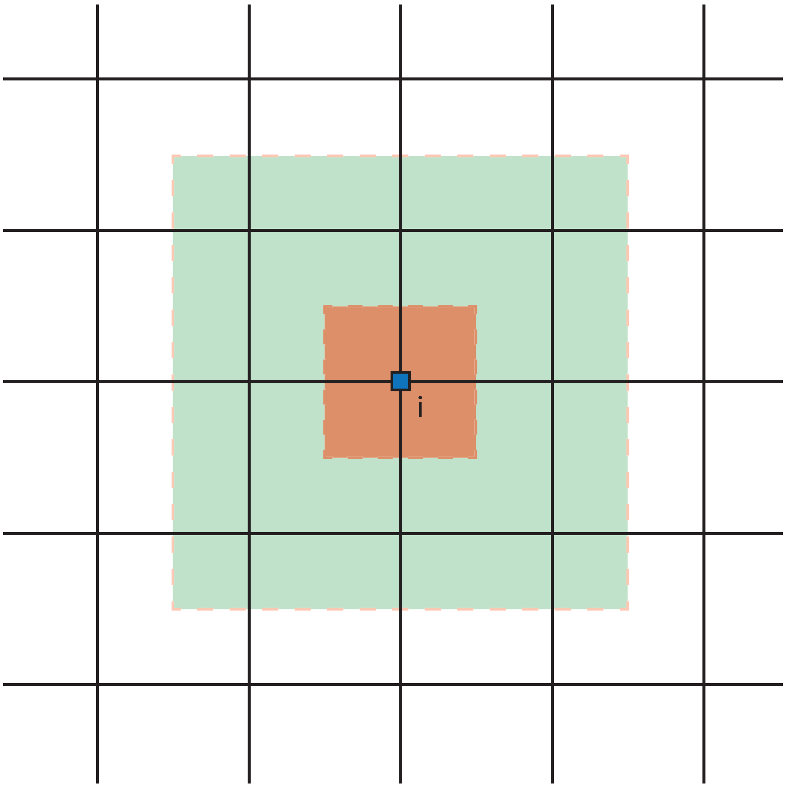


Figure 8.5: Compact support and areas of the piece-wise defined quadratic B-Spline centred at the i-th node. The piecewise definition of the quadratic B-Spline results in two areas, the red and green squares, where different expressions need to be evaluated. Outside the coloured areas the quadratic B-Spline is zero.

Fig. 8.6 shows the contributing nodes for an MP located in one of the coloured domains. Each element is thus divided into 4 areas in which different nodes must be considered. There are always 9 nodes involved in the calculation. Thus, the corresponding arrays from Sec. 8.1 and 8.1 must be enlarged. All sums over nodes then run till 9 and not 4. All arrays that have 4 columns for the shape functions now have 9 columns. This can also be extended to higher order B-Splines and other shape functions where, a bigger compact support will enlarge the arrays **N** and **dN**.

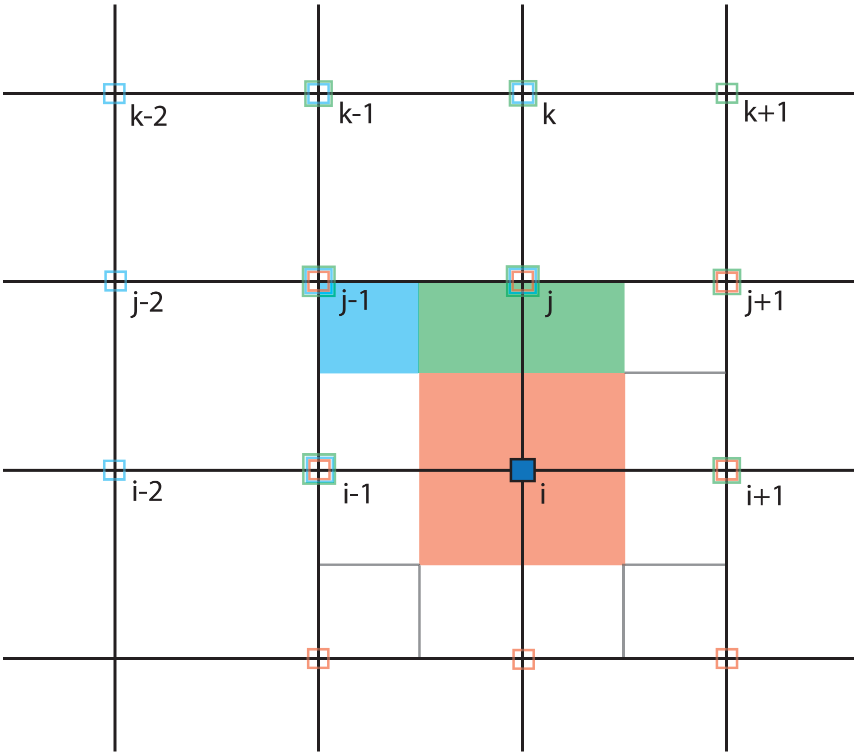


Figure 8.6: Contributing nodes for MP located in one of the coloured squares. MP located in the red area contribute to the nodes with red squares. MP located in the blue area contribute to the nodes with blues squares. MP located in the green area contribute to the nodes with green squares.

Chapter 9

Geotechnical Application

This chapter will focus on applying the presented methods to geotechnical problems. In particular, dynamic problems with large deformations are a challenging task. These problems include e.g. pile installation, dynamic compaction methods or explosions in the subsurface. In particular, the latter example will be considered. The simulation of such problems using MPM can lead to unfavourable MP distributions or empty elements, as described in the previous chapters. Therefore, the proposed methods are applied here. In addition, the numerical simulation of geotechnical problems requires advanced material models for the soil. Therefore, this chapter describes how the presented methods can be extended for other material models.

9.1 Constitutive Behaviour

Material models are often formulated as a relationship between strain and stress or their rates. The strain measure should be independent of rigid body motions and translations. Measures of this type can be obtained from the deformation gradient. As already shown in Sec. 4.2.1, this can be written by polar decomposition in the following form $\mathbf{F} = \mathbf{R}\mathbf{U}$, where $\mathbf{R}$ is a rotation matrix, and $\mathbf{U}$ represents the deformation. Thus, a suitable strain measure would be $\mathbf{U} - \mathbf{I}$, which does not include rigid-body motions and is zero for the case of no deformation, i.e., $\mathbf{F} = \mathbf{I}$. In Ch. 8 we have already introduced the strain tensor $\mathbf{E} = 0.5(\mathbf{F} + \mathbf{F}^T) - \mathbf{I}$ for small strains and its rate. This strain tensor depends on rigid-body motions. It is therefore only suitable for small strains, which are guaranteed by small time steps in the Updated-Lagrangian approach, for example. In general, the polar decomposition of $\mathbf{F}$ is very computationally expensive, so it is helpful to define other strain measures that do not depend on $\mathbf{R}$. Another strain measure that satisfies this requirement is the Green strain tensor, defined as $\mathbf{E} = 0.5(\mathbf{F}^T\mathbf{F} - \mathbf{I})$. Inserting the polar decomposition makes it easy to prove that $\mathbf{R}$ is eliminated in the Green strain tensor. The multiplication $\mathbf{F}^T\mathbf{F}$ for the Green strain tensor is much more efficient than the polar decomposition and is, therefore, better suited for large problems. Many material models are defined in terms of strain and stress rates. For this purpose, it is useful to consider the velocity gradient tensor defined as $\mathbf{L} = \partial\mathbf{v}/\partial\mathbf{x}$. This can be related to the deformation gradient as $\mathbf{L} = \dot{\mathbf{F}}\mathbf{F}^{-1}$. The velocity gradient can be decomposed into a symmetric and a skew-symmetric part as $\mathbf{L} = 0.5(\mathbf{L} + \mathbf{L}^T) + 0.5(\mathbf{L} - \mathbf{L}^T)$. These parts are denoted by the rate of deformation tensor $\mathbf{D} = 0.5(\mathbf{L} + \mathbf{L}^T)$ and the spin tensor $\mathbf{W} = 0.5(\mathbf{L} - \mathbf{L}^T)$. The stress measures associated with the presented strain measures are introduced as energy or work conjugated pairs. For example, the Cauchy stress and the strain rate $\mathbf{D}$ form an energetically conjugate pair. The substantial derivative of the Cauchy stress $\mathrm{D}\boldsymbol{\sigma}/\mathrm{D}t$ is not free of rigid-body motions and is incorrect for large deformations. Therefore, objective stress rates are usually used in rate-dependent material models. One such objective stress rate is the Jaumann rate, which is defined as follows,

$$\boldsymbol{\sigma}^{\nabla} = \frac{\mathrm{D}\boldsymbol{\sigma}}{\mathrm{D}t} - \mathbf{W}\boldsymbol{\sigma} - \boldsymbol{\sigma}\mathbf{W}^T\,. \tag{9.1}$$

For a material model, which calculates the Jaumann rate based on given strain measures the updated Cauchy stress can be calculates as

$$\boldsymbol{\sigma}^{(n+1)\Delta t} = \boldsymbol{\sigma}^{n\Delta t} + \frac{\mathrm{D}\boldsymbol{\sigma}}{\mathrm{D}t}\Delta t\,. \tag{9.2}$$

It should be noted that all derived quantities for the strains are calculated from the deformation gradient. In MPM, this corresponds to the calculations in Eqn. 8.9 and 8.10. Thus, all these quantities can be calculated at the corresponding MP. In the following, we present a geotechnical example of underground explosions, where different material models will be utilized. This example is dynamic, includes large deformations, and the propagating waves are significant.

9.2 Underground Explosions

Unexploded ordnance from the Second World War can still be found in large conurbations. The detonator is often still functional and can trigger a detonation. During exploratory work or dredging, the detonators can be activated by vibrations. In such cases, it is crucial to assess the effects of such detonations and develop safety measures to limit the damage. Depending on the depth at which the explosion takes place and the explosion energy, two extreme cases can generally be distinguished, as shown in Fig. 9.1. In case 1, the initiated stress waves propagate through the coherent grain structure from the explosion point in all directions. As soon as the waves arrive at the surface, they cause it to oscillate and are reflected into the ground as tensile waves since the surface is a free boundary. In the case of B, the ground above the explosion point is thrown out, and a crater is formed. This division of the initially continuous soil body interrupts the path of wave propagations in the soil, and a large part continues to propagate in the air as a pressure wave. Overall, the wave propagation in case B is very complex and can only be described qualitatively. The ground's fractionation and the crater's shape, which decisively influence the wave propagation, can only be described with great uncertainty.

Numerical example The detonation of explosives can be modelled in different ways (Mandal *et al.*, 2021). On the one hand, a time-dependent kinematic boundary condition,

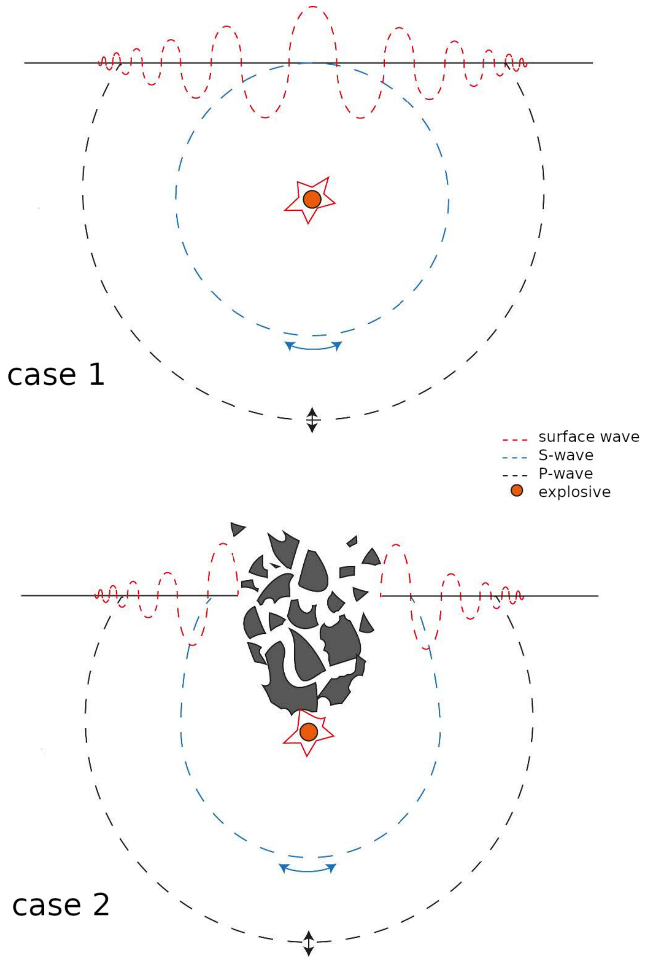


Figure 9.1: Schematic representation of wave propagation after an underground explosion. Case A shows an explosion without and case B with crater formation.

e.g. for the velocity, can be used. On the other hand, equations of state (EOS) or time-dependent pressure functions, which determine the hydrostatic pressure at MP, are suitable for many explosives. The Friedlander equation (Friedlander, 1946) describes a typical time course of the pressure

$$P(t) = P_0 + P_s \exp\left(-\frac{t}{t^*}\right)\left(1 - \frac{t}{t^*}\right) \text{ for } t > 0, \text{ else } P(t) = P_0 \, . \tag{9.3}$$

at a fixed location, where P_0 is the ambient pressure and P_s and t^* is the maximum pressure and the duration of the pressure wave.
The EOS, according to Jones-Wilkins-Lee (JWL) describes the hydrostatic pressure of the explosive using the change in volume and energy of an MP:

$$P(E,V) = A\left(1 - \frac{\omega}{R_1 V}\right) e^{-R_1 V} + B\left(1 - \frac{\omega}{R_2 V}\right) e^{-R_2 V} + \frac{\omega E}{V} \quad . \tag{9.4}$$

In Tab.9.1 typical model parameters for the explosive tetranitrate (PETN) for the JWL equation of state is shown.

density (g/cm^3)	A (Mbar)	B (Mbar)	ω (–)	R_1 (–)	R_2 (–)
1.77	6.17	0.169	0.25	4.4	1.2

Table 9.1: JWL-Parameter for PETN according to Lee *et al.* (1973) (1 Mbar correspond to 10^8 kN/m^2)

The rate of internal energy per unit initial volume $\dot{E}$ is obtained from the inner (double dot) product of the Cauchy stress and rate of deformation tensors. This leads to the following expression, which is calculated at each MP Ma *et al.*, 2009; X. Zhang *et al.*, 2017:

$$\dot{E} = \boldsymbol{\sigma} : \mathbf{D}\,, \tag{9.5}$$

where for simplicity, thermal expressions were neglected. Most models for explosives and their parameters are calibrated based on experiments and are usually valid only for detonations in air or water. For underground detonations, the parameters must be adapted from the literature. For example, this adjustment can be made based on existing measured values. Model parameters for the soil can be obtained from samples or estimations if the soil type is known. In the following, soil parameters for the hypoplastic soil model by Niemunis (2003) were obtained partly from samples in the harbour of Hamburg and partly estimated.

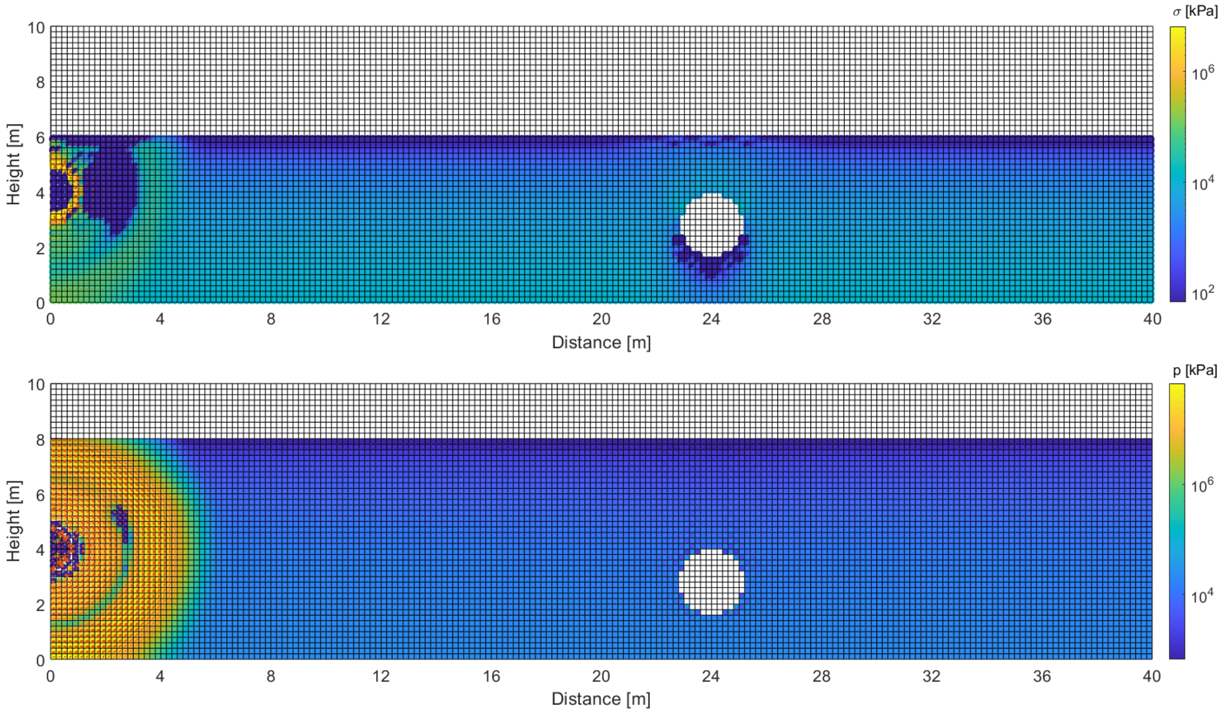
φ_c (rad)	h_s (kPa)	n (–)	e_{d0} (–)	e_{c0} (–)	e_{i0} (–)	α (–)	β (–)	m_T (–)	m_R (–)	$R_{\max}$ (–)	β_r (–)	χ (–)
0.5533	581 000	0.4038	0.442	0.818	0.941	0.033	1.78	*2.0*	*5.0*	*0.0001*	*0.5*	*6.0*

Table 9.2: Experimentally determined parameters and estimated parameters (*italic*) for the hypoplastic material model for a composite sample of sand.

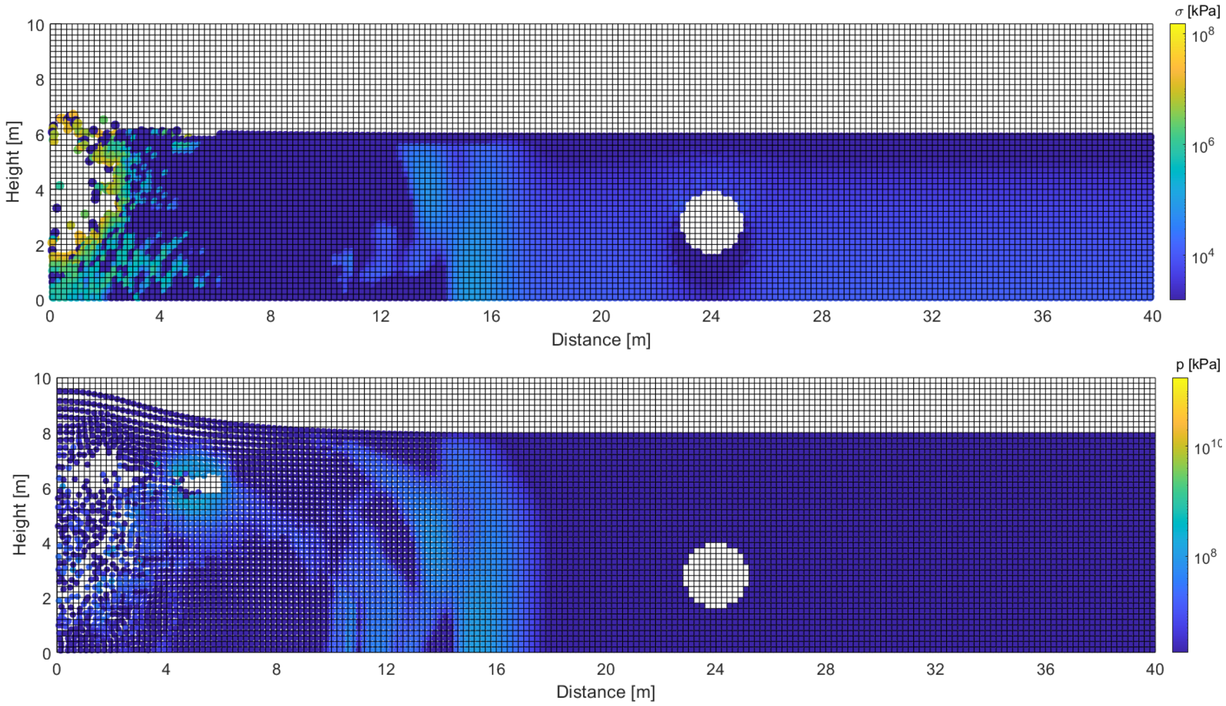
In addition, a modification to the material model mentioned above [1] was made to prevent zero void ratios during the simulation. This is since in Niemunis (2003) an upper and lower bound for the void ratio is given. The bounds for the void ratios are defined according to Bauer (1996), which for a very large mean pressure $\mathbf{tr}(\boldsymbol{\sigma})/3$ approaches zero. Near the detonation, such high pressures occur during the simulation. To prevent the void ratio to become zero, a logistic growth function of the form $f(\mathbf{tr}(\boldsymbol{\sigma})) = e_{s_{1/2}}/(1 + e^{k_1(\mathbf{tr}(\boldsymbol{\sigma})+k_2)})$ is added to the lower and upper bound of the void ratio. The values $e_{s_{1/2}}$ denote the new smallest and the largest void ratio that can appear. The constants k_1 and k_2 smooth and shift the function $f(\mathbf{tr}(\boldsymbol{\sigma}))$ from zero to $e_{s_{1/2}}$. In the following, the effect of a detonation on a tunnel is shown as an example. The example includes both saturated soil and free water. A linear polynomial basis with Wendland weighting functions and $\lambda = 0.01$ were applied for the simulations. Further tests have to be carried out to evaluate the simulation results. Also, the scaling of the explosive quantity must be investigated in more detail.
In Fig. 9.2 the results of two MPM simulations are shown, which describe a detonation in the saturated soil. Above the saturated soil, a 2 m thick free water layer is located. A

---

[1] The implementation of the hypoplastic soil model was provided by D.Zobel.



Explosives 50 kg PETN: solid(up), fluid(down)



Explosive 500 kg PETN: solid(up), fluid(down)

Figure 9.2: Calculated distribution of the 2-norm of the total stress tensor $||\boldsymbol{\sigma}||_2$ and the pore water pressure p for a blast charge of 50 kg PETN (top) and 500 kg PETN (bottom) after a time of $t = 0.0447$ s.

tunnel is placed at a distance of about 24 m. In the first simulation, the explosion of an extrapolated explosive charge of 50 kg and the second of 500 kg PETN is simulated. The different charge quantities cause in Fig. 9.2 the deformation of the grain structure and the formation of a surface wave. These models need further validation to understand better free and pore water's influence. Here they were used only as an example of the developed numerical method.

Chapter 10

Conclusions and Outlook

MPM development is a current research area with numerous publications each year. Many of the problems shown in this work, which have not yet been definitively solved, require sophisticated methods and further investigation. In this work, the approach MLS in MPM has been further investigated and developed. The proximity of MPM to meshfree methods and MLS seems to be a promising approach for transferring already established methods to MPM. The regularization proposed in this work improves the robustness of MPM so that MLS with higher order polynomial bases can be used despite unfavourable MP distribution. Besides accuracy also, efficiency is of great importance. Therefore, the regularization in MPM seems to be a promising method, as its computational costs are lower than the calculation of the pseudo inverse. The issue of contact loss remains a significant challenge in MPM. Larger compact supports of weighting functions can solve this problem only partly and come at the cost of higher computational times. As the compact supports in MPM are defined regarding the grid, a combined re-meshing approach might improve this issue. The proposed Wendland function could show advantages over the conventional weighting functions in the convergence comparisons. The presented renewal can be applied to both 1P and 2P-MPM. It is still essential to further improve the accuracy and stability of the methods, particularly for large deformations. Further studies on the optimal approximation of the different field variables are needed. Also, the combination of Gaussian integration as in iMPM needs further investigation. The efficiency of 2P-MPM could be improved in this work by parallelization, allowing the simulation of larger boundary value problems. Implementation in a high-level language could further provide a significant performance improvement here. This can also be investigated in MATLAB using the so-called MATLAB executable (MEX). Distributed memory parallelization would have to be considered to increase efficiency further to allow the code to be run on large computing clusters.

Bibliography

Abe, K., K. Soga, and S. Bandara (Mar. 2014): Material Point Method for Coupled Hydromechanical Problems. In: *Journal of Geotechnical Engineering* 140, p. 04013033. DOI: `10.1061/(ASCE)GT.1943-5606.0001011`.

Anderson, J. D. and J. Wendt (1995): *Computational fluid dynamics*. Vol. 206. Springer.

Bandara, S. (2013): Material Point Method to simulate large deformation problems in fluid-saturated granular medium. Dissertation. Department of Engineering, University of Cambridge.

Bardenhagen, S. and E. Kober (June 2004): The Generalized Interpolation Material Point Method. In: *CMES - Computer Modeling in Engineering and Sciences* 5.

Bauer, E. (1996): Calibration of a comprehensive hypoplastic model for granular materials. In: *Soils and foundations* 36(1), pp. 13–26. DOI: `10.3208/sandf.36.13`.

Bear, J. (1988): *Dynamics of fluids in porous media*. Dover books on physics and chemistry. New York: Dover.

Belytschko, T., Y. Krongauz, D. Organ, M. Fleming, and P. Krysl (1996): Meshless methods: an overview and recent developments. In: *Computer methods in applied mechanics and engineering* 139(1-4), pp. 3–47.

Belytschko, T., W. K. Liu, B. Moran, and K. Elkhodary (2013): *Nonlinear finite elements for continua and structures*. John wiley & sons.

Belytschko, T., Y. Y. Lu, and L. Gu (1994): Element-free Galerkin methods. In: *International journal for numerical methods in engineering* 37(2), pp. 229–256.

Beuth, L. (2012): Formulation and application of a quasi-static Material Point Method. Dissertation. Mitteilung des Instituts für Geotechnik der Universität Stuttgart, Heft 66.

Biot, M. A. (1956a): Theory of propagation of elastic waves in a fluid-saturated porous solid. I. Low-frequency range. In: *The Journal of the Acoustical Society of America* 28(2), pp. 168–178. DOI: `10.1121/1.1908239`.

Biot, M. A. (1956b): Theory of propagation of elastic waves in a fluid-saturated porous solid. II. Higher frequency range. In: *The Journal of the Acoustical Society of America* 28(2), pp. 179–191. DOI: `10.1121/1.1908241`.

Boer, R. de (2000): *Theory of porous media: highlights in historical development and current state*. Springer Science & Business Media. DOI: `10.1007/978-3-642-59637-7`.

Boor, C. de (Jan. 1978): *A Practical Guide to Spline*. Vol. Volume 27. DOI: `10.2307/2006241`.

Brackbill, J. (1988): The ringing instability in particle-in-cell calculations of low-speed flow. In: *Journal of Computational Physics* 75(2), pp. 469–492.

Brunton, S. L. and J. N. Kutz (2019): *Data-driven science and engineering: Machine learning, dynamical systems, and control.* Cambridge University Press. DOI: `10.1017/9781108380690`.

Chmelnizkij, A., F. Ceccato, J. Grabe, and P. Simonini (2019): 1D Wave propagation in saturated soils: verification of two-phase MPM. In: *2nd International conference on the material point method for modelling soil-water-structure interaction. Cambridge, UK.*

Coetzee, C., P. Vermeer, and A. Basson (2005): The modelling of anchors using the material point method. In: *International journal for numerical and analytical methods in geomechanics* 29(9), pp. 879–895.

Coetzee, C. J. (2004): The modelling of granular flow using the particle-in-cell method. PhD thesis. Stellenbosch: University of Stellenbosch.

Craig, R. F. (2004): *Craig's soil mechanics.* CRC press.

de Vaucorbeil, A., V. P. Nguyen, S. Sinaie, and J. Y. Wu (2020): *Chapter Two - Material point method after 25 years: Theory, implementation, and applications.* Ed. by S. P. Bordas and D. S. Balint. DOI: `https://doi.org/10.1016/bs.aams.2019.11.001`. URL: `https://www.sciencedirect.com/science/article/pii/S0065215619300146`.

Drumheller, D. (2000): On theories for reacting immiscible mixtures. In: *International journal of engineering science* 38(3), pp. 347–382.

Edwards, E. and R. Bridson (2012): A high-order accurate particle-in-cell method. In: *International Journal for Numerical Methods in Engineering* 90(9), pp. 1073–1088.

Ergun, S. (1952): Fluid flow through packed columns. In: *Chem. Eng. Prog.* 48, pp. 89–94.

Fern, E., A. Rohe, K. Soga, and E. Alonso (Jan. 2019): *The Material Point Method for Geotechnical Engineering: A Practical Guide.* DOI: `10.1201/9780429028090`.

Friedlander, F. G. (1946): The diffraction of sound pulses I. Diffraction by a semi-infinite plane. In: *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 186(1006), pp. 322–344. DOI: `10.1098/rspa.1946.0046`.

Garg, S., D. Brownell Jr, J. Pritchett, and R. Herrmann (1975): Shock- wave propagation in fluid- saturated porous media. In: *Journal of Applied Physics* 46(2), pp. 702–713. DOI: `10.1063/1.321634`.

Garg, S., A. H. Nayfeh, and A. Good (1974): Compressional waves in fluid-saturated elastic porous media. In: *Journal of Applied Physics* 45(5), pp. 1968–1974. DOI: `10.1063/1.1663532`.

Gingold, R. A. and J. J. Monaghan (1977): Smoothed particle hydrodynamics: theory and application to non-spherical stars. In: *Monthly notices of the royal astronomical society* 181(3), pp. 375–389.

Gritton, C., M. Berzins, and R. M. Kirby (2015): Improving accuracy in particle methods using null spaces and filters. In: *PARTICLES IV: proceedings of the IV International Conference on Particle-Based Methods: fundamentals and applications.* CIMNE, pp. 202–213.

Hammerquist, C. C. and J. A. Nairn (2017): A new method for material point method particle updates that reduces noise and enhances stability. In: *Computer methods in applied mechanics and engineering* 318, pp. 724–738.

Harlow, F. H. (Mar. 1962): *The particle-in-cell method for numerical solution of problems in fluid dynamics.* Tech. rep. DOI: `10.2172/4769185`. URL: `https://www.osti.gov/biblio/4769185`.

Hong, S. J., R. S. Sandhu, and W. E. Wolfe (1988): On Garg's solution of Biot's equations for wave propagation in a one-dimensional fluid-saturated elastic porous solid. In: *International Journal for Numerical and Analytical Methods in Geomechanics* 12(6), pp. 627–637. DOI: `10.1002/nag.1610120605`. URL: `http://dx.doi.org/10.1002/nag.1610120605`.

Huang, P., X. Zhang, S. Ma, and H. Wang (2008): Shared memory OpenMP parallelization of explicit MPM and its application to hypervelocity impact. In: *Computer Modeling in Engineering & Sciences* 38(2), pp. 119–148.

Huerta, A., T. Belytschko, S. Fernández-Méndez, T. Rabczuk, X. Zhuang, and M. Arroyo (2017): Meshfree Methods. In: *Encyclopedia of Computational Mechanics Second Edition.* John Wiley and Sons, Ltd, pp. 1–38. DOI: `https://doi.org/10.1002/9781119176817.ecm2005`. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119176817.ecm2005`.

Idelsohn, S. R., E. Oñate, F. Del Pin, and N. Calvo (2006): Fluid–structure interaction using the particle finite element method. In: *Computer methods in applied mechanics and engineering* 195(17-18), pp. 2100–2123.

Jiang, C., C. Schroeder, J. Teran, A. Stomakhin, and A. Selle (2016): The material point method for simulating continuum materials. In: *ACM SIGGRAPH 2016 Courses*, pp. 1–52.

Jong, G. d. J. de (1956): *"Wat gebeurt er in de grond tijdens het heien (Dutch)" (What happens in the soil during pile driving).* Vol. 25. DE INGENIEUR.

Al-Kafaji, I. (2013): Formulation of a dynamic Material Point Method (MPM) for geomechanical problems. Dissertation. Fakultät für Bau- und Umweltingenieurwissenschaften der Universität Stuttgart. DOI: `10.18419/opus-496`.

Knupp, P. (2002): *Verification of computer codes in computational science and engineering.* Chapman and Hall/CRC. DOI: `/10.1201/9781420035421`.

Koch, D. L. and R. J. Hill (2001): Inertial effects in suspension and porous-media flows. In: *Annual Review of Fluid Mechanics* 33(1), pp. 619–647.

Landshoff, R. (Jan. 1955): *A Numerical Method for Treating Fluid Flow in the Presence of Shocks.* Tech. rep. DOI: `10.2172/4364774`. URL: `https://www.osti.gov/biblio/4364774`.

Lee, E., M. Finger, and W. Collins (Jan. 1973): *JWL equation of state coefficients for high explosives.* Tech. rep. DOI: `10.2172/4479737`. URL: `https://www.osti.gov/biblio/4479737`.

Li, B., F. Habbal, and M. Ortiz (2010): Optimal transportation meshfree approximation schemes for fluid and plastic flows. In: *International journal for numerical methods in engineering* 83(12), pp. 1541–1579.

Li, X. and D. Sulsky (2000): v. In: *WIT Transactions on Information and Communication Technologies* 23. DOI: `10.2495/HPC000041`.

Liu, W. K., S. Jun, and Y. F. Zhang (1995): Reproducing kernel particle methods. In: *International journal for numerical methods in fluids* 20(8-9), pp. 1081–1106.

M.Steffen, P.Wallstedt, J.Guilkey, R.Kirby, and M.Berzins (July 2008): Examination and analysis of implementation choices within the Material Point Method (MPM). In: *CMES - Computer Modeling in Engineering and Sciences* 31, pp. 107–127.

Ma, S., X. Zhang, Y. Lian, and X. Zhou (2009): Simulation of high explosive explosion using adaptive material point method. In: *Computer Modeling in Engineering and Sciences (CMES)* 39(2), pp. 104–124. DOI: `10.3970/cmes.2009.039.101`.

Mackenzie-Helnwein, P., P. Arduino, W. Shin, J. Moore, and G. Miller (2010): Modeling strategies for multiphase drag interactions using the material point method. In: *International journal for numerical methods in engineering* 83(3), pp. 295–322.

Mandal, J., M. Goel, and A. Agarwal (2021): Surface and buried explosions: an explorative review with recent advances. In: *Archives of Computational Methods in Engineering*, pp. 1–21. DOI: `10.1007/s11831-021-09553-2`.

Martinelli, M. and A. Rohe (2015): Modelling fluidisation and sedimentation using material point method. In: *1st Pan-American Congress on Computational Mechanics - PANACM 2015*. Ed. by Idelsohn, V. Sonzogni, A. Coutinho, M. Cruchaga, L. A., and C. M. Buenos Aires, pp. 1–10.

Nealen, A. (2004): An as-short-as-possible introduction to the least squares, weighted least squares and moving least squares methods for scattered data approximation and interpolation. In: *URL: http://www. nealen. com/projects* 130(150), p. 25.

Nguyen, V. P., C. T. Nguyen, T. Rabczuk, and S. Natarajan (2017): On a family of convected particle domain interpolations in the material point method. In: *Finite Elements in Analysis and Design* 126, pp. 50–64. DOI: `https://doi.org/10.1016/j.finel.2016.11.007`. URL: `http://www.sciencedirect.com/science/article/pii/S0168874X16301020`.

Nguyen, V. P., T. Rabczuk, S. Bordas, and M. Duflot (2008): Meshless methods: a review and computer implementation aspects. In: *Mathematics and computers in simulation* 79(3), pp. 763–813.

Niemunis, A. (2003): Extended hypoplastic models for soils. Habilitation. Schriftreihe des Instituts für Grundbau und Bodenmechanik der Ruhr-Universität Bochum, Heft 34.

Parker, S. G. (2002): A component-based architecture for parallel multi-physics PDE simulation. In: *International Conference on Computational Science*. Springer, pp. 719–734.

Richtyer, R. (1950): A method for the numerical calculation of hydrodynamical shocks. In: *J. Appl. Phys* 21, pp. 380–385.

Sabel, M., C. Sator, and R. Müller (2014): A particle finite element method for machining simulations. In: *Computational Mechanics* 54(1), pp. 123–131.

Sadeghirad, A., R. M. Brannon, and J. Burghardt (2011): A convected particle domain interpolation technique to extend applicability of the material point method for problems involving massive deformations. In: *International Journal for Numerical Methods in Engineering* 86(12), pp. 1435–1456. DOI: `10.1002/nme.3110`. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.3110`.

Sadeghirad, A., R. Brannon, and J. Guilkey (2013): Second-order convected particle domain interpolation (CPDI2) with enrichment for weak discontinuities at material interfaces. In: *International Journal for Numerical Methods in Engineering* 95(11), pp. 928–

952. DOI: `10.1002/nme.4526`. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.4526`.

Schanz, M. and A.-D. Cheng (2000): Transient wave propagation in a one-dimensional poroelastic column. In: *Acta Mechanica* 145(1-4), pp. 1–18.

Schiller, L. (1933): A drag coefficient correlation. In: *Zeit. Ver. Deutsch. Ing.* 77, pp. 318–320.

Shepard, D. (1968): A two-dimensional interpolation function for irregularly-spaced data. In: *Proceedings of the 1968 23rd ACM national conference*, pp. 517–524.

Simon, B., O. Zienkiewicz, and D. Paul (1984): An analytical solution for the transient response of saturated porous elastic solids. In: *International Journal for Numerical and analytical methods in Geomechanics* 8(4), pp. 381–398.

Steffen, M., R. M. Kirby, and M. Berzins (2008): Analysis and reduction of quadrature errors in the material point method (MPM). In: *International journal for numerical methods in engineering* 76(6), pp. 922–948.

Sulsky, D., Z. Chen, and H. Schreyer (1994): A particle method for history-dependent materials. In: *Computer Methods in Applied Mechanics and Engineering* 118(1), pp. 179–196. DOI: `https://doi.org/10.1016/0045-7825(94)90112-0`. URL: `http://www.sciencedirect.com/science/article/pii/0045782594901120`.

Sulsky, D. and M. Gong (2016): Improving the material-point method. In: *Innovative numerical approaches for multi-field and multi-scale problems.* Springer, pp. 217–240.

Sulsky, D., S.-J. Zhou, and H. L. Schreyer (1995): Application of a particle-in-cell method to solid mechanics. In: *Computer physics communications* 87(1-2), pp. 236–252.

Tran, Q., E. Wobbes, W. T. Sołowski, M. Möller, and C. Vuik (2019): Moving least squares reconstruction for B-spline Material Point Method. English. In: Proceedings os the 2nd International Conference on the Material Point Method for Modelling Sil-Water-Structure Interaction; p. 7. URL: `http://urn.fi/URN:NBN:fi:aalto-201901301464`.

van Dalen, K. (2013): *Multi-component accoustic characterization of porous media.* Springer Theses – Recognizing Outstanding Ph. D. Research. Springer.

Verruijt, A. (2010): *An intorduction to soil dynamics.* Vol. 24. Theory and Applications of Transport in Porous Media. Springer.

Wallstedt and Guilkey (2008): An evaluation of explicit time integration schemes for use with the generalized interpolation material point method. In: *Journal of Computational Physics* 227(22), pp. 9628–9642.

Wallstedt and Guilkey (2011): A weighted least squares particle-in-cell method for solid mechanics. In: *International journal for numerical methods in engineering* 85(13), pp. 1687–1704.

Wang, Q., W. Zhou, Y. Cheng, G. Ma, X. Chang, Y. Miao, and E. Chen (2018): Regularized moving least-square method and regularized improved interpolating moving least-square method with nonsingular moment matrices. In: *Applied Mathematics and Computation* 325, pp. 120–145.

Wendland, H. (1995): Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. In: *Advances in computational Mathematics* 4(1), pp. 389–396.

Wendland, H. (2004): *Scattered Data Approximation.* Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press. DOI: `10.1017/CBO9780511617539`.

Więckowski, Z. (2013): Enhancement of the material point method for fluid-structure interaction and erosion. In: *Research project report, Deltares.*

Wobbes, E., M. Möller, V. Galavi, and C. Vuik (2019): Conservative Taylor least squares reconstruction with application to material point methods. In: *International Journal for Numerical Methods in Engineering* 117(3), pp. 271–290.

Yamaguchi, Y., S. Takase, S. Moriguchi, and K. Terada (2020): Solid–liquid coupled material point method for simulation of ground collapse with fluidization. In: *Computational Particle Mechanics* 7(2), pp. 209–223.

Zhang, X. Ma, and P. T. Giguere (2011): Material point method enhanced by modified gradient of shape function. In: *Journal of Computational Physics* 230(16), pp. 6379–6398.

Zhang, X., Z. Chen, and Y. Liu (2017): *The material point method : a continuum-based particle method for extreme loading cases.* First edition. Elsevier and Tsinghua University Press computational mechanics series. Amsterdam, Beijing: Elsevier Tsinghua University Press.

Zienkiewicz, O., A. Chan, M. Pastor, B. Schrefler, and T. Shiomi (Jan. 1999): *Computational Geomechanics With Special Reference to Earthquake Engineering.*

Zienkiewicz, O., C. Chang, and P. Bettess (1980): Drained, undrained, consolidating and dynamic behaviour assumptions in soils. In: *Geotechnique* 30(4), pp. 385–395.

Appendix A

Notation

Nomenclature

Abbreviations

MPM	Material Point Method
PIC	Particle in Cell Method
FEM	Finite Element Method
MMS	Manufactured Solution Method
MP	Material Point
GIMP	Generalized Interpolation Material Point
BSMPM	B-Spline MPM
CPDI	Convected Particle Domain Interpolation
SVD	singular value decomposition
LS	Least Squares Approxiamtion
WLS	Weighted Least Squares Approxiamtion
MLS	Moving Least Squares Approxiamtion
iMPM	Improved MPM
riMPM	Regularized Improved MPM
GP	Gauss Point
MMS	Method of Manufactured Solutions

Symbols

$\mathbf{F}(\mathbf{x},t)$	deformation gradient
$\mathbf{R}$	rotational (rigid motion) part of $\mathbf{F}$
$\mathbf{U}$	deformational part of $\mathbf{F}$
$\boldsymbol{\epsilon}$	strain tensor
ϵ_v	volumatric strain tensor
$\mathbf{X} = (X, Y)$	Lagrangian coordinated
$\mathbf{x} = (x, y)$	Eulerian coordinates
$\mathbf{v}$	general velocity vector
$\mathbf{v}_s$	solid velocity vector

$\mathbf{v}_f$	fluid velocity vector
$\mathbf{u}_s$	solid displacement vector
$\mathbf{u}_f$	fluid displacement vector
Ω_s	volume fraction of the solid material
Ω_v	volume fraction of void
$\boldsymbol{\sigma}$	Cauchy stress tensor
$\boldsymbol{\sigma}'$	effective stress tensor
$\boldsymbol{\sigma}_f$	partial stress fluid
$\boldsymbol{\sigma}_s$	partial stress solid
$\mathbf{P}$	first Piola-Kirchhoff stress tensor
$J = \det(\mathbf{F})$	Jacobian
$\mathbf{t}$	traction vector
$\mathbf{n}$	normal unit vector
$\boldsymbol{\tau}$	deviatoric stress
$\boldsymbol{\tau}_f$	viscous stress fluid
p	general pressure
p_f	pore fluid pressure
p_s	pressure solid
$\mathbf{I}$	identity tensor
n	porosity
n_s	volume fraction solid
n_f	volume fraction fluid
ρ	general density
ρ_f	density fluid
ρ_s	density solid
$\overline{\rho} = (1 - n)\rho$	partial density
$\overline{\rho}_\alpha$	partial density of $\alpha-$phase
g	gravitational acceleration
E	general Young's
E_s	Young's modulus for soil
m_v	compressibility of solid
ν	Poisson ratio
C_f	compressibility of fluid
$\mathbf{f}_d$	Drag-Force with buoyancy
$\overline{\mathbf{f}}_d$	Drag-Force without buoyancy
κ	intrinsic permeability
μ	dynamic viscosity of the fluid
α	Biot coefficient
$\mathbf{b}$	body force
$\mathbf{M}$	mass matrix
$\mathbf{f}_{ext}$	external forces
$\mathbf{f}_{int}$	internal forces
$\boldsymbol{\sigma_p}$	MP stress
$\hat{\mathbf{a}}$	nodal acceleration vector
m_p	MP mass

V_p	MP volume
Ω_p	MP domain
$N_i(x)$	linear nodal shape function
$\chi(\mathbf{x})$	MP shape function
$\phi_{ip}(\mathbf{x})$	MP weighting function
$\phi_{ip}^{x}(\mathbf{x})$	MP weighting function in x-direction
$\phi_{ip}^{y}(\mathbf{x})$	MP weighting function in y-direction
$\phi_{ip,x}^{x}(x)$	derivative of MP weighting function
$\phi_{i,k}(x)$	k-th order B-Spline at i-th knot
$\mathbf{v}_i^t$	nodal velocity
n_p	number of MP
n_n	number of nodes
$\mathcal{N}$	nullspace
$\mathrm{u_h}$	LS/MLS approximation
$\mathbf{a}(\mathbf{x})$	LS coefficient vector
$\mathbf{p}(\mathbf{x})$	polynomial basis
$\mathbf{P}$	polynomial matrix of evaluated values
$\mathbf{W}(\mathbf{x})$	weighting function matrix
$\mathbf{M}(\mathbf{x})$	moment matrix
$\mathbf{B}(\mathbf{x})$	B-matrix for MLS
$\phi_i^{\mathbf{MLS}}(\mathbf{x})$	MLS shape funtion of i-th data point
$w(\mathbf{x}-\tilde{\mathbf{x}}_i)$	weighting function
λ	regularization parameter
Ω^s	soil domain
Ω^f	pore fluid domain
δu	test function
$\mathbf{M}_s^{lump}$	lumped soil mass matrix
$\mathbf{M}_f^{lump}$	lumped pore fluid mass matrix
N_{sp}	number of soil MP
N_{fp}	number of fluid MP
m_{sp}	mass of p-th soil MP
m_{fp}	mass of p-th fluid MP
$\mathbf{x}_{sp}$	position of p-th soil MP
$\mathbf{x}_{fp}$	position of p-th fluid MP
N_n	number of nodes
κ_{fp}	intrinsic permeability at p-th fluid MP
$\hat{\mathbf{a}}_s$	nodal soil acceleration
$\hat{\mathbf{a}}_f$	nodal fluid acceleration
$\mathbf{f}_s^{ext}$	nodal body force soil
$\mathbf{f}_s^{int}$	nodal internal force soil
$\mathbf{f}_f^{ext}$	nodal body force fluid
$\mathbf{f}_f^{int}$	nodal internal force fluid
$(\hat{\mathbf{m}}\mathbf{v})_s^{t+\Delta t}$	nodal momentum soil
$(\hat{\mathbf{m}}\mathbf{v})_f^{t+\Delta t}$	nodal momentum fluid
$V_{s/fp}^{t+\Delta t}$	soil/fluid MP volume

L_D	local damping coefficient
ξ ,η	parent coordinates
$\mathbf{N}$	shape function array for Standard MPM
$\mathbf{dN}$	derivative array for Standard MPM
$\hat{\mathbf{v}}_f$	nodal fluid velocity vector
$\hat{\mathbf{v}}_s$	nodal solid velocity vector

Appendix B

Analytical Solution for $\alpha = 1$

The characteristic polynomial of $\mathbf{A}$ for the assumption $\alpha = 1$ becomes

$$p^4 + \left(-\frac{s+\kappa\, s+\gamma\, s^2+\kappa\, s^2+\gamma\,\kappa\, s^2-2\,\beta\,\kappa^2\, s^2-\beta\,\kappa\, s^2}{\kappa-\kappa^3}\right) p^2 + \frac{\gamma\, s^4+\kappa\, s^3+s^3-\beta^2\, s^4+\gamma\,\kappa\, s^4-\beta^2\,\kappa\, s^4}{\kappa-\kappa^3} \tag{B.1}$$

The resulting eigenvalues are:

$$p_1 = \sqrt{\frac{s+\kappa\, s+\gamma\, s^2+\kappa\, s^2-\sigma_3+\sigma_5-\sigma_7-\sigma_6}{\sigma_4}} \tag{B.2a}$$

$$p_2 = \sqrt{\frac{s+\kappa\, s+\gamma\, s^2+\kappa\, s^2+\sigma_3+\sigma_5-\sigma_7-\sigma_6}{\sigma_4}} \tag{B.2b}$$

$$p_3 = -p_1 \tag{B.2c}$$

$$p_4 = -p_2 \tag{B.2d}$$

with the parameters

$$\begin{aligned}\sigma_3 = s\,\sqrt{}&\overline{5\,\beta^2\,\kappa^2\,s^2+4\,\beta^2\,\kappa\,s^2-4\,\beta\,\gamma\,\kappa^3\,s^2-6\,\beta\,\gamma\,\kappa^2\,s^2-2\,\beta\,\gamma\,\kappa\,s^2-4\,\beta\,\kappa^3\,s^2-4\,\beta\,\kappa^3\,s}\\ &\overline{-\sigma_7-6\,\beta\,\kappa^2\,s-2\,\beta\,\kappa\,s+\gamma^2\,\kappa^2\,s^2+2\,\gamma^2\,\kappa\,s^2+\gamma^2\,s^2+4\,\gamma\,\kappa^4\,s^2+4\,\gamma\,\kappa^3\,s^2-2\,\gamma\,\kappa^2\,s^2}\\ &\overline{+4\,\gamma\,\kappa\,s+2\,\gamma\,s+4\,\kappa^4\,s+4\,\kappa^3\,s+\kappa^2\,s^2-2\,\kappa^2\,s+\kappa^2-2\,\kappa\,s+2\,\kappa+1}\\ &\overline{+2\,\gamma\,\kappa^2\,s-2\,\gamma\,\kappa\,s^2}\end{aligned}$$

$$\sigma_4 = 2\,\left(\kappa-\kappa^3\right)$$

$$\sigma_5 = \gamma\,\kappa\,s^2$$

$$\sigma_6 = \beta\,\kappa\,s^2$$

$$\sigma_7 = 2\,\beta\,\kappa^2\,s^2$$

and the eigenvectors

$\mathbf{v}_1 =$

$$\begin{pmatrix} \frac{\left(\kappa-\kappa^2\right)\sigma_2\,{\sigma_5}^{3/2}}{\sigma_4\,\sigma_1} - \frac{\sqrt{\sigma_5}\left(\beta^2\,\kappa^3\,s^2+\beta^2\,\kappa^2\,s^2+\sigma_3-3\,\beta\,\gamma\,\kappa^2\,s^2-\sigma_8-\beta\,\kappa^2\,s^2-3\,\beta\,\kappa^2\,s-2\,\beta\,\kappa\,s+\gamma^2\,\kappa\,s^2+\sigma_9+\gamma\,\kappa^3\,s^2+2\,\gamma\,\kappa\,s+2\,\gamma\,s+\kappa^3\,s+\kappa+1\right)}{s^2\,(\beta-\kappa)\left(s+\kappa\,s+\gamma\,s^2-\beta^2\,s^2+\sigma_{10}-\sigma_3\right)} \\ \frac{\sqrt{\sigma_5}\left(\kappa+\gamma\,s+\kappa\,s-\beta\,\kappa\,s+\gamma\,\kappa\,s-2\,\beta\,\kappa^2\,s+1\right)}{\gamma\,s^3+\kappa\,s^2+s^2-\beta^2\,s^3+\gamma\,\kappa\,s^3-\beta^2\,\kappa\,s^3} - \frac{\left(\kappa-\kappa^2\right)\sigma_5{}^{3/2}}{\sigma_1} \\ \frac{\left(\kappa-\kappa^2\right)\sigma_6}{2\left(\beta\,s^2-\kappa\,s^2\right)\left(\kappa-\kappa^3\right)} - \frac{\sigma_2}{\sigma_4} \\ 1 \end{pmatrix}$$

where

$\sigma_1 = -\beta^2\,s^4 + \gamma\,s^4 + s^3$

$\sigma_2 = \gamma\,s - \beta\,\kappa\,s + 1$

$\sigma_3 = \beta^2\,\kappa\,s^2$

$\sigma_4 = \beta\,s - \kappa\,s$

$\sigma_5 = \frac{\sigma_6}{2\left(\kappa-\kappa^3\right)}$

$$\sigma_6 = s + \kappa\,s + \gamma\,s^2 + \kappa\,s^2 - s\,\sqrt{5\,\beta^2\,\kappa^2\,s^2 + 4\,\beta^2\,\kappa\,s^2 - 4\,\beta\,\gamma\,\kappa^3\,s^2 - 6\,\beta\,\gamma\,\kappa^2\,s^2 - \sigma_8 - 4\,\beta\,\kappa^3\,s^2 - 4\,\beta\,\kappa^3\,s - \sigma_7 - 6\,\beta\,\kappa^2\,s - 2\,\beta\,\kappa\,s + \gamma^2\,\kappa^2\,s^2 + 2\,\gamma^2\,\kappa\,s^2 - 2\,\gamma\,\kappa^2\,s^2 + 2\,\gamma\,\kappa^2\,s - 2\,\gamma\,\kappa\,s^2 + 4\,\gamma\,\kappa\,s + 2\,\gamma\,s + 4\,\kappa^4\,s + 4\,\kappa^3\,s + \kappa^2\,s^2 - 2\,\kappa^2\,s + \kappa^2 - 2\,\kappa\,s + 2\,\kappa + 1 + \sigma_9 + 4\,\gamma\,\kappa^4\,s^2 + 4\,\gamma\,\kappa^3\,s^2} + \sigma_{10} - \sigma_7 - \beta\,\kappa\,s^2$$

$\sigma_7 = 2\,\beta\,\kappa^2\,s^2$

$\sigma_8 = 2\,\beta\,\gamma\,\kappa\,s^2$

$\sigma_9 = \gamma^2\,s^2$

$\sigma_{10} = \gamma\,\kappa\,s^2$

$\mathbf{v}_2 =$

$$\begin{pmatrix} \frac{\left(\kappa-\kappa^2\right)\sigma_2\,{\sigma_5}^{3/2}}{\sigma_4\,\sigma_1} - \frac{\sqrt{\sigma_5}\left(\beta^2\,\kappa^3\,s^2+\beta^2\,\kappa^2\,s^2+\sigma_3-3\,\beta\,\gamma\,\kappa^2\,s^2-\sigma_8-\beta\,\kappa^2\,s^2-3\,\beta\,\kappa^2\,s-2\,\beta\,\kappa\,s+\gamma^2\,\kappa\,s^2+\sigma_9+\gamma\,\kappa^3\,s^2+2\,\gamma\,\kappa\,s+2\,\gamma\,s+\kappa^3\,s+\kappa+1\right)}{s^2\,(\beta-\kappa)\left(s+\kappa\,s+\gamma\,s^2-\beta^2\,s^2+\sigma_{10}-\sigma_3\right)} \\ \frac{\sqrt{\sigma_5}\left(\kappa+\gamma\,s+\kappa\,s-\beta\,\kappa\,s+\gamma\,\kappa\,s-2\,\beta\,\kappa^2\,s+1\right)}{\gamma\,s^3+\kappa\,s^2+s^2-\beta^2\,s^3+\gamma\,\kappa\,s^3-\beta^2\,\kappa\,s^3} - \frac{\left(\kappa-\kappa^2\right)\sigma_5{}^{3/2}}{\sigma_1} \\ \frac{\left(\kappa-\kappa^2\right)\sigma_6}{2\left(\beta\,s^2-\kappa\,s^2\right)\left(\kappa-\kappa^3\right)} - \frac{\sigma_2}{\sigma_4} \\ 1 \end{pmatrix}$$

where

$\sigma_1 = -\beta^2\,s^4 + \gamma\,s^4 + s^3$

$\sigma_2 = \gamma\,s - \beta\,\kappa\,s + 1$

$\sigma_3 = \beta^2\,\kappa\,s^2$

$\sigma_4 = \beta\,s - \kappa\,s$

$\sigma_5 = \frac{\sigma_6}{2\left(\kappa-\kappa^3\right)}$

$$\sigma_6 = s + \kappa\,s + \gamma\,s^2 + \kappa\,s^2 + s\,\sqrt{5\,\beta^2\,\kappa^2\,s^2 + 4\,\beta^2\,\kappa\,s^2 - 4\,\beta\,\gamma\,\kappa^3\,s^2 - 6\,\beta\,\gamma\,\kappa^2\,s^2 - \sigma_8 - 4\,\beta\,\kappa^3\,s^2 - 4\,\beta\,\kappa^3\,s - \sigma_7 - 6\,\beta\,\kappa^2\,s - 2\,\beta\,\kappa\,s + \gamma^2\,\kappa^2\,s^2 + 2\,\gamma^2\,\kappa\,s^2 + \sigma_9 + 4\,\gamma\,\kappa^4\,s^2 + 4\,\gamma\,\kappa^3\,s^2 - 2\,\gamma\,\kappa^2\,s^2 + 2\,\gamma\,\kappa^2\,s - 2\,\gamma\,\kappa\,s^2 + 4\,\gamma\,\kappa\,s + 2\,\gamma\,s + 4\,\kappa^4\,s + 4\,\kappa^3\,s + \kappa^2\,s^2 - 2\,\kappa^2\,s + \kappa^2 - 2\,\kappa\,s + 2\,\kappa + 1} + \sigma_{10} - \sigma_7 - \beta\,\kappa\,s^2$$

$\sigma_7 = 2\,\beta\,\kappa^2\,s^2$

$\sigma_8 = 2\,\beta\,\gamma\,\kappa\,s^2$

$\sigma_9 = \gamma^2\,s^2$

$\sigma_{10} = \gamma\,\kappa\,s^2$

$\mathbf{v}_3 =$

$$\begin{pmatrix} \frac{\sqrt{\sigma_5}\,\left(\beta^2\,\kappa^3\,s^2+\beta^2\,\kappa^2\,s^2+\sigma_3-3\,\beta\,\gamma\,\kappa^2\,s^2-\sigma_8-\beta\,\kappa^2\,s^2-3\,\beta\,\kappa^2\,s-2\,\beta\,\kappa\,s+\gamma^2\,\kappa\,s^2+\sigma_9+\gamma\,\kappa^3\,s^2+2\,\gamma\,\kappa\,s+2\,\gamma\,s+\kappa^3\,s+\kappa+1\right)}{s^2\,(\beta-\kappa)\,\left(s+\kappa\,s+\gamma\,s^2-\beta^2\,s^2+\sigma_{10}-\sigma_3\right)} - \frac{\left(\kappa-\kappa^2\right)\,\sigma_2\,{\sigma_5}^{3/2}}{\sigma_4\,\sigma_1} \\ \frac{\left(\kappa-\kappa^2\right)\,{\sigma_5}^{3/2}}{\sigma_1} - \frac{\sqrt{\sigma_5}\,\left(\kappa+\gamma\,s+\kappa\,s-\beta\,\kappa\,s+\gamma\,\kappa\,s-2\,\beta\,\kappa^2\,s+1\right)}{\gamma\,s^3+\kappa\,s^2+s^2-\beta^2\,s^3+\gamma\,\kappa\,s^3-\beta^2\,\kappa\,s^3} \\ \frac{\left(\kappa-\kappa^2\right)\,\sigma_6}{2\,\left(\beta\,s^2-\kappa\,s^2\right)\,\left(\kappa-\kappa^3\right)} - \frac{\sigma_2}{\sigma_4} \\ 1 \end{pmatrix}$$

where

$\sigma_1 = -\beta^2\,s^4 + \gamma\,s^4 + s^3$

$\sigma_2 = \gamma\,s - \beta\,\kappa\,s + 1$

$\sigma_3 = \beta^2\,\kappa\,s^2$

$\sigma_4 = \beta\,s - \kappa\,s$

$\sigma_5 = \frac{\sigma_6}{2\,\left(\kappa-\kappa^3\right)}$

$$\begin{aligned}\sigma_6 = {} & s + \kappa\,s + \gamma\,s^2 + \kappa\,s^2 \\ & -s\,\sqrt{\begin{aligned}&5\,\beta^2\,\kappa^2\,s^2 + 4\,\beta^2\,\kappa\,s^2 - 4\,\beta\,\gamma\,\kappa^3\,s^2 - 6\,\beta\,\gamma\,\kappa^2\,s^2 - \sigma_8 - 4\,\beta\,\kappa^3\,s^2 - 4\,\beta\,\kappa^3\,s - \sigma_7 - 6\,\beta\,\kappa^2\,s - 2\,\beta\,\kappa\,s + \gamma^2\,\kappa^2\,s^2 + 2\,\gamma^2\,\kappa\,s^2 \\ &+\sigma_9 + 4\,\gamma\,\kappa^4\,s^2 + 4\,\gamma\,\kappa^3\,s^2 - 2\,\gamma\,\kappa^2\,s^2 + 2\,\gamma\,\kappa^2\,s - 2\,\gamma\,\kappa\,s^2 + 4\,\gamma\,\kappa\,s + 2\,\gamma\,s + 4\,\kappa^4\,s + 4\,\kappa^3\,s + \kappa^2\,s^2 - 2\,\kappa^2\,s + \kappa^2 - 2\,\kappa\,s + 2\,\kappa + 1\end{aligned}} \\ & +\sigma_{10} - \sigma_7 - \beta\,\kappa\,s^2\end{aligned}$$

$\sigma_7 = 2\,\beta\,\kappa^2\,s^2$

$\sigma_8 = 2\,\beta\,\gamma\,\kappa\,s^2$

$\sigma_9 = \gamma^2\,s^2$

$\sigma_{10} = \gamma\,\kappa\,s^2$

$\mathbf{v}_4 =$

$$\begin{pmatrix} \frac{\sqrt{\sigma_5}\,\left(\beta^2\,\kappa^3\,s^2+\beta^2\,\kappa^2\,s^2+\sigma_3-3\,\beta\,\gamma\,\kappa^2\,s^2-\sigma_8-\beta\,\kappa^2\,s^2-3\,\beta\,\kappa^2\,s-2\,\beta\,\kappa\,s+\gamma^2\,\kappa\,s^2+\sigma_9+\gamma\,\kappa^3\,s^2+2\,\gamma\,\kappa\,s+2\,\gamma\,s+\kappa^3\,s+\kappa+1\right)}{s^2\,(\beta-\kappa)\,\left(s+\kappa\,s+\gamma\,s^2-\beta^2\,s^2+\sigma_{10}-\sigma_3\right)} - \frac{\left(\kappa-\kappa^2\right)\,\sigma_2\,{\sigma_5}^{3/2}}{\sigma_4\,\sigma_1} \\ \frac{\left(\kappa-\kappa^2\right)\,{\sigma_5}^{3/2}}{\sigma_1} - \frac{\sqrt{\sigma_5}\,\left(\kappa+\gamma\,s+\kappa\,s-\beta\,\kappa\,s+\gamma\,\kappa\,s-2\,\beta\,\kappa^2\,s+1\right)}{\gamma\,s^3+\kappa\,s^2+s^2-\beta^2\,s^3+\gamma\,\kappa\,s^3-\beta^2\,\kappa\,s^3} \\ \frac{\left(\kappa-\kappa^2\right)\,\sigma_6}{2\,\left(\beta\,s^2-\kappa\,s^2\right)\,\left(\kappa-\kappa^3\right)} - \frac{\sigma_2}{\sigma_4} \\ 1 \end{pmatrix}$$

where

$\sigma_1 = -\beta^2\,s^4 + \gamma\,s^4 + s^3$

$\sigma_2 = \gamma\,s - \beta\,\kappa\,s + 1$

$\sigma_3 = \beta^2\,\kappa\,s^2$

$\sigma_4 = \beta\,s - \kappa\,s$

$\sigma_5 = \frac{\sigma_6}{2\,\left(\kappa-\kappa^3\right)}$

$$\begin{aligned}\sigma_6 = {} & s + \kappa\,s + \gamma\,s^2 + \kappa\,s^2 \\ & +s\,\sqrt{\begin{aligned}&5\,\beta^2\,\kappa^2\,s^2 + 4\,\beta^2\,\kappa\,s^2 - 4\,\beta\,\gamma\,\kappa^3\,s^2 - 6\,\beta\,\gamma\,\kappa^2\,s^2 - \sigma_8 - 4\,\beta\,\kappa^3\,s^2 - 4\,\beta\,\kappa^3\,s - \sigma_7 - 6\,\beta\,\kappa^2\,s - 2\,\beta\,\kappa\,s + \gamma^2\,\kappa^2\,s^2 + 2\,\gamma^2\,\kappa\,s^2 \\ &+\sigma_9 + 4\,\gamma\,\kappa^4\,s^2 + 4\,\gamma\,\kappa^3\,s^2 - 2\,\gamma\,\kappa^2\,s^2 + 2\,\gamma\,\kappa^2\,s - 2\,\gamma\,\kappa\,s^2 + 4\,\gamma\,\kappa\,s + 2\,\gamma\,s + 4\,\kappa^4\,s + 4\,\kappa^3\,s + \kappa^2\,s^2 - 2\,\kappa^2\,s + \kappa^2 - 2\,\kappa\,s + 2\,\kappa + 1\end{aligned}} \\ & +\sigma_{10} - \sigma_7 - \beta\,\kappa\,s^2\end{aligned}$$

$\sigma_7 = 2\,\beta\,\kappa^2\,s^2$

$\sigma_8 = 2\,\beta\,\gamma\,\kappa\,s^2$

$\sigma_9 = \gamma^2\,s^2$

$\sigma_{10} = \gamma\,\kappa\,s^2$

Appendix C

Parallel GPU code for 2P-MPM wave propagation

Init-Params

```
clear all
%%%%%%%%%%%%%%%%%%%%%
% Parameter Initialization
%%%%%%%%%%%%%%%%%%%%%

%%% Material Parameter %%%
E = 5e9; nu = 0.0; rho=2650.0; Kw = 2e9; rhoW = 1000;
np=0.4; dyvis = 1e-6;  k = 1e-13; c = sqrt(E/rho);
D=E/((1+nu)*(1-2*nu))*[1-nu nu 0; nu 1-nu 0;0 0 (1-2*nu)/2];
%%% Local Damping %%%
LD = 0.0;
%%% Bulk Viscosity %%%
C1 = 0.42;     %Linear
C2 = 1.2;      %Quadratic
%%%%%%%%%%%%%%%%%%%%%
% Copy to GPU
%%%%%%%%%%%%%%%%%%%%%
E = gpuArray(E); nu = gpuArray(nu); rho = gpuArray(rho); LD = ...
    gpuArray(LD);
Kw = gpuArray(Kw); rhoW = gpuArray(rhoW); np = gpuArray(np);
dyvis = gpuArray(dyvis); k = gpuArray(k); C1 = gpuArray(C1);
C2 = gpuArray(C2); D = gpuArray(D);
```

Init-Node-Geometry

```
%%%%%%%%%%%%%%%%%%%%%
% Mesh Initialization
%%%%%%%%%%%%%%%%%%%%%
H = 1.1; % height of column
W = 0.0001; % width of column
```

```
dx=0.0001; dy=dx; % element's edge length
ndx=(W/dx); ndy=(H/dx); % number of elements per direction
nn=(ndx+1)*(ndy+1); % total number of the grid nodeIDs
ne=ndx*ndy; % total number of the background elements
x=zeros(2,nn);
connect=zeros(4,ne); % grid connectectivities

%%% node coordinates %%%
x(1,1:nn/2)=0; x(1,nn/2+1:end)=dx;
x(2,1:nn/2)=H-(0:1:nn/2-1)*dx; x(2,nn/2+1:end)=H-(0:1:nn/2-1)*dx;

%%% connectectivity %%%
for j=1:ndx
    for i=1:ndy
        elemID=(j-1)*dy+i;
        connect(4,elemID)=elemID+(j-1); ...
            connect(1,elemID)=connect(4,elemID)+1;
        connect(2,elemID)=connect(1,elemID)+dy+1; ...
            connect(3,elemID)=connect(2,elemID)-1;
    end
end
%%%%%%%%%%%%%%%%%%%%%
% Time Integration
%%%%%%%%%%%%%%%%%%%%%
nT = 8.0071e+03; % number of time steps
RR = (1-np)*rho+np*rhoW;
CC1 = sqrt((E/RR)+(Kw/RR/np)); % undrained wave speed
CC2 = sqrt(np/rhoW/((1-np)*(1/E)+np*(1/Kw))); % drained wave speed
dt = 0.2*dx/max(CC1,CC2);
```

Init-MP-Geometry

```
%%%%%%%%%%%%%%%%%%%%%
% MP-coordinates
%%%%%%%%%%%%%%%%%%%%%

%%% SOLID %%%
nps=10000;xps=zeros(2,nps)
xps(1,1:nps)=dx/2;
xps(2,1:nps)=1-dx/2:-dx:dx/2;

%%% LIQUID %%%
npW=10000;xpW=zeros(2,npW)
xpW(1,1:npW)=dx/2;
xpW(2,1:npW)=1-dx/2:-dx:dx/2;
```

Init-Arrays

```
%%%%%%%%%%% %%%%%%%%
vol0s=dx*dx*ones(1,nps); % initial particle volumes
```

```
vol0W=dx*dx*ones(1,nps); % initial particle volumes
r0s=rho*ones(1,nps); % initial particle densities
Vps=zeros(2,nps); % Initial particles velocities
r0W=rhoW*ones(1,nps); % initial particle densities
VpW=zeros(2,nps); % Initial particles velocities

%%%%%%%%%%%%%% SOLID %%%%%%%%
sigma=zeros(3,nps); % initial particle stresses
sigmaBV=zeros(3,nps);
F=zeros(2,2,nps); % initial particle deformation gradient
F(1,1,:)=ones(1,1,nps); F(2,2,:)=ones(1,1,nps);
%%%%%%%%%%%%%% LIQUID %%%%%%%%
WPressure=zeros(3,npW);
WPressureBV=zeros(3,npW);
FW=zeros(2,2,npW); % initial particle deformation gradient
FW(1,1,:)=ones(1,1,npW); FW(2,2,:)=ones(1,1,npW);

Vns=zeros(2,nn); % initial assignment to the grid node velocities
VnW=zeros(2,nn);
Ans=zeros(2,nn); % initial assignment to the grid node acceleration
AnW=zeros(2,nn);
```

Init-BC + Copy to GPU

```
%%%%%%%%%%%%%% Randbedingungen %%%%%%%%
bcx1 = find(x(1,:)==0); bcx2 = find(x(1,:)==dx);
bcx = [bcx1 bcx2];
bcy1 = find(x(2,:)==0); bcy2 = find(x(2,:)==H);
bcy = [bcy1 bcy2];

%%%%%%%%%%%%%% Copy to GPU %%%%%%%%
c = gpuArray(c); bcx = gpuArray(bcx); bcy = gpuArray(bcy);
ndy = gpuArray(ndy); ne = gpuArray(ne);
nps = gpuArray(nps); npW = gpuArray(npW);
dx = gpuArray(dx); dy = gpuArray(dy); nn = gpuArray(nn);
x = gpuArray(x); connect = gpuArray(connect);
xps = gpuArray(xps); xpW = gpuArray(xps);
bcx = gpuArray(bcx); bcy = gpuArray(bcy);

Vps=gpuArray(zeros(2,nps)); VpW=gpuArray(zeros(2,nps));
Vns=gpuArray(zeros(2,nn)); VnW=gpuArray(zeros(2,nn));

vol0s=gpuArray(vol0s); vol0W=gpuArray(vol0W);
r0s=gpuArray(r0s); r0W=gpuArray(r0W);

sigma=gpuArray(zeros(3,nps)); sigmaBV=gpuArray(zeros(3,nps));
WPressure=gpuArray(zeros(3,npW)); WPressureBV=gpuArray(zeros(3,npW));

F = gpuArray(F); FW = gpuArray(FW); n=1;
n=gpuArray(n);  nT=gpuArray(nT);
```

BEGIN LOOP OVER TIMESTEPS

```
for n=1:nT
```

Arrays resetted to zero each timestep

```
%%%%%%%%%%%%%% Copy to GPU %%%%%%%
    fints=gpuArray(zeros(2,nn));  % internal force vector solid
    fexts=gpuArray(zeros(2,nn));  % external force vector solid
    moms=gpuArray(zeros(2,nn));   % linear momentum vector solid

    fintW=gpuArray(zeros(2,nn)); % internal force vector fluid
    fextW=gpuArray(zeros(2,nn)); % external force vector fluid
    momW=gpuArray(zeros(2,nn));  % linear momentum vector fluid

    dragf = gpuArray(zeros(2,nn));

    ms=gpuArray(zeros(1,nn));
    mW=gpuArray(zeros(1,nn));
```

Update particle volume and density for the current time step

```
    %%%%%%%%%%%%%% SOLID %%%%%%%
    detFs = reshape((F(1,1,:).*F(2,2,:))-(F(1,2,:).*F(2,1,:)),1,nps);
    vols = detFs.*vol0s; rps=(1./detFs).*r0s;
    %%%%%%%%%%%%%% LIQUID %%%%%%%%
    detFW = reshape((FW(1,1,:).*FW(2,2,:))-(FW(1,2,:).*FW(2,1,:)),1,npW);
    volW = detFW.*vol0W; rpW=(1./detFW).*r0W;
```

P2N Solid

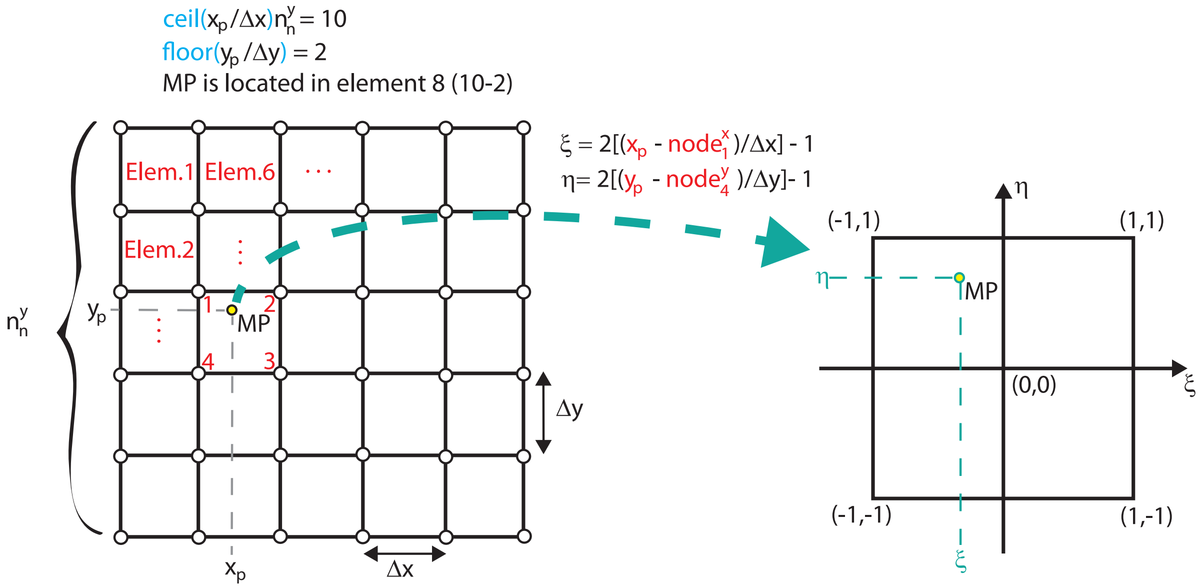
Transfer Information from solid **MPs** to **nodeIDs** to calculate:

- mass: **ms(1,nn)**
- momentum: **moms(2,nn)**
- external force: **fexts(2,nn)**

Elements and nodeIDs occupied by solid MPs

```
       elemIDs=(ceil(xps(1,:)/dx))*ndy-floor(xps(2,:)/dx);
       nodeIDs=connect(:,elemIDs);
```

Shape functions for solid MPs



$$\varphi(:,n_p) = \tfrac{1}{4}\begin{pmatrix} (1-\xi_p)(1-\zeta_p) \\ (1+\xi_p)(1-\zeta_p) \\ (1+\xi_p)(1+\zeta_p) \\ (1-\xi_p)(1+\zeta_p) \end{pmatrix}$$

```
    locxps = [2*(xps(1,:)-x(1,nodeIDs(1,:)))/dx-1
    2*(xps(2,:)-x(2,nodeIDs(1,:)))/dy-1];
    phis = 1/4*[(1-locxps(1,:)).*(1-locxps(2,:)); ...
        (1+locxps(1,:)).*(1-locxps(2,:)); ...
    (1+locxps(1,:)).*(1+locxps(2,:)); (1-locxps(1,:)).*(1+locxps(2,:))];
```

Nodal mass and momentum for solid MPs

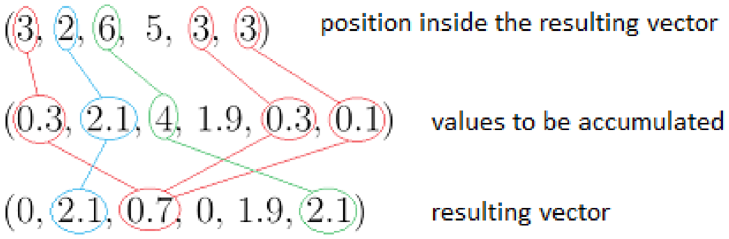
mass portion of particle n_p at each node

$$am(:,n_p) = \varphi(:,n_p)\cdot\rho_p\cdot vols$$

momentum portion of particle n_p at each node in x- and y-direction

$$A(:,n_p) = am\cdot V_p(1,n_p) \quad \text{(x-direction)} \qquad\qquad B(:,n_p) = am\cdot V_p(2,n_p) \quad \text{(y-direction)}$$

The Matlab function **accumarray** , accumulates the elements of a vector(real) according to another vector(integer).



nodeIDs consists 4 (for Standard MPM) related nodeIDs for each MP

AA accumulates all mass portions of material points in vector **am** at the nodeIDs

QQ1/QQ2 accumulates all momentum portions of material points in vector **A/B** in **x/y-direction** at the nodeIDs

In **moms** the nodal momentum of **QQ1/QQ2** is filled with zeros at the end to reach a vector length of **nn** (number of nodeIDs)

```
     am = (1-np)*phis(1:4,:).*(rps(:).*vols(:))';
     A = am.*Vps(1,:); B = am.*Vps(2,:);
     AA = accumarray(nodeIDs(1:4*nps)',am(1:4*nps)');
     ms(1,:) =[AA;zeros(nn-length(AA),1)];
     QQ1 = accumarray(nodeIDs(1:4*nps)',A(1:4*nps)');
     QQ2 = accumarray(nodeIDs(1:4*nps)',B(1:4*nps)');
     moms(1,:) = [QQ1;zeros(nn-length(QQ1),1)];
     moms(2,:) = [QQ2;zeros(nn-length(QQ2),1)];
```

External Force Solid

```
      fnode = find(x(2,:)==1); fexts(2,fnode)= -dx/2;
```

P2N Liquid

Transfer Information from liquid **MPs** to **nodeIDW** to calculate:

- mass: **mW(1,nnW)**
- momentum: **momW(2,nnW)**
- external force: **fextW(2,nn)**

Elements and nodeIDs occupied by liquid MPs

```
     elemIDW = (ceil(xpW(1,:)/dx))*ndy-floor(xpW(2,:)/dx);
     nodeIDW=connect(:,elemIDW);
```

Shape functions for liquid MPs

```
locxpW=[2*(xpW(1,:)-x(1,nodeIDW(1,:)))/dx-1
2*(xpW(2,:)-x(2,nodeIDW(1,:)))/dx-1];

phiW=1/4*[(1-locxpW(1,:)).*(1-locxpW(2,:)); ...
    (1+locxpW(1,:)).*(1-locxpW(2,:)); ...
(1+locxpW(1,:)).*(1+locxpW(2,:)); (1-locxpW(1,:)).*(1+locxpW(2,:))];
```

Nodal mass and momentum for liquid MPs

```
amW = np*phiW(1:4,:).*(rpW(:).*volW(:))';

AW = amW.*VpW(1,:);
BW = amW.*VpW(2,:);

AAW = accumarray(nodeIDW(1:4*npW)',amW(1:4*npW)');

mW(1,:) = [AAW;zeros(nn-length(AAW),1)];

QQW1 = accumarray(nodeIDW(1:4*npW)',AW(1:4*npW)');
QQW2 = accumarray(nodeIDW(1:4*npW)',BW(1:4*npW)');

momW(1,:) = [QQW1;zeros(nn-length(QQW1),1)];
momW(2,:) = [QQW2;zeros(nn-length(QQW2),1)];
```

External Force Liquid

```
fextW(1,:)= 0;  fextW(2,:)= 0;
```

Boundary Conditions for momentum

```
moms(1,bcx)=0; moms(2,bcy)=0; momW(1,bcx)=0; momW(2,bcy)=0;
```

Internal Forces

Calculate the internal forces at the grid nodeIDs

Derivative Shape Functions Solid

$$\nabla\varphi(:,n_p) = \begin{pmatrix} \frac{\partial\varphi_1}{\partial\xi} \\ \frac{\partial\varphi_1}{\partial\zeta} \\ \frac{\partial\varphi_1}{\partial\xi} \\ \frac{\partial\varphi_1}{\partial\zeta} \\ \frac{\partial\varphi_2}{\partial\xi} \\ \frac{\partial\varphi_2}{\partial\zeta} \\ \frac{\partial\varphi_2}{\partial\xi} \\ \frac{\partial\varphi_2}{\partial\zeta} \\ \frac{\partial\varphi_3}{\partial\xi} \\ \frac{\partial\varphi_3}{\partial\zeta} \\ \frac{\partial\varphi_3}{\partial\xi} \\ \frac{\partial\varphi_3}{\partial\zeta} \\ \frac{\partial\varphi_4}{\partial\xi} \\ \frac{\partial\varphi_4}{\partial\zeta} \\ \frac{\partial\varphi_4}{\partial\xi} \\ \frac{\partial\varphi_4}{\partial\zeta} \end{pmatrix} \quad \text{(16x1-vector)} = \quad \frac{2}{4\cdot dx}\cdot \begin{pmatrix} -(1-\zeta_p) \\ -(1-\xi_p) \\ -(1-\zeta_p) \\ -(1-\xi_p) \\ (1-\zeta_p) \\ -(1+\xi_p) \\ (1-\zeta_p) \\ -(1+\xi_p) \\ (1+\zeta_p) \\ (1+\xi_p) \\ (1+\zeta_p) \\ (1+\xi_p) \\ -(1+\zeta_p) \\ (1-\xi_p) \\ -(1-\zeta_p) \\ (1-\xi_p) \end{pmatrix}$$

```
      dphis = 1/4*(2/dx)*...
         [-(1-locxps(2,:)); -(1-locxps(1,:)); -(1-locxps(2,:)); ...
            -(1-locxps(1,:)); ...
            1-locxps(2,:); -(1+locxps(1,:));    1-locxps(2,:); ...
               -(1+locxps(1,:)); ...
            1+locxps(2,:);    1+locxps(1,:);    1+locxps(2,:);    ...
               1+locxps(1,:); ...
          -(1+locxps(2,:));    1-locxps(1,:); -(1+locxps(2,:));    ...
             1-locxps(1,:)];
```

Divergence of Sigma and SigmaBV for internal forces solid

$$\mathbf{Asigma}(:\,,n_p) = \begin{pmatrix} \sigma^p_{xx} \\ \sigma^p_{yy} \\ \sigma^p_{xy} \\ \sigma^p_{xy} \end{pmatrix} \cdot Vol_p\,, \qquad \mathbf{ASig}(:\,,n_p) = \begin{pmatrix} \text{Asigma}(:\,,n_p) \\ \text{Asigma}(:\,,n_p) \\ \text{Asigma}(:\,,n_p) \\ \text{Asigma}(:\,,n_p) \end{pmatrix} = \begin{pmatrix} \sigma^p_{xx} \\ \sigma^p_{yy} \\ \sigma^p_{xy} \\ \sigma^p_{xy} \\ \sigma^p_{xx} \\ \sigma^p_{yy} \\ \sigma^p_{xy} \\ \sigma^p_{xy} \\ \sigma^p_{xx} \\ \sigma^p_{xx} \\ \sigma^p_{xy} \\ \sigma^p_{xy} \\ \sigma^p_{xx} \\ \sigma^p_{yy} \\ \sigma^p_{xy} \\ \sigma^p_{xy} \end{pmatrix} \cdot$$

Vol_p

AsigmaBV and **ASigBV** are claculated accordingly

$$\mathbf{Bsigma}(:,n_p) = \nabla\varphi(:,n_p)\cdot(\mathbf{ASig}{+}\mathbf{ASigBV}) = \begin{pmatrix} \sigma^p_{xx}\frac{\partial\varphi_1}{\partial\xi} \\ \sigma^p_{yy}\frac{\partial\varphi_1}{\partial\zeta} \\ \sigma^p_{xy}\frac{\partial\varphi_1}{\partial\xi} \\ \sigma^p_{xy}\frac{\partial\varphi_1}{\partial\zeta} \\ \sigma^p_{xx}\frac{\partial\varphi_2}{\partial\xi} \\ \sigma^p_{yy}\frac{\partial\varphi_2}{\partial\zeta} \\ \sigma^p_{xy}\frac{\partial\varphi_2}{\partial\xi} \\ \sigma^p_{xy}\frac{\partial\varphi_2}{\partial\zeta} \\ \sigma^p_{xx}\frac{\partial\varphi_3}{\partial\xi} \\ \sigma^p_{yy}\frac{\partial\varphi_3}{\partial\zeta} \\ \sigma^p_{xy}\frac{\partial\varphi_3}{\partial\xi} \\ \sigma^p_{xy}\frac{\partial\varphi_3}{\partial\zeta} \\ \sigma^p_{xx}\frac{\partial\varphi_4}{\partial\xi} \\ \sigma^p_{yy}\frac{\partial\varphi_4}{\partial\zeta} \\ \sigma^p_{xy}\frac{\partial\varphi_4}{\partial\xi} \\ \sigma^p_{xy}\frac{\partial\varphi_4}{\partial\zeta} \end{pmatrix}\cdot Vol_p$$

$$\mathbf{BSigx}(:,n_p) = \begin{pmatrix} \sigma^p_{xx}\frac{\partial\varphi_1}{\partial\xi} + \sigma^p_{xy}\frac{\partial\varphi_1}{\partial\xi} \\ \sigma^p_{xx}\frac{\partial\varphi_2}{\partial\xi} + \sigma^p_{xy}\frac{\partial\varphi_2}{\partial\xi} \\ \sigma^p_{xx}\frac{\partial\varphi_3}{\partial\xi} + \sigma^p_{xy}\frac{\partial\varphi_3}{\partial\xi} \\ \sigma^p_{xx}\frac{\partial\varphi_4}{\partial\xi} + \sigma^p_{xy}\frac{\partial\varphi_4}{\partial\xi} \end{pmatrix}\cdot Vol_p$$

$$\mathbf{BSigy}(:,n_p) = \begin{pmatrix} \sigma^p_{yy}\frac{\partial\varphi_1}{\partial\zeta} + \sigma^p_{xy}\frac{\partial\varphi_1}{\partial\zeta} \\ \sigma^p_{yy}\frac{\partial\varphi_2}{\partial\zeta} + \sigma^p_{xy}\frac{\partial\varphi_2}{\partial\zeta} \\ \sigma^p_{yy}\frac{\partial\varphi_3}{\partial\zeta} + \sigma^p_{xy}\frac{\partial\varphi_3}{\partial\zeta} \\ \sigma^p_{yy}\frac{\partial\varphi_4}{\partial\zeta} + \sigma^p_{xy}\frac{\partial\varphi_4}{\partial\zeta} \end{pmatrix}\cdot Vol_p$$

FF1 and **FF2** accumulate the **BSigx** and **BSigy** at the corresponding nodeIDs. **fints** equals extended **FF1** and **FF2** to the size of **nnodes**.

```
        Asigma = [sigma; sigma(3,:)].*vols;
        ASig = [ Asigma ; Asigma ; Asigma; Asigma];

        AsigmaBV = [sigmaBV; sigmaBV(3,:)].*vols;
        ASigBV = [ AsigmaBV ; AsigmaBV ; AsigmaBV; AsigmaBV];

        Bsigma = dphis.*(ASig+ASigBV);
```

```

    BSigx = [Bsigma(1,:) +  Bsigma(3,:);    Bsigma(5,:) +  ...
       Bsigma(7,:); ...
            Bsigma(9,:) +  Bsigma(11,:);   Bsigma(13,:) + ...
               Bsigma(15,:)];

    BSigy = [Bsigma(2,:) +  Bsigma(4,:);    Bsigma(6,:) +  ...
       Bsigma(8,:); ...
            Bsigma(10,:) + Bsigma(12,:);   Bsigma(14,:) + ...
               Bsigma(16,:)];


    FF1 = accumarray(nodeIDs(1:4*nps)',BSigx(1:4*nps)');
    FF2 = accumarray(nodeIDs(1:4*nps)',BSigy(1:4*nps)');

    fints(1,:) = [FF1;zeros(nn-length(FF1),1)];

    fints(2,:) = [FF2;zeros(nn-length(FF2),1)];
```

Derivative Shape Functions Liquid

```
    dphiW = 1/4*(2/dx)*...
       [-(1-locxpW(2,:)); -(1-locxpW(1,:)); -(1-locxpW(2,:)); ...
          -(1-locxpW(1,:)); ...
           1-locxpW(2,:); -(1+locxpW(1,:));    1-locxpW(2,:); ...
              -(1+locxpW(1,:)); ...
           1+locxpW(2,:);    1+locxpW(1,:);    1+locxpW(2,:);    ...
              1+locxpW(1,:); ...
        -(1+locxpW(2,:));    1-locxpW(1,:); -(1+locxpW(2,:));    ...
           1-locxpW(1,:)];
```

`Gradient of p` for internal force liquid

```
    AWPressureW = [WPressure; WPressure(3,:)].*volW;
    AWPressure= [ AWPressureW ; AWPressureW ; AWPressureW; ...
       AWPressureW];
    AWPressureBV = [WPressureBV; WPressureBV(3,:)].*volW;
    AWPBV = [ AWPressureBV ; AWPressureBV ; AWPressureBV; ...
       AWPressureBV];
    BWPressure= dphiW.*(AWPressure+AWPBV);
    BWPx = [BWPressure(1,:);    BWPressure(5,:); ...
            BWPressure(9,:);   BWPressure(13,:)];
    BWPy = [BWPressure(2,:);    BWPressure(6,:); ...
            BWPressure(10,:);   BWPressure(14,:)];
    FFW1 =  accumarray(nodeIDW(1:4*npW)',BWPx(1:4*npW)');
    FFW2 =  accumarray(nodeIDW(1:4*npW)',BWPy(1:4*npW)');
    fintW(1,:) = [FFW1;zeros(nn-length(FFW1),1)];
    fintW(2,:) = [FFW2;zeros(nn-length(FFW2),1)];
```

Drag-Force

```
    nodeI = intersect(unique(nodeIDs),unique(nodeIDW)); VI=dx^2/2;
    dragf(:,nodeI) = (np^2*dyvis/k)*(VnW(:,nodeI)-Vns(:,nodeI))*VI;
```

Calculate momentum rate

```
    dmom = fexts - fints + (1-np)*fintW - LD*sign(Vns).*abs(fexts - ...
       fints + (1-np)*fintW) + dragf;
    dmomW = fextW + np*fintW - LD*sign(VnW).*abs(fextW + np*fintW)- ...
       dragf;
```

Boundary Conditions for momentum rate

```
    dmom(1,bcx)=0; dmom(2,bcy)=0;
    dmomW(1,bcx)=0; dmomW(2,bcy)=0;
```

Time Integration

```
    moms = moms + dmom*dt;  momW = momW + dmomW*dt;
```

N2P

update the particle velocities and locations

Nodal and MP `velocities/accelerations` solid

$$\mathrm{V}_\mathrm{x}(:,n_p) = \begin{pmatrix} V_{x_p}^1 \\ V_{x_p}^2 \\ V_{x_p}^3 \\ V_{x_p}^4 \end{pmatrix}, \mathrm{V}_\mathrm{y}(:,n_p) = \begin{pmatrix} V_{y_p}^1 \\ V_{y_p}^2 \\ V_{y_p}^3 \\ V_{y_p}^4 \end{pmatrix}, \mathrm{A}_\mathrm{x}(:,n_p) = \begin{pmatrix} A_{x_p}^1 \\ A_{x_p}^2 \\ A_{x_p}^3 \\ A_{x_p}^4 \end{pmatrix}, \mathrm{V}_\mathrm{y}(:,n_p) = \begin{pmatrix} A_{y_p}^1 \\ A_{y_p}^2 \\ A_{y_p}^3 \\ A_{y_p}^4 \end{pmatrix}$$

$V_p^{t+1} = V_p^t + \sum_{i=1}^{4}[\varphi_i(x_p^t)A_i^t]dt$

$x_p^{t+1} = x_p^t + \sum_{i=1}^{4}[\varphi_i(x_p^t)V_i^t]dt$

```
    Vns = moms./ms; Ans = dmom./ms;
```

```
    Vx = reshape(Vns(1,nodeIDs),4,nps);
    Vy = reshape(Vns(2,nodeIDs),4,nps);
    Ax = reshape(Ans(1,nodeIDs),4,nps);
    Ay = reshape(Ans(2,nodeIDs),4,nps);
    Vps(1,:) = Vps(1,:) + dot(phis,Ax)*dt;
    Vps(2,:) = Vps(2,:) + dot(phis,Ay)*dt;
    xps(1,:) = xps(1,:) + dot(phis,Vx)*dt;
    xps(2,:) = xps(2,:) + dot(phis,Vy)*dt;
```

Nodal and MP velocities/accelerations liquid

SEE NODAL AND MP VELOCITIES/ACCELERATIONS FOR SOLID

```
    VnW = momW./mW; AnW = dmomW./mW;
```

```
    VxW = reshape(VnW(1,nodeIDW),4,npW);
    VyW = reshape(VnW(2,nodeIDW),4,npW);
    AxW = reshape(AgnW(1,nodeIDW),4,npW);
    AyW = reshape(AgnW(2,nodeIDW),4,npW);

    VpW(1,:) = VpW(1,:) + dot(phiW,AxW)*dt;
    VpW(2,:) = VpW(2,:) + dot(phiW,AyW)*dt;

    xpW(1,:) = xpW(1,:) + dot(phiW,VxW)*dt;
    xpW(2,:) = xpW(2,:) + dot(phiW,VyW)*dt;
```

Update Deformation/Velocity Gradients for solid/liquid MPs

$$\mathrm{dNx}(:,n_p) = \begin{pmatrix} \frac{\partial\varphi_1}{\partial\xi_p} \\ \frac{\partial\varphi_2}{\partial\xi_p} \\ \frac{\partial\varphi_3}{\partial\xi_p} \\ \frac{\partial\varphi_4}{\partial\xi_p} \end{pmatrix}, \qquad \mathrm{dNy}(:,n_p) = \begin{pmatrix} \frac{\partial\varphi_1}{\partial\zeta_p} \\ \frac{\partial\varphi_2}{\partial\zeta_p} \\ \frac{\partial\varphi_3}{\partial\zeta_p} \\ \frac{\partial\varphi_4}{\partial\zeta_p} \end{pmatrix}$$

$$\mathrm{A_{incrF}}(:,n_p) = \begin{pmatrix} \mathrm{dNx}\cdot\mathrm{V_x}\mathrm{dNy}\cdot\mathrm{V_x} \\ \mathrm{dNx}\cdot\mathrm{V_y}\mathrm{dNy}\cdot\mathrm{V_y} \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{4}\frac{\partial\varphi_i}{\partial\xi_p}\mathrm{V^i_{x_p}} & \sum_{i=1}^{4}\frac{\partial\varphi_i}{\partial\zeta_p}\mathrm{V^i_{x_p}} \\ \sum_{i=1}^{4}\frac{\partial\varphi_i}{\partial\xi_p}\mathrm{V^i_{y_p}} & \sum_{i=1}^{4}\frac{\partial\varphi_i}{\partial\zeta_p}\mathrm{V^i_{y_p}} \end{pmatrix} \quad \text{(Velocity Gradient)}$$

$$\mathrm{F_{incr}}(:,\mathrm{n_p}) = \Delta\mathrm{F}(:,\mathrm{n_p}) = \mathrm{I_2} + \mathrm{A_{incrF}}\cdot\mathrm{dt}$$

$$\dot{\mathrm{F}}(:,\mathrm{n_p}) = \left(\Delta\mathrm{F}(:,\mathrm{n_p})\cdot\mathrm{F}(:,\mathrm{n_p}) - \mathrm{F}(:,\mathrm{n_p})\right)/\mathrm{dt}$$

$$\mathrm{F}(:,\mathrm{n_p}) = \Delta\mathrm{F}(:,\mathrm{n_p})\cdot\mathrm{F}(:,\mathrm{n_p}) \quad \text{(Deformation Gradient)}$$

```
        IN(1,1,:)=gpuArray(ones(1,1,nps));
        IN(2,2,:)=gpuArray(ones(1,1,nps));
        INW(1,1,:)=gpuArray(ones(1,1,npW));
        INW(2,2,:)=gpuArray(ones(1,1,npW));
        dNx = [dphis(1,:); dphis(5,:); dphis(9,:); dphis(13,:)];
        dNy = [dphis(2,:); dphis(6,:); dphis(10,:); dphis(14,:)];
        AincrF(1,1,:) = dot(dNx,Vx);
        AincrF(2,2,:) = dot(dNy,Vy);
        AincrF(1,2,:) = dot(dNy,Vx);
        AincrF(2,1,:) = dot(dNx,Vy);
        incrF = IN + AincrF*dt;
        dNxW = [dphiW(1,:); dphiW(5,:); dphiW(9,:); dphiW(13,:)];
        dNyW = [dphiW(2,:); dphiW(6,:); dphiW(10,:); dphiW(14,:)];
        DF = (incrF.*F-F)/dt;
        F=incrF.*F;
```

Update pore pressure/div(v-w)/deformation gradient

$\Omega_{\mathrm{MP_S}} =$ set of all solid elements

$\Omega_{\mathrm{MP_W}} =$ set of all liquid elements

$\mathrm{Lia}(:,\mathrm{n_{pW}}) = 1 \quad if \quad n_{pW} \in \Omega_{MP_S}, \quad else \quad \mathrm{Lia}(:,\mathrm{n_{pW}}) = 0$

$\mathrm{Lic}(:,\mathrm{n_{pW}}) = 1 \quad if \quad n_{pS} \in \Omega_{MP_W}, \quad else \quad \mathrm{Lia}(:,\mathrm{n_{pW}}) = 0$

MPfs(Lia) $\hat{=}$ indices of liquid MPs with $\mathrm{n_{pW}} \in \Omega_{\mathrm{MP_S}}$

MPsf(Lic) $\hat{=}$ indices of solid MPs with $\mathrm{n_p} \in \Omega_{\mathrm{MP_W}}$

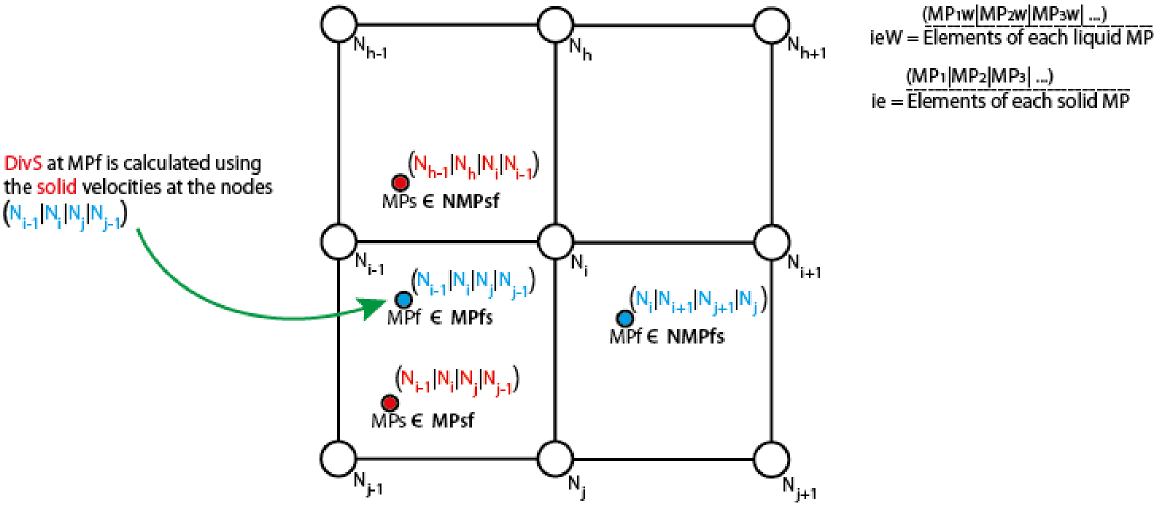
NMPfs($\sim$ Lia) $\hat{=}$ indices of liquid MPs with $\mathrm{n_p} \notin \Omega_{\mathrm{MP_W}}$

NMPsf($\sim$ Lic) $\hat{=}$ indices of solid MPs with $\mathrm{n_p} \notin \Omega_{\mathrm{MP_S}}$

Vsfx and Vsfy $\hat{=}$ solid velocities at nodeIDs of an element containing both kinds of MPs

Divergence of solid velocities with respect to gradients of liquid shape functions:
$\mathrm{DivS}(:,\mathrm{n_{pW}}) = \mathrm{dNxW} \cdot \mathrm{Vsfx} + \mathrm{dNyW} \cdot \mathrm{Vsfy} = \sum_{i=1}^{4} \frac{\partial \varphi_{iW}}{\partial \xi_{pW}} \mathrm{Vsf^i_{x_p}} + \sum_{i=1}^{4} \frac{\partial \varphi_{iW}}{\partial \zeta_{pW}} \mathrm{Vsf^i_{y_p}}$

Divergence of liquid velocities with respect to gradients of liquid shape functions:
$\mathrm{DivW}(:,\mathrm{n_{pW}}) = \mathrm{dNxW} \cdot \mathrm{VxW} + \mathrm{dNyW} \cdot \mathrm{VyW} = \sum_{i=1}^{4} \frac{\partial \varphi_{iW}}{\partial \xi_{pW}} \mathrm{V^i_{x_{pW}}} + \sum_{i=1}^{4} \frac{\partial \varphi_{iW}}{\partial \zeta_{pW}} \mathrm{V^i_{y_{pW}}}$



DEpsW(: ,n_{pW}) volumetric strain for **all** liquid MPs

WPressure(: ,n_{pW}) pore pressure for **all** liquid MPs

```
Lia = ismember(elemIDW,elemIDs);
MPfs=find(Lia); NMPfs=find(¬Lia);
Lic= ismember(elemIDs,elemIDW);
MPsf=find(Lic); NMPsf=find(¬Lic);
Vsfx = reshape(Vns(1,nodeIDW),4,npW);
Vsfy = reshape(Vns(2,nodeIDW),4,npW);
DivS = dot(dNxW(:,MPfs),Vsfx(:,MPfs)) + ...
    dot(dNyW(:,MPfs),Vsfy(:,MPfs));
DivW = dot(dNxW(:,MPfs),VxW(:,MPfs)) + ...
    dot(dNyW(:,MPfs),VyW(:,MPfs));
DEpsW(MPfs) = (np*DivW + (1-np)*DivS)*dt;
WPressure(:,MPfs) = WPressure(:,MPfs) - [1 1 ...
    0]'.*DEpsW(MPfs)*Kw/np;
DEpsW2=zeros(2,2,npW);
DEpsW2(1,1,:)=DEpsW(:); DEpsW2(2,2,:)=DEpsW(:);
FW = INW + DEpsW2;
```

Linear Elastic Model

```
Eps = 0.5*(F+permute(F,[2 1 3]))-IN;
epsE = [Eps(1,1,:);Eps(2,2,:);Eps(1,2,:)];
DepsE = 0.5*(permute(AincrF,[2 1 3])+AincrF);
D=E/((1+nu)*(1-2*nu))*[1-nu nu 0; nu 1-nu 0;0 0 (1-2*nu)/2];
sigma = D*squeeze(epsE);
```

Bulk Viscosity

```
        BVI=find(DepsV < 0);
        sigmaBV(1,BVI)= -(C2*rho*(dx*DepsV(BVI)).^2 ...
        - C1*rho*dx*c*DepsV(BVI));
        sigmaBV(2,BVI)= -(C2*rho*(dx*DepsV(BVI)).^2 ...
        - C1*rho*dx*c*DepsV(BVI));
```

Visualize Results

```
 plot(1-xps(2,:),reshape(-sigma(2,:),1,nps),'r--o',1-xpW(2,:), ..
 reshape(WPressure(2,:),1,nps),'b--o',1-xps(2,:), ...
 -reshape(sigma(2,:),1,nps)-reshape(-WPressure(2,:),1,nps),'black--')
 xlabel('Distance [m]') ylabel('Absolute Pressure and Stress [Pa]')
 drawnow
```

END LOOP OVER TIMESTEPS

```
end
```