MODELING ELECTROCHEMISTRY PROBLEMS USING THE MATERIAL POINT AND FINITE VOLUME METHODS

by

Christopher Eric Gritton

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Computing

School of Computing

The University of Utah

August 2024

The University of Utah Graduate School

STATEMENT OF DISSERTATION APPROVAL

The dissertation of **Christopher Eric Gritton**
has been approved by the following supervisory committee members:

Martin Berzins ,	Chair(s)	**18 June 2024** Date Approved
Robert Michael Kirby II ,	Member	**18 June 2024** Date Approved
Christopher R. Johnson ,	Member	**18 June 2024** Date Approved
Dmitro Bedrov ,	Member	**18 June 2024** Date Approved
James Edward Guilkey ,	Member	**18 June 2024** Date Approved

by **Mary W. Hall** , Chair/Dean of
the Department/College/School of **Computing**
and by **Daryl P. Butt** , Dean of The Graduate School.

ABSTRACT

The demand for batteries is continually growing. The use of personal electronic devices such as smart phones and tablets, the increase in renewable energy sources, and the continuing trend to replace fossil-fuel-based personal transport with electric vehicles all require the use of batteries. With this increase in demand is the need for lighter and more efficient batteries. Computational modeling plays a key role in the development of new designs and the exploration of novel of materials, which in turn leads to improvements in battery efficiency and weight. The modeling of a battery cell involves capturing the physics of electrochemical diffusion, deformation, and electrostatics. The multiphysics nature, as well as the difference in time scales at which the physical processes occur, make this problem difficult. The numerical method used to model the multiphysics of an battery cell undergoing a charge/disscharge cycle is the Material Point Method (MPM). This work focuses on two key areas: improving upon current approaches to MPM and applying MPM and the finite volume method to battery modeling. The improvements in MPM have been achieved in the following ways: 1) developing approaches by which information is transferred back and forth between particles and the grid through the use of nullspace filters and corrected derivatives; 2) analyzing different time integration schemes that can be used to address the multiscale nature of modeling the coupled diffusion/deformation problem using MPM; and 3) the use MPM in the reference configuration to solve diffusion type problems. The application of MPM to modeling battery problems has been demonstrated through the simulation of a silicon electrode within a lithium ion battery undergoing multiple charge/discharge cycles. The finite volume vethod is applied to modeling a full battery cell during the charging phase, which involves the coupled physics of electrochemical diffusion and electrostatics.

For God and my family.

CONTENTS

NOTATION AND SYMBOLS

$\mathbf{I}$	Identitiy matrix
$\mathbf{F}$	Deformation gradient
$\boldsymbol{\epsilon}$	Strain tensor
$\mathbf{E}$	Green-Lagrange strain tensor
$\mathbf{e}$	Euler-Almansi strain tensor
$\boldsymbol{\sigma}$	Cauchy stress tensor
$\mathbf{P}$	First Piola-Kirchhoff stress tensor
$\mathbb{S}$	Compliance tensor
$\mathbb{C}$	Stiffness tensor

CHAPTER 1

INTRODUCTION

Batteries are ubiquitous in today's society. They are used in personal eletronic devices such as smartphones, tablets, and laptops. Electronic vehicle (EV) now account for 1% of vehicles in current use in the US and 7.4% of vehicles sold in 2023 [1]. The use of batteries for home energy storage will continue to grow. In 2020 almost 4% of US homes had solar power as means of power generation, and by 2030 the usage of home solar power is expected to triple [2]. In the 2023 the US saw a 35% year-over-year increase in added solar capacity [3]. By 2027 it is projected that nearly 30% of all solar installs will be paired with battery storage [4].

Batteries are also becoming important in national defense. Today's warfighter now carries, in addition to their standard tactical gear, a variety of mission critical electronic devices. Thoughout history soldiers have needed to carry a large quantity of gear over long distances for combat. For a Roman legionaire in 100 B.C., the average weight of their gear was 80 lbs, a Union soldier in the Civil War carried an average of 60 lbs, and a soldier in WWII carried an average of 75 lbs [5]. Today's soldier now carries between 70-100lbs, and for many the number is even higher [6]. When looking at all the pieces of gear that contribute to the total weight, batteries account for 20 lbs of the total [6].

Innovation in battery technologies and materials, such as litium-ion batteries (LIB), has allowed for increases in charge capacity and decreases in weight and form factor. These improvements have contributed to the wide spread adoption of items ranging from smartphones to EVs. As was stated in the *Economist*, "the lithium-ion battery is the technology of our time" [7].

The process of developing novel battery designs and materials requires new insights that stem from both physical and computational experimentation. For a researcher in electrochemistry and electrochemical cell design, computational tools are a valuable asset.

Through the use of computer-aided engineering (CAE), scientists and engineers can explore new ideas in material design and electrochemical cell development [8, 9]. The suite of computational modeling capabilities of CAE tools ranges in scale from the atomic all the way up to the systems level [10].

Within that scale, ranging from atomistic to the system level, there are software tools, mathematical models, and numerical methods capable of modeling physical phenomena at the continuum scale. Numerical methods such as finite difference methods (FDM), finite element methods (FEM), and finite volume methods (FVM) are all popular numerical methods for modeling continuum level phenomena. All three methods utilize a grid or mesh to discretize the problem domain. Another family of numerical methods utilize particles to represent the problem domain [11]. Within the family of particle methods there are two types. The first type utilizes only the particles in solving the model equations. Examples of this type are smooth particle hydrodynamics (SPH) [12] and reproducing kernel particle methods (RKPM) [13]. The second type utilizes both particles and a mesh. Examples of this type are particle-in-cell (PIC) [14], fluid implicit particle (FLIP) [15], and the material point method (MPM) [16].

1.1 The Material Point Method and Recent Advances

The material point method (MPM) has been the subject of continual research and development since its introduction in the mid 1990s [16, 17, 18, 19]. One of the key advantages of MPM is its ability to model large deformation problems, which becomes important in modeling electrodes in an electrochemical cells. The charge capacity of a cell comes as a result of the quantity of ions an electrode can absorb and release during a charge discharge cycle. As the ratio of ions to electrode material goes up, so does the volume expansion of electrode itself. Large changes in volume of the electrode lead to large deformations [20]. For this reason MPM becomes a viable method for use in continuum scale modeling of electrochemcial materials. A more complete discussion on electrode materials and the volume expansion of electrodes during a charge/discharge cycle is presented in Chapter 2 and Chapter 7.

A more complete discusion of MPM, including a discussion of the original algorithm;

improvements upon the original method, and a historical discussion covering some of the key particle methods that preceeded MPM's development, is presented in Chapter 3. At this point a brief discussion of the more recent advancements in MPM is presented.

In 2008 the original use of B-splines in MPM was proposed by Steffen et al. [21], and in recent years, is has continued to be a topic of research. Koster et al. [22] have developed an approach on unstructured meshes that incorporates mesh refinement and utilizes Powell-Sabin spines. Chen and Li [23] developed a novel C^1 coursened shape function constructed using B-splines. Their approach shows improvements in stability and accuracy when modeling nonlinear heterogenous materials. Sadeghirad [24] combined B-splines with the convected particle domain interpolation technique (CPDI) to create the BSCPDI variant of MPM. Isogeometric analysis was developed by Hughes et al. [25] to combine the Non-Uniform Rational Basis Splines (NURBS) used to model geometry in CAD programs with FEM. This approach has been adapted by Moutsanidis et al. [26] for MPM.

Recent research has also been applied to resolving stability issues in MPM. One such issue is associated with volumetric locking, which comes when modeling a nearly incompressible material that resists volumetric change, resulting in a overly stiff reaction in the model. For FEM one solution is to use higher order basis functions while reducing the number of integration points in the element, but this is not possible for MPM because of the movement of particles, which act as integration points, through the background grid [27]. A variety of recent approaches have been proposed to resolve the volumetric locking issue. Wang et al. [28] adopted the use of the $\overline{F}$-patch method coupled with simplex elements to overcome volumetric locking. For each element the $\overline{F}$-patch method creates a "patch" of the surrounding elements which constitutes the initial volume. Particle positions are then updated, and the new patch is the set of elements based on the new positions of the particles. The deformation gradient increment is then scaled based on the ratio the new and initial patch volumes. Xie et al. [29] proposed an implicit B-spline MPM that uses an $\overline{F}$ method to resolve instabilities from volumetric locking. Their approach differs from the previous approach in that the incremental deformation gradient is scaled based on a ratio of an averaged incremental deformation gradient Jacobian and the incremental deformation gradient. The averaging is done by mapping the incremental Jacobian from particle to nodes and back.

Along with improvements in the method itself, MPM is also seeing wider adoption for a variety of applications. One area in particular is in the geotechnical field. In 2024 a quick search of journal articles involving MPM will produce a large variety of articles related the geotechnical field. For example Di Perna et al. [30] used MPM for modeling landslide-structure interactions, Zhao et al. [31] introduced a two-phase for modeling debris flows using MPM, and Antonello [32] demonstrated the use of MPM to model landslide kinematics in senstive clay. On top of journal articles, an entire book by Fern et al. [33] has been published that focuses on the use of MPM in the geotechnical field.

MPM has been applied to modeling additive manufacturing. For example Yildizdag [34] presented an MPM based framework for simulating the extrusion process used in 3D printing, which uses a level set method for tracking the free surface boundary conditions. Lian et al. [35] proposed a multiphysics version of MPM for metal additive manufacturing. There approach is designed to model heat transfer, fluid flow, and phase change.

The use of MPM in robotics simulations and robotics design is an interesting application for MPM. Bianchi et al. [36] created a bioinspired aquatic snake robot. As part of the design process, they created a computer model of the robot and used MLS-MPM to model the fluid. The computer simulation was used for refining control parameters. Davy et al. [37] have introduced a framework for magnetic soft robots that use MPM. Magnetic soft robots are made of a soft magnetic material that can be controlled by a magnetic field. MPM is well suited for modeling these types of materials and has an advantage over FEM in that it handles self-collisions by design, which is important for these types of robots that fold in on themselves. Cochevelou et al. [38] proposed a method for optimizing the topology of a soft robot under the senarios of walking, walking up stairs, and avoiding obstacles. They use MLS-MPM for simulating the soft robot material.

Over the last seven years, two survey papers [18, 19] have also been published covering the key developments in MPM over the last 25 years. Three books have been published on the subject of MPM. Two of the books covers the theory, implementation, variations of the method, and applications in a general sense [39, 40], while the third, as was noted earlier, covers the use of MPM as applied to the geotechnical field [33].

1.2 Motivation

Even with all the advancements that have been made in battery technologies, there is much room for improvement. As was noted previously, the modern warfighter carries a variety of electronic devices that all require a battery in some form or another. Modern soldiers operate in extreme environments where their equipment is subject to large variations in temperature; exposure to dust, water, and other debris; and physical abuse. As a result battery designs need to be improved so that they can meet the required demands of the operating environment. Improvements also need to be made in increasing charge capacity and reduction of weight. Doing so will reduce the total weight the soldier is required to carry.

The motivation of this dissertation is to provide improvements in numerical models that will aid scientists and engineers in the development of new and improved battery designs and will allow for the exploration of novel battery materials.

1.3 Thesis Statement

The material point and finite volume methods are shown to be valuable methods for modeling electrochemistry materials. An examination of the issues associated with the stability and accuracy MPM is presented along with solutions to such issues.

The focus of this research has been on the use of the material point method (MPM) and the finite volume method (FVM) for modeling electrochemical materials and to examine the issues associated with the stability and accuracy of MPM generally and when applied to electrochemical problems.

1.4 Contributions

The focus of this research has been on understanding the issues associated with the stability and accuracy of MPM, using MPM for multiphysics problem involving diffusion, the application of MPM to modeling electrodes, and examining the use of FVM for modeling a full electrochemical cell. The belief is that the analysis, insights, and solutions presented here can be of use to researchers of MPM generally, while demonstrating the use of MPM and FVM to electochemical problems specifically. Here are a number of specific contributions:

1. In MPM there is a mismatch in dimensionality between particles and nodes that

results in a nullspace. When values are mapped from nodes to particles, numerical noise can be mapped to the particles as a result of this nullspace. A singular value decomposition (SVD) can be used remove the noise, but the SVD is computationally expensive. An approximate local nullspace filter, which is computationaly efficient, is introduced to aid in the removal of the nullspace noise.

2. In particle methods such as smooth particle hydrodynamics (SPH) and reproducing kernal particle methods (RKPM) the use of corrective derivatives have been used to improve the accuracy of the methods. This dissertation proposes the use of corrective derivatives to improve the accuracy of mapping particle values to nodes in MPM.
3. Stability constraints can become an issue when using explicit time-integration methods for multiphysics problems, because of the differences in timescales between the physical processes being modeled. In order to overcome the stability constraints of the system, implicit or semi-implicit methods can be used. Implicit methods address the issues associated with stability, but do not address the accuracy associated with the time step size. An analysis of local time stepping errors for implict MPM is provided in this work along with the method for computing the local error. This local error can be used to gauge the accuracy of the time step size or be used in an adaptive time stepping algorithm.
4. Traditional MPM solves the conservation of momentum equation in the reference configuration. An alternative is to solve the equation in the spatial configuration. An advantage to this approach is that because the position of the particles in the reference configuration does not change, stability analysis becomes feasible. A stability analysis for the 1D MPM solution in the reference configuration is presented and is shown, under the correct assumptions, to be similar to a finite difference solution.
5. A method for solving the diffusion equation in the reference configuration using MPM is presented along with a discussion of the stability constraints of the given solutions.
6. In order to improve the charge density of batteries, electrode materials are going to need to increase the ratio of ions to host materials. With this increase comes a greater change in volume of the electrode during the full charge/discharge cycle. The research presented here demonstrates the use of MPM as applied to electrode modeling and its ability to model the large deformation behavior that comes with the changes in

volume.

7. A preliminary model, based on the Poisson-Nernst-Planck equations, for simulating the charge/discharge cycle of a full electrochemical cell is provided. Also, an analysis of the use of FVM as a numerical method for solving the model equations, along with the solutions used to address numerical instabilities, is presented.

1.5 Document Organizations

This dissertation will proceed as follows: Chapter 2 will present an overview of the fundamental physics that drive the process in an electrochemical cells, including a discussion on the driving mechanisms for chemical diffusion and deformation. Chapter 3 will give an introduction to the different numerical methods that will be used in modeling an electrochemical cell. Because of the central role that MPM plays in this dissertation, a discussion of the key historical developments in the evolution of MPM along with coverage of key recent developments will be presented. Chapter 4 will discuss the stability issues that arise in MPM along with proposed solutions to the issues. Chapter 5 will present the various approaches to time integration methods for MPM. A discussion of implicit MPM will be given along with analysis of local time step errors. Chapter 6 will discuss modeling diffusion when coupled with MPM in both the spatial and reference configurations. Chapter 7 will present the use of MPM to model a silicon anode that undergoes multiple charge/discharge cycles. Chapter 8 will present the use of the finite volume method to model a full electrochemical cell undergoing a full charge/discharge cycle. Chapter 9 will present concluding remarks along with discussion of future work.

CHAPTER 2

PHYSICS AND COMPUTATIONAL MODELS OF THE ELECTROCHEMICAL CELL

The basic components of an electrochemical cell are an external electronic conductor; two half-cells, which consist of an electrode and an electrolyte; and, in some cases, a permeable membrane or salt bridge that separates the electrolytes of the two half-cells. When the external conductor connects the two electrodes in the half-cells, a circuit is made, and electrons are allowed to transport from one electrode to the other. The movement of electrons across the external conductor is one-half of the electric circuit. The second half of the circuit consists of ions moving across the electrolyte from one electrode to the other [41, 42]. Fig. 2.1 shows a basic schematic of the flow of charge in an electrochemical cell circuit.

2.1 Reduction and Oxidation Reactions

The driving physical principle that makes electrochemical cells useful is the reduction-oxidation (redox) reaction. Oxidation occurs when a chemical species loses one or more electrons in a chemical reaction, given as

$$\frac{n}{z}A \rightarrow \frac{n}{z}A^{z+} + ne- . \tag{2.1}$$

A reduction reaction occurs when a chemical species gains one or more electrons in a chemical reaction, shown as

$$\frac{n}{z}A^{z+} + ne- \rightarrow \frac{n}{z}A. \tag{2.2}$$

The variable n is the number of electrons that are involved in either chemical reaction, and the variable z is the valence of element A. In a standard chemical redox reaction, both the reduction and oxidation reactions occur in the same locale. The combustion reaction involving methane and oxygen, which results in carbon dioxide and water, is an example of a standard redox reaction. That is,

$$CH_4 + 2O_2 \rightarrow CO_2 + 2H_2O. \tag{2.3}$$

The combustion reaction consists of two chemical reactions, the oxidation, i.e.,

$$CH_4 + 2H_2O \rightarrow CO_2 + 8H^+ + 8e^- \tag{2.4}$$

and the reduction, as follows:

$$O_2 + 4H^+ + 4e^- \rightarrow 2H_2O. \tag{2.5}$$

An electrochemical redox reaction differs from a standard chemical redox reaction in that the reduction and oxidation reactions occur in separate locations. Each reaction occurs in one of the half-cells. In standard and electrochemical redox reactions, the rates of the two reactions are coupled because of the physical law of conservation of charge and electroneutrality [42]. Generally, an electrochemical cell is a closed system, i.e., no mass enters or leaves the system. In an electrochemical cell, electric charge is carried by electrons or ionic chemical species, all of which have mass. Because mass is conserved in a closed system, the charge must also be conserved [42].

The principle of electroneutrality stipulates that the balance of charge within a material be zero. For the electrolyte, this means that the concentrations of charge associated with the positive and negative ionic species must be equal [42, 43], i.e.,

$$\sum_i z_i c_i = 0. \tag{2.6}$$

Within the electrode, the concentration of free electrons and the concentration of charge associated with the positive ionic species must be equal [42, 43], as in

$$\sum_i z_i c_i - c_e = 0. \tag{2.7}$$

The variables z_i, c_i, and c_e are the valence, concentration of ionic species i, and the concentration of free electrons, respectively.

Finally, redox reactions within an electrochemcial system are heterogeneous; they do not occur in the bulk electrolyte or in the electrode materials but rather at the interface between the electrode and the electrolyte [42].

2.2 Types of Electrochemical Cells

Electrochemical cells are either galvanic or electrolytic. In galvanic cells, the redox reactions occur spontaneously, driven by the potential difference in the two electrodes. For

electrolytic cells, work is required to drive the redox reactions. Some electrochemical cells are both galvanic and electrolytic, for example, the rechargeable lithium ion battery. To better explain the principles pertaining to electrochemcial cells, examples of both galvanic and electrolytic cells are discussed in Sections 2.2.1 and 2.2.2.

2.2.1 Example of a Galvanic Cell

A simple example of a galvanic cell is the Daniell cell, which consists of one half-cell containing a zinc (Zn) anode and aqueous zinc sulfate electrolyte, as in

$$\mathrm{ZnSO_4} \rightarrow \mathrm{Zn^{2+}(aq)} + \mathrm{SO_4^{2-}(aq)}. \tag{2.8}$$

The other half-cell consists of a copper (Cu) cathode and aqueous copper (II) sulfate electrolyte, as follows:

$$\mathrm{CuSO_4} \rightarrow \mathrm{Cu^{2+}(aq)} + \mathrm{SO_4^{2-}(aq)}. \tag{2.9}$$

A salt bridge containing a sodium sulfate ($\mathrm{Na_2SO_4}$) solution connects the two half-cells. The image on the left of Fig. 2.2 shows a simple diagram of the Daniell cell.

When a circuit is made between the two half-cells, electrons begin to flow, driven by the difference in standard potentials of zinc and cooper. The standard electrode potential of zinc is $-0.76\,V$, and the standard electrode potential of copper is $+0.34\,V$. The voltage of the cell is determined by the difference in potentials of the cathodic half-cell reaction and anodic half-cell reaction. In the case of the copper and zinc electrodes, the voltage is

$$\underbrace{(.34\,V)}_{\text{cathode}} - \underbrace{(-.76\,V)}_{\text{anode}} = 1.0\,V. \tag{2.10}$$

As electrons flow from the anode to the cathode, the copper ions, $\mathrm{Cu^{2+}}$, within the electrolyte are reduced at the cathode/electrolyte interface, producing a new layer of copper on the cathode, which is

$$\mathrm{Cu^{+2}_{(aqueous)}} + 2\mathrm{e^-} \rightarrow \mathrm{Cu_{(solid)}}. \tag{2.11}$$

On the anode side, as electrons are leaving, the anode zinc atoms are being oxidized at the anode/electrolyte interface and are leaving the anode and entering into the electrolyte as zinc ions, shown as

$$\mathrm{Zn_{(solid)}} \rightarrow \mathrm{Zn^{+2}_{(aqueous)}} + 2\mathrm{e^-}. \tag{2.12}$$

As the copper ions are reduced, the charge-neutral electrolyte in the cathode half-cell will become more negatively charged. In the anode half-cell, the zinc atoms in the electrode

are oxidized and enter the electrolyte, thus making it more positive. A salt bridge is used to maintain the charge neutrality of the electrolytes. As the copper ions are reduced at the cathode/electrolyte interface in the cathode half-cell, the positive potassium ions in the salt bridge enter the electrolyte to balance the negative sulfate ions. As zinc is oxidized and enters the electrolyte solution in the anode half-cell, negatively charged sulfate ions enter the electrolyte to maintain the charge balance.

2.2.2 Example of an Electrolytic Cell

In the case of the electrolytic cell, the redox reactions do not occur spontaneously, so an externally applied voltage, potential, or fuel source is needed to drive the redox reactions. An example of an electrolytic cell is a simple fuel cell [44], which consists of two platinum electrodes and a sulfuric acid electrolyte. External fuel sources of hydrogen and oxygen gases are used to drive the redox reactions. Hydrogen gas is allowed to percolate past one electrode and oxygen past the other. The image to the right in Fig. 2.2 shows a diagram of the fuel cell configuration.

As the hydrogen gas passes by the electrode, an oxidation reaction occurs at the electrode/electrolyte interface, producing the following reaction:

$$\mathrm{H_{2(gas)}} \rightarrow 2\mathrm{H^{+}_{(aqueous)}} + 2\mathrm{e^{-}}. \tag{2.13}$$

The protons enter the electrolyte solution, and the electrons pass through the external circuit to the opposite electrode.

On the other side of the fuel cell, the percolating oxygen gas is reduced at the electrode interface. The components of the reduction reaction consist of the excess protons in the electrolyte, the free electrons in the electrode, and the oxygen gas. Both the excess protons and the free electrons are made available as a result of the oxidation reactions occurring at the opposite electrode. Combining these elements produces the following reaction:

$$\frac{1}{2}\mathrm{O_2} + 2\mathrm{H^{+}_{(aqueous)}} + 2\mathrm{e^{-}} \rightarrow \mathrm{H_2O}. \tag{2.14}$$

$\mathrm{H_2O}$ is formed instead of the individual gases $\mathrm{H_2}$ and $\mathrm{O_2}$ on the reduction reaction side of the fuel cell, because $\mathrm{H_2O}$ is at a lower energy configuration compared to the state of the individual gasses.

2.3 Current Electrochemical Materials

Lithium-ion batteries (LIB) have become the defacto for battery use cases where light weight and energy density are a primary requirement. The use cases for LIBs range from powering small portable devices, to electric vehicles, to home energy storage. There is also a range of different materials that are used for the various components within a LIB cell depending on the application. For the purposes of this dissertation, the focus will be on the anode material used within LIBs.

Carbon is common element used in LIB anodes with graphite being the most common anode material [45]. It is a material that is low cost and widely available. It is an intercalation type anode in that the Li-ions move in between the empty spaces of the grid-like structure of the material. Graphite has a low operating volatage and has a theoretical specific capacity of 372mAhg^{-1} . The positive features of graphite is the small volume change that occurs at full charge and its ability to withstand a high number of charge/discharge cycles with the drawbacks being diminished electrochemical properties at low temperatures and lithium dendrite formation, which can lead to short circuiting of the cell [45, 46, 47].

Titanium is another element that is being used as a commercially viable anode material. Common titanium-based anode materials are lithium titanium oxide (LTO) and titanium dioxide. LTO like graphite undergoes minimal strain at full charge and unlike graphite preserves its electrochemical properties at low temperatures. LTO has a theoretical charge capacity of 175mAhg^{-1}, which is lower than graphite [47, 45].

Silicon (Si) as an anode material has some highly desirable properties as well as some major drawbacks. Silicon is an abundant element that is easily obtained at a low cost with minimal environmental impact. As an anode material it has a theoretical specific capacity of 3579mAhg^{-1} at room temperature and a max specific capacity of 4200mAhg^{-1} at 410°C [48], which is a 10x improvement over graphite. Silicon falls within the alloy-type anodes in that, as the lithium moves through the anode, it reacts with the host material forming an alloy. This process allows for high specific capacities at lower potentials [45].

Silicon's high specific capacity comes at a cost in terms of a 300% volume increase at full charge [48]. Over just a few charge/discharge cycles, the anode material can become highly deformed and eventually fractures.

Graphite, LTO, and Si are just three examples of materials that are being used in

electrochemcial cell development. Each of these materials has pros and cons associated with it. A summary of the materials characteristics for each of the anode materials can be found in Table 2.1. For the remainder of this dissertation, the focus will be on LIBs with silicon as the anode material.

2.4 The Thermodynamic Potential of an Electrochemical Cell

A variety of physical processes are involved in the movement of lithium ions through an electrochemical cell during the entire charge/discharge cycle. What drives thoses physical processes is the thermodynamic potential. The purpose of this section is to describe the thermodynamic potential and derive the equations that define the potential within the electrochemical cell.

2.4.1 Thermodynamic Potentials

The internal energy of a system is a function of both entropy and volume, i.e.,

$$U = U(S, V) \tag{2.15}$$

where S is entropy and V is the volume of the system. Experimentally, the equation is difficult to work with because of the inherent difficulty of measuring the quantities of entropy and volume [44]. Applying the Legendre transform to (2.15) produces alternative thermodynamic potentials. The alternative thermodynamic potentials are the Gibbs energy,

$$G(T, p) = U + pV - TS \tag{2.16}$$

where T is temperature and p is pressure, enthalpy,

$$H(S, p) = U + pV \tag{2.17}$$

and Helmholtz energy [44],

$$F(T, V) = U - TS. \tag{2.18}$$

2.4.2 Derivation of the Chemical Potential

The original concept of a chemical potential was developed by Gibbs in his work *On the Equilibrium of Heterogeneous Substances* [49], in which he noted the need to account

for changes in the energy of a system that come as a result of changes in the quantity of a given substance [50]. This formulation can be expressed for an open system as

$$dU = TdS - pdV + \sum_i \mu_i dn_i \tag{2.19}$$

where μ_i is the chemical potential of species i, and dn_i is the change in the quantity of species i in the system. From this formulation, it can be seen that the chemical potential relates the change in the energy of the system to the change in the quantity of the species.

Up to this point, the four thermodynamic potentials, as described in Section 2.4.1, are functions of state dependent upon the intrinsic physical quantities of temperature and pressure and the extrinsic physical quantities of entropy and volume. As noted by Gibbs [49], the changes in the quantities of a given chemical species must also be accounted for as follows:

$$U = U(S, V, n_1, \ldots, n_n), \tag{2.20}$$

$$G = G(T, p, n_1, \ldots, n_n), \tag{2.21}$$

$$H = H(S, p, n_1, \ldots, n_n), \tag{2.22}$$

$$F = F(T, V, n_1, \ldots, n_n). \tag{2.23}$$

The change of the thermodynamic potentials can likewise be derived as follows:

$$dU = \frac{\partial U}{\partial S}dS + \frac{\partial U}{\partial V}dV + \sum_i \frac{\partial U}{\partial n_i}dn_i, \tag{2.24}$$

$$dG = \frac{\partial G}{\partial T}dT + \frac{\partial G}{\partial p}dp + \sum_i \frac{\partial G}{\partial n_i}dn_i, \tag{2.25}$$

$$dH = \frac{\partial H}{\partial S}dS + \frac{\partial H}{\partial p}dp + \sum_i \frac{\partial H}{\partial n_i}dn_i, \tag{2.26}$$

$$dF = \frac{\partial F}{\partial T}dT + \frac{\partial F}{\partial V}dV + \sum_i \frac{\partial F}{\partial n_i}dn_i. \tag{2.27}$$

From these derivations, the relationships for temperature, pressure, volume, and entropy are defined as follows:

$$T = \left(\frac{\partial U}{\partial S}\right)_{V,n} = \left(\frac{\partial H}{\partial S}\right)_{p,n}, \tag{2.28}$$

$$-p = \left(\frac{\partial U}{\partial V}\right)_{S,n} = \left(\frac{\partial F}{\partial V}\right)_{T,n}, \tag{2.29}$$

$$V = \left(\frac{\partial G}{\partial p}\right)_{T,n} = \left(\frac{\partial H}{\partial p}\right)_{S,n}, \tag{2.30}$$

$$-S = \left(\frac{\partial G}{\partial T}\right)_{p,n} = \left(\frac{\partial F}{\partial T}\right)_{V,n}. \tag{2.31}$$

For the chemical potential, the following equalities hold:

$$\mu_i = \left(\frac{\partial U}{\partial n_i}\right)_{S,V,n_{j\neq i}} = \left(\frac{\partial G}{\partial n_i}\right)_{T,p,n_{j\neq i}} = \left(\frac{\partial H}{\partial n_i}\right)_{S,p,n_{j\neq i}} = \left(\frac{\partial F}{\partial n_i}\right)_{T,V,n_{j\neq i}}. \tag{2.32}$$

By combining the results found in (2.24) with (2.28), (2.29), and (2.32), the original relationship observed by Gibbs, as described by (2.19), can be seen.

As shown by the equalities found in (2.32), the chemical potential is the change in energy with respect to the change in the quantity of chemical species i. The change in the Gibbs free energy is then defined as

$$dG = -SdT + Vdp + \sum_i \mu_i dn_i. \tag{2.33}$$

The relationships for the other thermodynamic potentials are defined in a like manner. If the temperature, pressure, and quantities of chemical species other than species i are held constant, the chemical potential is the change in the Gibbs free energy with respect to a change in the quantity of chemical species i, as follows:

$$\mu_i = \frac{dG}{dn_i}. \tag{2.34}$$

2.4.3 The Chemical Potential

The definition of the chemical potential can take many forms depending on the nature of the material. For an ideal solution or mixture, such as ideal gas, the chemical potential is defined as

$$\mu_i = \mu_i^0 + k_B T \ln c_i \tag{2.35}$$

where μ_i^0 is the chemical potential of the species i in its standard state and k_B is Boltzmann's constant.

For nonideal mixtures, the chemical potential can be seen as a combination of an ideal and non-ideal chemical potential, e.g.,

$$\mu_i = \mu_i^{\text{ideal}} + \mu_i^{\text{non-ideal}}. \tag{2.36}$$

To account for this nonideal nature of the mixture, Lewis introduced the concept of thermodynamic activity [43]. Using the concept of thermodynamic activity, the chemical potential is defined as

$$\mu_i = \mu_i^0 + k_B T \ln a_i \tag{2.37}$$

where a_i is the thermodynamic activity of chemical species i. The chemical potential in this form is known as the standard chemical potential [43]. The thermal activity is a measure that accounts for the deviation from the standard state where a thermal activity of $a_i = 1$ represents the standard state, given as

$$\mu_i = \mu_i^0. \tag{2.38}$$

The thermal activity is defined as follows:

$$a_i = c_i \gamma_i \tag{2.39}$$

where γ_i is the activity coefficient that is a measure of the departure of a mixture from the ideal mixture. An activity coefficient of 1 indicates an ideal mixture. Substituting (2.39) into (2.37) produces the following:

$$\mu_i = \mu_i^0 + k_B T \ln(c_i \gamma_i), \tag{2.40}$$

$$= \mu_i^0 + k_B T \ln c_i + k_B T \ln \gamma_i. \tag{2.41}$$

From (2.41), both the ideal,

$$\mu_i^{\text{ideal}} = \mu_i^0 + k_B T \ln \bar{c}_i, \tag{2.42}$$

and nonideal potentials,

$$\mu_i^{\text{non-ideal}} = k_B T \ln \gamma_i, \tag{2.43}$$

of the standard chemical potential can be seen.

2.4.4 The Electrochemical Potential

For noncharged chemical species, the chemical potential describes the change in thermodynamics potential with respect to a change in the quantity of a given chemical species. For a charged chemical species, such as ions or electrons, additional considerations need to be made.

A charged chemical species is acted upon by electric and magnetic fields. These fields exert a force on a charged particle. The mathematics that describe the electrostatics and electrodynamics are defined by Maxwell's equations. When a change in the thermodynamic potential occurs as a result of change in the quantity of chemical species, the chemical potential is not sufficient, and thus the chemical potential needs to be extended to account for the electrodynamic effects on the thermodynamic potential. The potential of charged chemical species acting under an electric field is defined as

$$\mu_i^{elec} = z_i e \phi \tag{2.44}$$

where z is the valence of the chemical species i, e is the elemental charge, and ϕ is the electrostatic potential. The electrochemical potential is determined by combining the potential defined by (2.44) with the chemical potential, as follows:

$$\overline{\mu}_i = \mu_i + z_i e \phi. \tag{2.45}$$

2.4.5 Stress-Dependent Chemical Potential

A final component that can contribute to the overall thermodynamic potential is stress. In the work proposed by Li et al. [51] and Larché and Cahn [52], [53], [54], [55], a framework was developed that addressed the added contribution of stress on the overall chemical potential.

The Larché and Cahn chemical potential for a guest chemical species, i, within a host material is generally defined as follows [55], [56], and [57]:

$$\mu_i = \mu_i^0 + k_B T \ln(\gamma_i c_i) - \Omega_i \left(\frac{d\mathbf{E}^c}{dc_i} : \boldsymbol{\sigma} + \boldsymbol{\sigma} : \frac{d\mathbb{S}}{dc_i} : \boldsymbol{\sigma} \right) \tag{2.46}$$

where Ω_i is the partial molar volume which is the change in volume per change in concentration, i.e., $\Delta V / \Delta c_i$, $\mathbf{E}^c$ is the concentration dependent strain, $\boldsymbol{\sigma}$ is the Cauchy stress, and $\mathbb{S}$ is the compliance tensor. In the case where the concentration-dependent strain is a linear isotropic relationship, e.g.,

$$\mathbf{E}^c = \beta(c - c_0)\mathbf{I}, \tag{2.47}$$

then (2.46) reduces to

$$\mu_i = \mu_i^0 + k_B T \ln(\gamma_i c_i) - \Omega_i \left(\beta tr(\boldsymbol{\sigma}) + \boldsymbol{\sigma} : \frac{d\mathbb{S}}{dc_i} : \boldsymbol{\sigma} \right). \tag{2.48}$$

2.5 Description of a Continuum Body

In the previous section a description of a stress dependent chemical potential was given. In order to model stress and strain we need a brief discussion of the two description of a continuum body, the reference and spatial configurations. The reference configuration is the geometric description of a continuum body at a fixed point in time, Ω_0. Other terms used for the reference configuration are the material configuration and Lagrangian configuration. The current configuration is the geometric description of the continuum body at a given point in time, Ω. Other terms used for the current configuration are spatial configuration and Eulerian configuration.

The mapping from the reference configuration to the current configuration is given as

$$\mathbf{x} = \Phi(\mathbf{X}, t) \tag{2.49}$$

where $\mathbf{x}$ is the coordinate in the spatial configuration, and $\mathbf{X}$ is the coordinate in the reference configuration. As a point of notation, vector and tensor values will be in bold. Unless otherwise specified, vector values in the reference configuration will use upper case letters, where as vector values in the spatial configuration will be upper case, following the convention used by [58]. This mapping is continuous, one-to-one, and onto and thus invertible, i.e.,

$$\mathbf{X} = \Phi^{-1}(\mathbf{x}, t). \tag{2.50}$$

The initial configuration is the configuration at $t = 0$. In most cases the initial and reference configuration are the same, i.e.,

$$\mathbf{x} = \Phi(\mathbf{X}, 0) \tag{2.51}$$

such that $\mathbf{x} = \mathbf{X}$. Fig. 2.3 is a schematic description of the mapping between the reference and current configurations defined by (2.49).

By defining a continuum body in terms of its reference and current configurations we are able to derive mathematical models that describe physical properties such as stress and strain of a deformible body. It also allows us to derive mathematical descriptions of other physical process occuring on a physical body undergoing deformation.

Using (2.49) we can derive a handful of other equations that are used to derive mathematical models of the key physical process mentioned above. The first of these equations is

the deformation gradient. The deformation gradient is derivative of (2.49) with respect to the material coordinates as described by

$$\mathbf{F} = \frac{\partial \Phi}{\partial \mathbf{X}}. \tag{2.52}$$

An alternate description using index notation is written as

$$\mathrm{F}_{ij} = \frac{\partial \Phi_i}{\partial \mathrm{X}_j} = \Phi_{i,j}. \tag{2.53}$$

The inverse of the deformation gradient is the gradient of (2.50) with respect to the spatial coordinates as defined follows:

$$\mathbf{F}^{-1} = \frac{\partial \Phi^{-1}}{\partial \mathbf{x}}. \tag{2.54}$$

The gradient and divergence operators with respect to the material coordinates will be represented, respectively, by $\nabla_{\mathbf{X}}()$ and $\nabla_{\mathbf{X}} \cdot ()$ and the gradient and divergence operators with respect to the spatial coordinate will be defined, respectively, by $\nabla()$ and $\nabla \cdot ()$. Using this notation the deformation and inverse of the deformation gradients are defined as

$$\mathbf{F} = \nabla_{\mathbf{X}} \Phi(\mathbf{X}, t) \tag{2.55}$$

and

$$\mathbf{F}^{-1} = \nabla \Phi^{-1}(\mathbf{x}, t) \tag{2.56}$$

respectively.

The velocity in terms of the material coordinates is derived as

$$\mathbf{V}(\mathbf{X}, t) = \frac{\partial \Phi(\mathbf{X}, t)}{\partial t}. \tag{2.57}$$

Substituting (2.50) into (2.57) produces the following relationship between the velocity in material and spatial coordinates:

$$\mathbf{V}(\mathbf{X}, t) = \mathbf{V}(\Phi^{-1}(\mathbf{x}, t), t) = \mathbf{v}(\mathbf{x}, t). \tag{2.58}$$

The material time derivative is defined as

$$\frac{D()}{Dt} = \frac{\partial()}{\partial t} + \mathbf{v} \cdot \nabla(). \tag{2.59}$$

The common shorthand notation of using a dot over a given field quantity to denote a material time derivative will also be used in this dissertation, e.g.,

$$\frac{Dg}{Dt} \equiv \dot{g}. \tag{2.60}$$

2.6 Diffusion

The total mass of a continuum body at any given point in time is the sum of the mass of the host material and the guest materials, i.e.,

$$m(t) = m_h(t) + m_g(t), \tag{2.61}$$

where subscripts h and g indicate host and guest materials, respectively. By definition the total mass can be derived at any point in time by integrating the density over the continuum body,

$$m(t) = \int_{\Omega_0} \rho_0(t)\, dV = \int_{\Omega} \rho(t)\, dv. \tag{2.62}$$

Using (2.61) and (2.62) the following relationships for the host and guest material are defined respectively as

$$m_h(t) = \int_{\Omega} \rho_h dv \quad \text{and} \quad m_g(t) = \int_{\Omega} \rho_g dv. \tag{2.63}$$

The density and concentration of the host and guest materials are related to each other by the equations

$$\rho_h = M_h c_h \quad \text{and} \quad \rho_g = M_g c_g \tag{2.64}$$

where c_h and c_g are the host and guest concentrations and M_h and M_g are the molar masses of the host and guest material, respectively. Using the relationships defined by (2.63) and (2.64) the total quantity of material for the host and guest species is defined, respectively, as

$$N_h(t) = \frac{m_h(t)}{M_h} = \int_{\Omega} c_h dv \quad \text{and} \quad N_g(t) = \frac{m_g(t)}{M_g} = \int_{\Omega} c_g dv. \tag{2.65}$$

The total quantity of the host material does not change, i.e.,

$$\dot{N}_h(t) = \frac{D}{Dt} \int_{\Omega} c_h dv = 0 \tag{2.66}$$

The change in the total quantity of guest species comes as result of flux of guest material across the continuum body's surface, i.e.,

$$\dot{N}_g(t) = \frac{D}{Dt} \int_{\Omega} c_g dv = -\int_{\Gamma} \mathbf{j}_g \cdot \mathbf{n} da. \tag{2.67}$$

As was noted earlier, the continuum body, Ω, deforms over time and thus the time derivative and volume integral are not interchangeable. The same method that was used to rewrite

the change in linear momentum equation defined by (2.86) is used to rewrite the change in quantity of the guest species. The results of the method produce the following equality,

$$\frac{D}{Dt}\int_{\Omega} c_g dv = \int_{\Omega} [\dot{c}_g + c_g(\nabla \cdot \mathbf{v})] dv. \tag{2.68}$$

Using (2.68) and by means of the divergence theorem, (2.67) can be rewritten as

$$\int_{\Omega} [\dot{c}_g + c_g(\nabla \cdot \mathbf{v})] dv = -\int_{\Omega} \nabla \cdot \mathbf{j}_g dv. \tag{2.69}$$

Noting that (2.69) must be true for any arbitrary volume, it can be written in its local form as

$$\dot{c}_g + c_g(\nabla \cdot \mathbf{v}) = -\nabla \cdot \mathbf{j}_g. \tag{2.70}$$

The flux of the guest species is a function of the local concentration and the chemical potential, μ_g, of the guest species within the host material,

$$\mathbf{j}_g = -M_g c_g \nabla \mu_g. \tag{2.71}$$

M_g is the mobility defined by the Einstein relation [59],

$$M_g = \frac{D_g^{trace}}{RT} \tag{2.72}$$

where D_g^{trace} is the tracer diffusivity, R is the gas constant, and T is the temperature. If guest species carries a charge and an electrostatic field is present, then the mass flux is defined by the generalized Nernst-Planck transport equation [41, 42], which is

$$\mathbf{j}_g = -M_g c_g [\nabla \mu_g + z_g F \nabla \phi]. \tag{2.73}$$

where z_g is the valence and F is the Faraday constant.

2.6.1 Electrostatics

The governing equation for the electrostatic field, ϕ, as used in Equation 2.73 is defined as follows [60]:

$$\nabla \cdot \tilde{\mathbf{E}} = \tilde{\rho}. \tag{2.74}$$

The vector field $\tilde{\mathbf{E}}$ is the electric field which is defined as a force per unit charge and is proportional to gradient of the electrostatic potential, i.e.,

$$\tilde{\mathbf{E}} = -\epsilon \nabla \phi. \tag{2.75}$$

The values ϵ and ϕ are the electric permittivity and electrostatic potential, respectively. The charge density, $\tilde{\rho}$, is related to the concentrations of charged chemical species in the following manner:

$$\tilde{\rho} = q \sum_i z_i c_i \tag{2.76}$$

where q is the unit charge and z_i is the valence of chemical species i. In the case were the permittivity is constant through out the domain, (2.74) and (2.75) can be combined and simplified to the Poisson equation,

$$\nabla^2 \phi = -\frac{\tilde{\rho}}{\epsilon}. \tag{2.77}$$

2.7 Deformation

An understanding of the physical process of deformation along with the derivation of the governing equations can begin with the equations for linear momentum in the spatial and reference configuration, respectively [58],

$$\mathbf{LM}(t) = \int_\Omega \rho(\mathbf{x}, t)\mathbf{v}(\mathbf{x}, t)dv = \int_{\Omega_0} \rho_0(\mathbf{X}, t)\mathbf{V}(\mathbf{X}, t)dV. \tag{2.78}$$

The values ρ and ρ_0 respectively represent density in the current and reference configurations, and the values $\mathbf{v}$ and $\mathbf{V}$ respectively represent velocity in the current and reference configurations. The change in linear momentum is defined as

$$\overline{\dot{\mathbf{LM}}} = \frac{D}{Dt}\int_\Omega \rho(\mathbf{x}, t)\mathbf{v}(\mathbf{x}, t)dv = \frac{D}{Dt}\int_{\Omega_0} \rho_0(\mathbf{X}, t)\mathbf{V}(\mathbf{X}, t)dV. \tag{2.79}$$

In the reference configuration, the geometry of the continuum body remains unchanged, as a result the time derivative and the volume integral are interchangeable, as in

$$\frac{D}{Dt}\int_{\Omega_0} \rho_0 \mathbf{V} dV = \int_{\Omega_0} \overline{\dot{\rho_0 \mathbf{V}}} dV = \int_{\Omega_0} (\dot{\rho_0}\mathbf{V} + \rho_0 \dot{\mathbf{V}})dV. \tag{2.80}$$

In the spatial configuration, the geometry changes with respect to time, and thus a change in variables is needed in order to swap the time derivative and volume integral. The relationship between a defined volume in the reference configuration and its deformed state in the spatial configuration is [58]

$$v = JV \tag{2.81}$$

where J is the volume ratio which is the determinant of the deformation gradient, $J = \det(\mathbf{F})$. The material time derivative of the volume ratio is defined by [58]

$$\dot{J} = J(\nabla \cdot \mathbf{v}). \tag{2.82}$$

Using (2.81) and a change of variables, the equation for the change in linear momentum is written as a volume integral over the material coordinates, given as

$$\frac{D}{Dt}\int_{\Omega} \rho(\mathbf{x},t)\mathbf{v}(\mathbf{x},t)dv = \frac{D}{Dt}\int_{\Omega_0} \rho(\Phi(\mathbf{X},t),t)\mathbf{v}(\Phi(\mathbf{X},t),t)J(\mathbf{X},t)dV \tag{2.83}$$

$$= \int_{\Omega_0} \frac{D}{Dt}[\rho(\Phi(\mathbf{X},t),t)\mathbf{v}(\Phi(\mathbf{X},t),t)J(\mathbf{X},t)]dV. \tag{2.84}$$

Then, applying the product rule and (2.82) leads to

$$\int_{\Omega_0} \dot{\overline{(\rho \mathbf{v} J)}}dV = \int_{\Omega_0} [\dot{\rho}\mathbf{v} + \rho\dot{\mathbf{v}} + \rho\mathbf{v}(\nabla \cdot \mathbf{v})]JdV. \tag{2.85}$$

Using a change of variables again leads to the following relationships:

$$\frac{D}{Dt}\int_{\Omega} \rho\mathbf{v}dv = \int_{\Omega} [\rho\dot{\mathbf{v}} + (\dot{\rho} + \rho(\nabla \cdot \mathbf{v}))\mathbf{v}]dv. \tag{2.86}$$

The change in linear momentum comes as a result of stresses acting on the surface of the continuum body and body forces. In the spatial configuration, the stresses acting on the surface are defined by the Cauchy traction vector, which is

$$\mathbf{t}(\mathbf{x},t,\mathbf{n}) = \boldsymbol{\sigma}(\mathbf{x},t)\mathbf{n} \tag{2.87}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor, and $\mathbf{n}$ is the surface normal in the spatial configuration. In the material configuration the surface stresses are defined by the first Piola-Kirchhoff traction vector, given as

$$\mathbf{T}(\mathbf{X},t,\mathbf{N}) = \mathbf{P}(\mathbf{X},t)\mathbf{N} \tag{2.88}$$

where $\mathbf{P}$ is the first Piola-Kirchhoff stress tensor, and $\mathbf{N}$ is the surface normal in the reference configuration. Traction vectors are defined as force per unit area. The body forces are symbolized by $\mathbf{b}$ in the spatial configuration and $\mathbf{B}$ in the material configuration and are defined as force per unit volume. An example of a common body force is the force due to gravity, i.e.,

$$\mathbf{b} = \rho\mathbf{g} \tag{2.89}$$

where $\mathbf{g}$ is the gravitational acceleration.

The total force acting on the continuum body in the spatial description is then defined as

$$\text{Total Force} = \int_{\Gamma} \mathbf{t} da + \int_{\Omega} \mathbf{b} dv = \int_{\Gamma} \boldsymbol{\sigma}\mathbf{n} da + \int_{\Omega} \mathbf{b} dv \tag{2.90}$$

where Γ is the surface of the continuum body, Ω. The total force in the material description is given as

$$\text{Total Force} = \int_{\Gamma_0} \mathbf{T} dA + \int_{\Omega_0} \mathbf{B} dV = \int_{\Gamma_0} \mathbf{P}\mathbf{N} dA + \int_{\Omega_0} \mathbf{B} dV \tag{2.91}$$

where Γ_0 is the surface of the continuum body, Ω_0. Using the divergence theorem, (2.90) and (2.91) can be rewritten as

$$\text{Total Force} = \int_{\Omega} (\nabla \cdot \boldsymbol{\sigma} + \mathbf{b}) dv \tag{2.92}$$

and

$$\text{Total Force} = \int_{\Omega_0} (\nabla_{\mathbf{X}} \cdot \mathbf{P} + \mathbf{B}) dV \tag{2.93}$$

respectively. Combining (2.86) and (2.92) produces the momentum balance equation in the spatial configuration

$$\int_{\Omega} [\rho \dot{\mathbf{v}} + (\dot{\rho} + \rho(\nabla \cdot \mathbf{v}))\mathbf{v}] dv = \int_{\Omega} (\nabla \cdot \boldsymbol{\sigma} + \mathbf{b}) dv \tag{2.94}$$

and combining (2.80) and (2.93) produces the momentum balance equation in the material configuration

$$\int_{\Omega_0} (\dot{\rho_0}\mathbf{V} + \rho_0 \dot{\mathbf{V}}) dV = \int_{\Omega_0} (\nabla_{\mathbf{X}} \cdot \mathbf{P} + \mathbf{B}) dV. \tag{2.95}$$

Given that (2.94) and (2.95) must hold for any arbitrary volume the following is also true

$$\rho \dot{\mathbf{v}} + (\dot{\rho} + \rho(\nabla \cdot \mathbf{v}))\mathbf{v} = \nabla \cdot \boldsymbol{\sigma} + \mathbf{b} \tag{2.96}$$

and

$$\dot{\rho_0}\mathbf{V} + \rho_0 \dot{\mathbf{V}} = \nabla_{\mathbf{X}} \cdot \mathbf{P} + \mathbf{B}. \tag{2.97}$$

In those cases where mass is conserved, i.e.,

$$\frac{D}{Dt} \int_{\Omega_0} \rho_0 dV = \frac{D}{Dt} \int_{\Omega} \rho dv = 0 \tag{2.98}$$

(2.96) and (2.97) reduce to

$$\rho \dot{\mathbf{v}} = \nabla \cdot \boldsymbol{\sigma} + \mathbf{b} \tag{2.99}$$

and

$$\rho_0 \dot{\mathbf{V}} = \nabla_{\mathbf{X}} \cdot \mathbf{P} + \mathbf{B}, \tag{2.100}$$

respectively.

A general description of the physical processes of diffusion and deformation have been presented in Sections 2.6 and 2.7, respectively. In the final section of this chapter, a brief discussion on how the two physical process are coupled is presented.

2.8 Chemical/Mechanical Coupling of Diffusion and Deformation Processes

The original work by Larché and Cahn's established the foundation for modeling the coupled physical process of deformation and diffusion [52], [53], [54], and [55]. The Larché and Cahn model consists of an embedded network or lattice within the solid that tracks the displacement of the solid material. Diffusion occurs through substitution of chemical species, i.e., atom/atom or atom/vacancy, between lattice sites. In the Larché and Cahn model, the deformation is assumed to be elastic and the quantity of lattice sites are conserved.

New models have been developed that extend the work of Larché and Cahn as applied to electrochemistry problems. [61], [62], and [63] extended the Larché and Cahn model to include plastic deformation while still conserving the number of lattice sites. Drosdov separated the guest species into two types, mobile and immobile, which accounted for the alloying of the guest species with the host medium [64]. Bower et al. developed a model that included lattice sites for both the guest and host species [65]. In this model the guest species can occupy both host and guest lattice sites. Their model also accounts for irreversible changes in the underlying network where lattice may be created do to viscoplastic flow. Full models have also been developed that involve the coupling of deformation and plastic flow, diffusion, and electrostatics as applied to a complete electrochemical cell [66].

This section will give an overview of the different strategies that are used to couple the diffusion and deformation processes. First will be a discussion on how changes in concentration affect stress followed by a discussion on the effects of stress or strain on the transport of the guest chemical species in the host medium.

2.8.1 Chemical Diffusion's Effects on Stress

Work relating the effects of chemical diffusion on stress can be found in the original work by Prussin [67]. The method used by Prussin was by means of an additive decomposition of the total strain of the host material into elastic and concentration dependent portions, given as

$$\boldsymbol{\epsilon} = \boldsymbol{\epsilon}^e + \boldsymbol{\epsilon}^c, \tag{2.101}$$

where $\boldsymbol{\epsilon}^e$ and $\boldsymbol{\epsilon}^c$ are the elastic and concentration-dependent strains, respectively. This method of strain decomposition is akin to the method used for modeling thermal stress [68]. Like the model used by Timoshenko [68] for thermal stress, Prussin uses a linear relationship between concentration levels and strain for the concentration-dependent portion of the total strain,

$$\boldsymbol{\epsilon}^c = \beta(c - c_0)\mathbf{I}, \tag{2.102}$$

where β is the chemical expansion term and c_0 is the strain-free concentration. By defining the total strain as an additive decomposition of elastic and concentration-dependent parts, an unbounded material with a uniform insertion of guest chemical species leads to a uniform change in volume with strain being solely dependent on the concentration levels. In this scenario the total strain would be the concentration dependent strain. In the case of (2.102), concentration-dependent strain is a linear isotropic relationship to concentration. Generally, concentration-dependent strain is a function of the current and initial concentration levels, e.g.,

$$\boldsymbol{\epsilon}^c = \mathbf{f}(c - c_0). \tag{2.103}$$

In the case of a linear anisotropic relationship, the chemical expansion term is a tensor, as shown by

$$\boldsymbol{\epsilon}^c = \boldsymbol{\beta}(c - c_0). \tag{2.104}$$

For linear elastic constitutive models the relationship between stress and elastic strain is defined as follows:

$$\boldsymbol{\epsilon}^e = \mathbb{S} : \boldsymbol{\sigma} \tag{2.105}$$

where $\mathbb{S}$ is the compliance tensor. Combining (2.101) with (2.102) and (2.105) produces the following equation:

$$\boldsymbol{\epsilon} = \mathbb{S} : \boldsymbol{\sigma} + \beta(c - c_0)\mathbf{I}. \tag{2.106}$$

Given that (2.106) is linear, stress is then defined as

$$\boldsymbol{\sigma} = \mathbb{C} : [\boldsymbol{\epsilon} - \beta(c - c_0)\mathbf{I}], \tag{2.107}$$

where $\mathbb{C}$ is the elastic or stiffness tensor.

This additive decomposition of strain can also be extended to include plastic behavior. As presented in the work by Stephensen [69], the total strain is represented as

$$\boldsymbol{\epsilon} = \boldsymbol{\epsilon}^e + \boldsymbol{\epsilon}^p + \boldsymbol{\epsilon}^c. \tag{2.108}$$

A review of plastic and viscoplastic theory based on the additive decomposition of strain can be found in [70].

Christensen and Newman [71] proposed an alternate approach to coupling chemical diffusion and stress by defining the total stress as the difference between the elastic stress and a thermodynamic pressure,

$$\boldsymbol{\sigma}^{total} = \boldsymbol{\sigma} - \mathbf{p}^{therm}. \tag{2.109}$$

This method is equivalent to the additive strain decomposition model when the total stress is defined as,

$$\boldsymbol{\sigma}^{total} = \mathbb{C} : \boldsymbol{\epsilon}, \tag{2.110}$$

and the thermodynamic pressure is defined by,

$$\mathbf{p}^{therm} = \beta(c - c_0)\mathbb{C} : \mathbf{I}. \tag{2.111}$$

An alternate approach to modeling elastic and nonelastic deformation behavior is through a multiplicative decomposition of the deformation gradient [66], [63], as follows:

$$\mathbf{F} = \mathbf{F}^e\mathbf{F}^* \tag{2.112}$$

This approach, made popular by Lee [72], consists of elastic, $\mathbf{F}^e$, and nonelastic, $\mathbf{F}^*$, components. In the case of a constitutive model consisting of elastic-, plastic-, and concentration-dependent deformations, the nonelastic component of (2.112) is

$$\mathbf{F}^* = \mathbf{F}^c\mathbf{F}^p, \tag{2.113}$$

with $\mathbf{F}^c$ and $\mathbf{F}^p$ being the concentration-dependent and plastic deformation contributions, respectively. Fig. 2.4 shows schematically how the deformation decomposition works for concentration-dependent and elastic deformations.

The correlation between strain and the deformation gradient is dependent upon the strain measure. Using the Green-Lagrangian strain measure, we have

$$\mathbf{E} = \frac{1}{2}(\mathbf{F}^T\mathbf{F} - \mathbf{I}). \tag{2.114}$$

The relationship between the elastic strain and the concentration dependent strain can be derived as follows:

$$\begin{aligned}\mathbf{E} &= \frac{1}{2}\left[(\mathbf{F}^c)^T(\mathbf{F}^e)^T\mathbf{F}^e\mathbf{F}^c - \mathbf{I}\right] \\ &= \frac{1}{2}\left[(\mathbf{F}^c)^T((\mathbf{F}^e)^T\mathbf{F}^e - \mathbf{I} + \mathbf{I})\mathbf{F}^c - \mathbf{I}\right] \\ &= (\mathbf{F}^c)^T\mathbf{E}^e\mathbf{F}^c + \mathbf{E}^c\end{aligned}$$

where

$$\mathbf{E}^e = \frac{1}{2}\left[(\mathbf{F}^e)^T\mathbf{F}^e - \mathbf{I}\right] \tag{2.115}$$

$$\mathbf{E}^c = \frac{1}{2}\left[(\mathbf{F}^c)^T\mathbf{F}^c - \mathbf{I}\right]. \tag{2.116}$$

The elastic strain, in terms of the total strain and the concentration-dependent strain, is then

$$\mathbf{E}^e = (\mathbf{F}^c)^{-T}(\mathbf{E} - \mathbf{E}^c)(\mathbf{F}^c)^{-1}. \tag{2.117}$$

A review of constitutive models based on the multiplicative deformation gradient can be found in [73] and as applied to battery electrodes [57], [63].

2.9 Conclusion

This chapter has covered some of the fundemental physics that are involved in an electrochemcial cell. A brief explaination of the types of electrochemical cells was presented along with a survey of some of the materials that are used in a cell. One material that will be of interest in this thesis is silicon because of its large carrying capacity for lithium at full charge. The physical processes of diffusion and deformation were presented along with a discussion on the coupling of the two processes. An understanding of the coupled processes is of particular importantance when examining silicon as an anode material because at full charge a silicon anode can swell up to 300%.

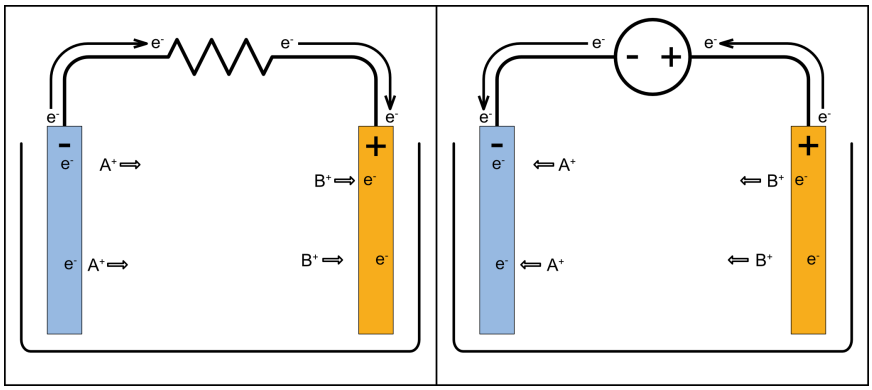

Fig. 2.1. Schematic of galvanic and electrolytic cells.

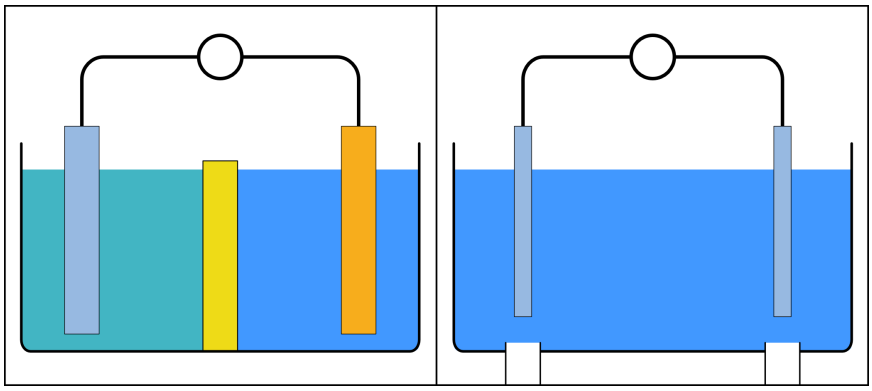

Fig. 2.2. Diagrams of the Daniell cell and fuel cell.

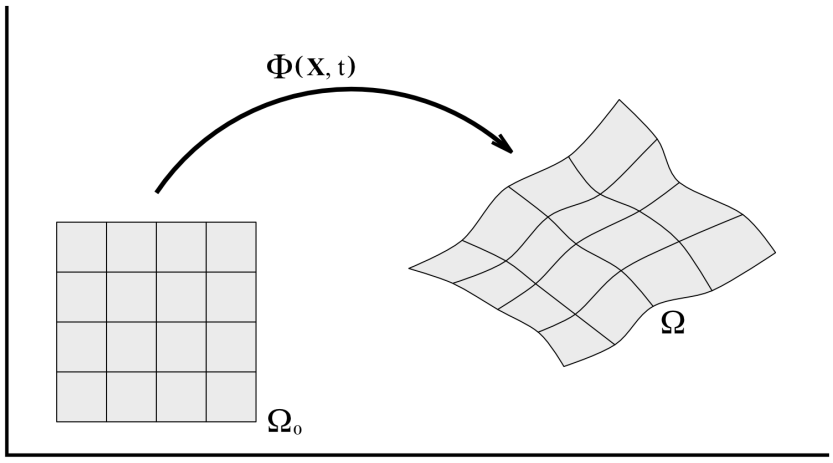

Fig. 2.3. Schematic of deformation.

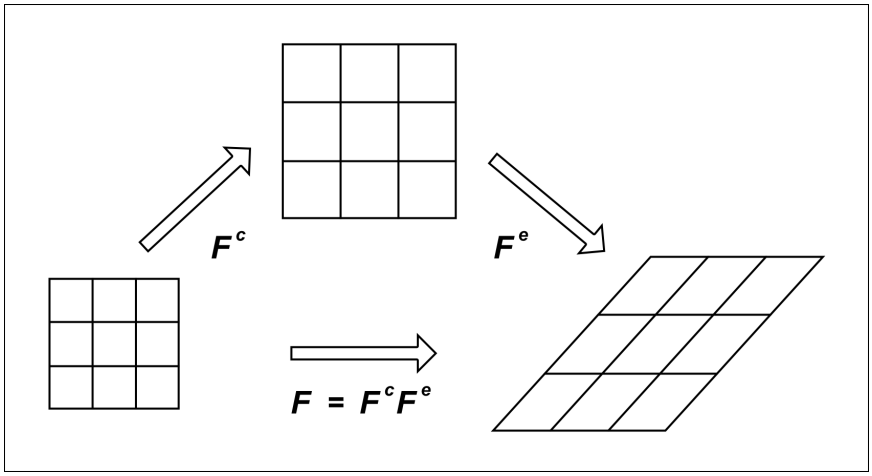

Fig. 2.4. Deformation decomposition.

Table 2.1. Summary of anode materials [45, 46, 47, 48].

Material	Capacity($mAhg^{-1}$)	Volume Change	Cycle Life
Graphite	372	10%	> 1000
LTO	175	0.1%	-
Si	3579	300%	$200 - 300$

CHAPTER 3

NUMERICAL METHODS FOR MODELING ELECTROCHEMISTRY PROBLEMS

Because of the multiphysics nature of electrochemistry problem, special care needs to be taken when implementing numerical codes to simulate these problems. Different numerical methods and approaches are used in this publication but the material point method (MPM) is of primary focus both as a tool for the simulation of electrochemistry problems and as topic of research. Because of the central role that MPM plays in this work and because of the resent interest in the method generally, a discussion of the method, salient features, and some of the key historical developments that lead to its development will be covered.

3.1 The Origins of the Particle-in-Cell Method

MPM belongs to a larger class of numerical method called particle-in-cell (PIC) methods. The original PIC method was developed by Francis Harlow and colleagues in 1955 at what was then the Los Alamos Scientific Laboratory [74, 14]. At the time Los Alamos was developing new computing machines that allowed scientists and engineers to explore solutions to difficult problems. Harlow's original focus at Los Alamos was in the field of fluid dynamics, specifically supersonic flows and shock behavior [75].

Harlow explored two approaches to model supersonic flows. The first was a Lagrangian approach in which the computational mesh moved and distorted with the flow of the fluid. The advantage of this approach was that it could easily model interfaces. The disadvantage of this approach was that it was difficult to model large distortions, slippages, and cavitations. The second was an Eulerian approach where the mesh remained fixed with the fluid flowing through it. The advantage of the Eulerian approach was that it could handle distortions and large slippages easily but had difficulty handling interfaces and smeared out sharp features [14], [75]. Today, many of the negatives associated with both the Lagrangian

and Eulerian approaches have been resolved, but at the time these issues where the catalyst for the development of the original PIC method [75].

In 1955, Francis Harlow and Martha Evans began to look at ways to combine the two approaches that would take advantage of the positives of the Eulerian and Lagrangian approaches while mitigating the negatives. Their method involved using a fixed Eulerian mesh combined with Lagrangian particles or marker [74]. Quantities such as velocity, density, internal energy, and pressure are mesh values while mass is assigned to the particles. PIC uses the Eulerian mesh to resolve the field variables, i.e., pressure, velocity, and energy, whereas Lagrangian particles act as fluid elements that transport according to the fluid flow [74], [76], [77], [14].

The original PIC method developed by Harlow [74] was extended to solve plasma physics problems [78, 79], which was then followed by the development of the fluid-implicit-particle method (FLIP) [15] for fluids problems. The FLIP method then lead to the development of the material point method (MPM) for solid mechanics problem. What follows in Sections 3.2, 3.3, 3.4, and 3.5 is a brief discussion of each of these developmental phases.

3.2 The Formulation of the Original PIC Method

Each time step of the PIC method consists of an Eulerian and Lagrangian phase [14]. To see how the two phases are implemented, the PIC method is applied to the following set of conservative equations for fluid flow:

$$\frac{\partial \rho}{\partial t} + \mathbf{v} \cdot \nabla \rho + \rho \nabla \cdot \mathbf{v} = 0 \tag{3.1}$$

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \rho \left(\mathbf{v} \cdot \nabla \mathbf{v} \right) + \nabla p = 0 \tag{3.2}$$

$$\rho \frac{\partial e}{\partial t} + \rho \left(\mathbf{v} \cdot \nabla e \right) + p \nabla \cdot \mathbf{v} = 0 \tag{3.3}$$

where (3.1), (3.2), and (3.3) represents the equations for the conservation of mass, momentum, and energy, respectively. The variables ρ, $\mathbf{v}$, p, and e represents mass density, velocity, pressure, and specific internal energy, respectively. As will be seen in Section 3.2.1, a mass is assigned to each particle in the initial discretization of the problem. Mass is conserved because the quantity of mass assigned to each particle remains fixed through the duration of the simulation and thus (3.1) does not need to be solved explicitly.

3.2.1 The Preprocessing Phase

The preproccessing phase of the PIC method involves discretizing the problem domain onto a cell-based Cartesian mesh. The field variables for velocity and internal energy are assigned to the cell centers and mass is assigned to the particles base on the number of particles in each cell, i.e.,

$$m_p = \frac{\rho^0 v_c}{N_c}. \tag{3.4}$$

Subscript c represents the cell index and subscript p represent the particle index. The initial density of the material is ρ_0, v_c is the cell volume, and N_c is the number of particles in cell.

3.2.2 The Eulerian Phase

The field values of velocity, specific internal energy, and density are represented by a set of mean value approximations at each cell in the Eulerian grid. During the Eulerian phase, the particles are assumed to be fixed, and fluid motion and other transport process are ignored.

3.2.2.1 Calculation of Cell Density

The total mass of the cell is determined by the following equation:

$$m_c^n = N_c^n m_p. \tag{3.5}$$

Superscript n indicates that N_c^n represents the the number particles in the cell at the beginning of the timestep. The density of the cell is simply the total cell mass divided by the cell volume, i.e.,

$$\rho_c^n = \frac{m_c^n}{v_c}. \tag{3.6}$$

An integer number of particles in a cell results in an approximation of density that is not continuous but rather step-wise across the domain.

3.2.2.2 Calculation of Cell Pressure

The calculation of the cell pressure is dependent upon the chosen equation of state, which is function of density and specific internal energy, e.g.,

$$p_c^n = f(\rho_c^n, e_c^n). \tag{3.7}$$

For a cell that consists of just one material (3.7) is sufficient for determining the cell pressure. In the case of a cell containing a mixture of material types the calculation of the cell pressure

may not be as straightforward. In some cases, simply adding the partial pressures of the different particle types may be sufficient. When the equation of state is dependent upon variables other than density, a simple summing of partial pressures may still be possible, but in most cases this method may produce total pressures that vary greatly from what should be the actual pressure [80], [14].

3.2.2.3 Approximation of Momentum Conservation Equation

Because particles are considered fixed during the Eulerian phase of the PIC calculation the transport term,

$$\mathbf{v} \cdot \nabla \mathbf{v} \tag{3.8}$$

found in (3.2) is not approximated. Accounting for the absence of the transport term, (3.2) is written as

$$\frac{\partial \mathbf{v}}{\partial t} = -\frac{1}{\rho} \nabla p. \tag{3.9}$$

The pressure gradient is approximated using a centered finite difference scheme. On a 2 dimensional domain (2d) let the components of $\mathbf{v}$ be represented by u and v and let the subscripts i and j be the indicies of the cell-centers in the x and y directions, respectively. The approximation of (3.9) uses a centered finite difference approximation

$$\frac{\partial u_{ij}}{\partial t} = -\frac{1}{\rho_{ij}^n} \frac{p_{i+1,j}^n - p_{i-1,j}^n}{2\Delta x} \tag{3.10}$$

$$\frac{\partial v_{ij}}{\partial t} = -\frac{1}{\rho_{ij}^n} \frac{p_{i,j+1}^n - p_{i,j-1}^n}{2\Delta y}. \tag{3.11}$$

An intermediate cell-centered velocity is computed as follows:

$$u_{ij}^* = u_{ij}^n + \Delta t \frac{\partial u_{ij}}{\partial t} \tag{3.12}$$

$$v_{ij}^* = v_{ij}^n + \Delta t \frac{\partial v_{ij}}{\partial t} \tag{3.13}$$

where u_{ij}^* and v_{ij}^* represent intermediary values for the updated cell values of u_{ij} and v_{ij}, respectively. The intermediary values are the results of the Eulerian phase of the calculation and have yet to take into account the transport terms.

3.2.2.4 Approximation of Energy Conservation Equation

As was the case in the approximation of the momentum conservation equations, the transport term,

$$\mathbf{v} \cdot \nabla e, \tag{3.14}$$

as found in (3.3) is absent in the approximation of the energy conservation equation. The change in the internal energy of the cell on a 2d grid is approximated using the following centered finite difference scheme:

$$\frac{\partial e_{ij}}{\partial t} = -p_{ij}^n \left(\frac{\overline{u}_{i+1,j} - \overline{u}_{i-1,j}}{2\Delta x} + \frac{\overline{v}_{i,j+1} - \overline{v}_{i,j-1}}{2\Delta y} \right), \tag{3.15}$$

where $\overline{u}$ and $\overline{v}$ are average values based on the current velocity and the updated intermediate velocity values,

$$\overline{u}_{ij} = \frac{1}{2}(u_{ij}^* + u_{ij}^n) \tag{3.16}$$

$$\overline{v}_{ij} = \frac{1}{2}(v_{ij}^* + v_{ij}^n). \tag{3.17}$$

Harlow et al. [14] found that averaging the old and updated velocity values provided for increased stability and an increase in accuracy with regard to the behavior of fluid entropy. As was done in the updates to the cell velocities, an intermediary value for the specific internal energy of the cell is computed as follows:

$$e_c^* = e_c^n + \Delta t \frac{\partial e_c}{\partial t}. \tag{3.18}$$

In addressing the issue of how to compute the cell energy for a cell containing more than one type of material, two points of concern need to be addressed. The first is to choose a numerical approach that would best model the transport of energy across the interface between two different materials. The second is that the numerical approach should be convenient from a developer's perspective [14]. One approach would be to model the change in temperature as adiabatic compression, but for even simple models the equations can be moderately complex. A second approach is to apply the change in energy equally across all material types in the cell. A third alternative is to apply the change in specific internal energy to each material type [14].

3.2.2.5 Compute Intermediate Cell Values for Total Energy and Momentum

Intermediary cell values for the total energy and momentum are computed as follows:

$$\text{Total Energy}_c = m_c \left[e_c^* + \frac{1}{2}(\mathbf{v}_c^* \cdot \mathbf{v}_c^*) \right] \tag{3.19}$$

$$\mathbf{p}_c^* = m_c \mathbf{v}_c^* \tag{3.20}$$

where $\mathbf{p}_c^*$ is the cell momentum.

3.2.3 The Lagrangian Phase

In the Eulerian phase we saw particle how particle mass was used to compute cell densities from which the change in energy with respect to time and velocities were computed at cell centers. In the Lagrangian phase value particles are moved and cell values are updated.

3.2.3.1 Update Particle Position

The position of the particles is updated using the intermediary cell velocities,

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \sum_c \mathrm{S}_{cp}^n \mathbf{v}_c^*, \tag{3.21}$$

where S_{cp}^n are the values of the interpolating functions located at the cell-centers and evaluated at the particle positions, i.e.,

$$\mathrm{S}_{cp}^n = S_c(\mathbf{x}_p^n). \tag{3.22}$$

3.2.3.2 Update Cell Velocity and Internal Energy

The final step in the PIC method is to account for the change in cell velocity and energy as a particle moves from one cell to another. First, the new mass of the cell is computed,

$$m_c^{n+1} = N_c^{n+1} m_p. \tag{3.23}$$

Then, using the new cell mass, the updated cell velocity is computed as follows:

$$\mathbf{v}_c^{n+1} = \frac{1}{m_c^{n+1}} \mathbf{p}_c^*. \tag{3.24}$$

The specific internal energy is computed following a similar approach,

$$e_c^{n+1} = \frac{\text{Total Energy}_c}{m_{ij}^{n+1}} - \frac{1}{2}(\mathbf{v}_c \cdot \mathbf{v}_c). \tag{3.25}$$

3.3 PIC and Plasma Physics

Along with Harlow's work in the 1950s and 1960s developing the original PIC method for modeling fluid dynamics problems, the PIC method was also being used as a method for solving plasma physics problems. The pioneering work in this period using PIC methods for solving plasma physics problems was done by Buneman [78] and Dawson [79]. Due to the computational restraints of the time, the original work focused on one dimension problems but as computer power increased, plasma physics simulations were being done in two and three dimensions.

Two general approaches can be taken to model plasma. The first models the plasma as a fluid, and the second uses a kinetic model where the distribution function is determined by solving either the Boltzmann or Vlasov equation. For high temperature plasmas, the kinetic model approach is used [81].

The Vlasov equation for charged species acting under an electrostatic field is defined by [81],

$$\frac{\partial f_s}{\partial t} + \mathbf{v} \cdot \nabla f_s + \frac{q_s}{m_s} \tilde{\mathbf{E}} \cdot \nabla_{\mathbf{v}} f_s = 0, \tag{3.26}$$

where $f(\mathbf{x}, \mathbf{v}, t)$ is the distribution function for species s. The variables q and m are the charge and mass of the charged species, respectively; $\tilde{\mathbf{E}}$ is the electric field; and $\nabla_{\mathbf{v}}$ is the gradient operator with respect to the velocity components. In the case of a 3d problem, the gradient with respect to velocity would be

$$\nabla_{\mathbf{v}} = \begin{bmatrix} \frac{\partial}{\partial u} \\ \frac{\partial}{\partial v} \\ \frac{\partial}{\partial w} \end{bmatrix}. \tag{3.27}$$

The electric field, $\tilde{\mathbf{E}}$, is defined in terms of the electrostatic potential, ϕ, by the following relationship:

$$\tilde{\mathbf{E}} = -\epsilon \nabla \phi \tag{3.28}$$

where ϵ is the permittivity. In turn the electrostatic potential field is related to the charge density by Poisson's equation,

$$\nabla^2 \phi = -\frac{\tilde{\rho}}{\epsilon}. \tag{3.29}$$

The charge density field is related to the distribution functions of each of the species, s, in the following manner:

$$\tilde{\rho} = \sum_s q_s \int f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v}. \tag{3.30}$$

The preprocessing phase for the PIC method begins with assigning a charge and velocity to each particle. The initial particle velocity is given based on initial conditions. The charge quantity for each particle is based on the initial particle position, velocity, and volume as follows:

$$\text{charge}_p = f_0(\mathbf{x}_p, \mathbf{v}_p) v_p. \tag{3.31}$$

The basic steps for each time step are as follows: first, map the particle charges to the grid to determine the charge density. Second, solve for the electrostatic field on the grid using a Poisson solver of choice to solve (3.29). Third, approximate the gradients of the electrostatic field in order to approximate the electric field as defined by (3.28) on the grid. Fourth, interpolate the electric field to the particles. Fifth, update particle position and velocity based on a chosen numerical integration scheme. A common approach in PIC methods for plasma is to use a second-order leapfrog scheme

$$\mathbf{x}_p^{n+1} = \mathbf{x}^n + \Delta t \, \mathbf{v}_p^{n+1/2} \tag{3.32}$$

$$\mathbf{v}_p^{n+1/2} = \mathbf{v}_p^{n-1/2} + \Delta t \, \frac{q_p}{m_p} \tilde{\mathbf{E}}_p. \tag{3.33}$$

A key difference between the original PIC method developed by Harlow and the PIC methods in simulating plasma is that particles carry more information. In the Harlow version, mass is assigned to each particle, but properties such as velocity and energy are assigned to the grid. In plasma simulations quantities such as mass, charge, and velocity are assigned to the particles. The grid is used as a means to solve the given set of equations but now information is carried from one time step to the next. Any information that is assigned to the grid variable is discarded at the beginning of the next time step. For a more complete discussion of thy use of PIC in plasma physics simulations, the reader is refered to [11] and [82].

For the rest of this dissertation when the term PIC is used it will be used to denote a particle-in-cell methods where information such a mass, velocity, etc. is assigned to the particle and original PIC to denote Harlow's original particle-in-cell method.

3.4 FLIP – The Fluid Implicit Particle Method

Building upon the concepts that were being used in the plasma physics simulation community Brackbill et. al. introduced the fluid-implicit-particle (FLIP) method in the 1980s [15], [83]. FLIP, like other PIC methods, uses both Lagrangian particles and an Eulerian grid to model fluid flow. In the FLIP method, the physical quantities of mass, velocity, and internal energy are assigned to the particle. The grid is used solely for computational purposes and does not carry information from one time step to the next. Because the grid does not carry any information between time steps it can be adapted to arbitrary configurations at the beginning of each time step [15].

3.4.1 Preprocessing Phase

As is done in PIC methods the problem domain is discretized using a grid and a set of particles for each material. Based on the material type, the physical quantities of mass, velocity, internal energy, and constitutive properties are assigned to each particle consistent with the initial conditions. For example, given an even distribution of particles per cell, the particle mass can be derived in the following manner:

$$m_p = \frac{\rho^0 v_c}{N_c}. \tag{3.34}$$

3.4.2 The Basics of the Time Step Algorithm

In FLIP values on the grid can be either assigned to nodes or cell centers. The subscripts c, i, and p represent the indicies for cell centers, grid nodes, and particles, respectively. The basis functions located at the cell centers and evaluated at the particle position are represented by

$$\mathrm{S}_{cp} = S_c(\mathbf{x}_p) \tag{3.35}$$

and the basis functions located at the nodes, evaluated at the particle position are represented by

$$\mathrm{S}_{ip} = S_i(\mathbf{x}_p). \tag{3.36}$$

A description of the basic steps that are used to complete the calculation of one time step proceed as follows:

- **Map Particle Values to Grid:** The first step in the process is to compute the cell mass density and energy along with mapping particle velocities to cell nodes,

$$\rho_c = \frac{1}{V_c}\sum_p m_p S_{cp}, \tag{3.37}$$

$$I_c = \frac{1}{\rho_c V_c}\sum_p e_p S_{cp}, \tag{3.38}$$

$$m_i = \sum_p m_p S_{ip}, \tag{3.39}$$

$$\mathbf{v}_i = \frac{1}{m_i}\sum_p m_p \mathbf{v}_p S_{ip}, \tag{3.40}$$

  where ρ_c is the cell density and I_c specific internal energy of the cell.

- **Solve Conservation of Momentum and Conservation of Energy Equations:** Once particle values are mapped to the grid the conservation equation for momentum and energy can then be solved. The continuity equation is accounted for by means of assigning mass directly to the particles as was done in the original PIC method. In the method proposed by Brackbill et. al. [83], a finite volume menthod combined with backward Euler time integration scheme is used to compute the values on the grid. A complete description of the method can be found in [15].

- **Update Particle Values:** The key point to the FLIP method is that changes in particle values are interpolated from the grid to the particles and then particle values are updated. For velocity this would be

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \sum_i S_{ip}(\mathbf{v}_i^{n+1} - \mathbf{v}_i^n), \tag{3.41}$$

  The update to the particle position is then

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \sum_i S_{ip}\mathbf{v}_i^\theta, \tag{3.42}$$

  where

$$\mathbf{v}_i^\theta = \theta\mathbf{v}_i^{n+1} + (1-\theta)\mathbf{v}_i^n, \tag{3.43}$$

  with θ being chosen on the closed set $[0, 1]$.

3.4.3 FLIP's Improvements over Original PIC

The original PIC method suffered from excessive diffusion that came about from the averaging that was done as particles crossed from one cell to the next. By assigning mass,

velocity, and internal energy to the particles, the excessive diffusion that comes from averaging cell values is eliminated [84], [15]. The second key difference between original PIC and PIC methods in general is that FLIP interpolates the difference in a value to the particle and then updates the particle values verses updating the particle value based on the grid value [84].

3.5 The Material Point Method

The material point method (MPM) takes the concepts found in the FLIP method and extends them to solid mechanics problems. The original MPM was developed by Sulsky et al. [16, 17] in the mid 1990s. In many PIC methods the basic steps of the method are to first map values to the grid. Second, solve the governing equations using a method such as finite difference method. Third, interpolate updated grid values to the particle and update particle values. Finally, advect particles and reset grid values. MPM follows the same basic steps of other PIC methods but the formulation of the method, as originally presented by Sulsky et al. [16], is based upon a finite element formulation.

3.5.1 The Governing Equations

In it standard formulation, MPM is used to solve solid mechanics problems defined by the conservation of mass

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{v} = 0, \tag{3.44}$$

and conservation of linear momentum equations,

$$\rho \dot{\mathbf{v}} = \nabla \cdot \boldsymbol{\sigma} + \mathbf{b}, \tag{3.45}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress and $\mathbf{b}$ is the applied external force. For the purposes of this explaining the basic steps of MPM the following constitutive model will be used:

$$\boldsymbol{\sigma} = \mathbb{C} : \mathbf{e}, \tag{3.46}$$

$$\mathbf{e} = \frac{1}{2}\left(\mathbf{I} - \mathbf{F}^{-T}\mathbf{F}^{-1}\right), \tag{3.47}$$

where $\mathbb{C}$ is the 4th order stiffness tensor, $\mathbf{e}$ is the Euler-Almansi strain tensor, and $\mathbf{F}$ is the deformation gradient. The constitutive model can further be simplified by assuming the

material is a homogeneous and isotropic material. Based on the simplifying assumption the constitutive model can be reduced to the following:

$$\boldsymbol{\sigma} = 2G\mathbf{e}_{dev} + 3K\mathbf{e}_{vol}, \tag{3.48}$$

where G is the shear modulus and K is the bulk modulus. The strain tensor is decomposed into its deviatoric and volumetric components,

$$\mathbf{e}_{vol} = \frac{\mathrm{tr}(\mathbf{e})}{3}\mathbf{I}, \tag{3.49}$$

$$\mathbf{e}_{dev} = \mathbf{e} - \mathbf{e}_{vol}. \tag{3.50}$$

3.5.2 The General MPM Algorithm

There are different variations on the original MPM algorithm, but for the purpose of understanding the fundamentals of the method the description that follows will be based on the original work presented by Sulsky et al. [16] and descriptions of the method found in the following literature [85, 86].

3.5.2.1 Preprocessing Phase

The initial phase involved in the MPM algorithm is the preprocessing phase where the different material domains are discretized into a set of particle. Based on the division of each material domain, each particle is assigned a mass and initial volume based on the mass density and initial discretization of the domain. The values for velocity, stress, strain, and other necessary state variables are assigned to each particle base on the given set of initial conditions [85].

3.5.2.2 Time Step Algorithm

The steps of the time step algorithm are as follows:

1. **Map Particle Values to the Background Grid:** The mapping from particle to the grid nodes is done in the following manner:

$$m_i = \sum_p \phi_{ip} m_p, \tag{3.51}$$

$$\mathbf{p}_i^n = \sum_p \phi_{ip} \mathbf{v}_p^n m_p, \tag{3.52}$$

$$\mathbf{v}_i^n = \frac{1}{m_i}\mathbf{p}_i^n, \tag{3.53}$$

where m_p and m_i are the particle and nodal masses, respectively, $\mathbf{p}_i$ is the nodal momentum, $\mathbf{v}_p$ and $\mathbf{v}_i$ are the particle and nodal velocities, and ϕ_{ip} are values of the basis functions centered at the grid nodes and evaluated at the particle positions, $\phi_{ip} = \phi_i(\mathbf{x}_p)$. In this section the subscript i indicates a nodal value and subscript p indicates a particle value. Linear basis functions are a common first choice when implementing the MPM algorithm. For a one dimension problem, the linear basis function on a uniform grid with nodal spacing of h is shown in Fig. 3.1 and is defined as

$$\phi_i(x) = \begin{cases} 0, & x < x_i - h, \\ 1 + \frac{x - x_i}{h}, & x_i - h \leq x < x_i, \\ 1 - \frac{x - x_i}{h}, & x_i \leq x < x_i + h, \\ 0, & x \leq x_i + h. \end{cases} \tag{3.54}$$

2. **Compute the Internal Forces at the Grid Nodes:** The internal forces at the grid nodes are computed as follows:

$$\mathbf{f}_i^{int} = -\sum_p \nabla \phi_{ip} \cdot \sigma_p^n V_p^n, \tag{3.55}$$

where V_p is the current particle volume.

3. **Compute Nodal Acceleration:** The nodal acceleration is computed using both internal and external forces,

$$\mathbf{a}_i^n = \frac{\mathbf{f}_i^{int} + \mathbf{f}_i^{ext}}{m_i}. \tag{3.56}$$

4. **Update Nodal Velocity:** The nodal velocities are updated using an explicit Euler time integration scheme,

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \Delta t \mathbf{a}_i^n. \tag{3.57}$$

5. **Compute Velocity Gradients** Velocity gradients are computed for each particle. The velocity gradient is the tensor product of the updated nodal velocity and the gradient of the basis functions evaluated at the particle position,

$$\nabla \mathbf{v}_p^{n+1} = \sum_i \mathbf{v}_i^{n+1} \otimes \nabla \phi_{ip}. \tag{3.58}$$

6. **Update Deformation Gradients:** Deformation gradients are updated using an explicit Euler time integration and the newly computed velocity gradients,

$$\mathbf{F}_p^{n+1} = \left(\mathbf{I} + \nabla \mathbf{v}_p^{n+1} \Delta t\right) \mathbf{F}_p^n. \tag{3.59}$$

7. **Update Particle Velocity and Position:** The particle velocities and positions are updated using an explicit Euler time integration and the updated nodal velocities and accelerations,

$$\mathbf{v}_p^{n+1} = \mathbf{v}_p^n + \delta t \sum_i \phi_{ip} \mathbf{a}_i^n, \tag{3.60}$$

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \delta t \sum_i \phi_{ip} \mathbf{v}_i^{n+1}. \tag{3.61}$$

8. **Update Particle Volumes:** Particle volumes are updated as follows:

$$V_p^{n+1} = J V_p^0, \tag{3.62}$$

where V_p^0 is the initial particle volume and J is the determinant of the updated deformation gradient,

$$J = \det\left(\mathbf{F}_p^{n+1}\right). \tag{3.63}$$

9. **Update Particle Stresses** Updated particle stresses computed based on the given constitutive model. In the case of the constitutive model defined in Section 3.5.1 the steps to compute the updated particle stress would proceed in the following order:

   (a) $\mathbf{e}_p^{n+1} = \frac{1}{2}\left(\mathbf{I} - (\mathbf{F}^{n+1})^{-T}(\mathbf{F}^{n+1})^{-1}\right)$.

   (b) $\mathbf{e}_{p,vol}^{n+1} = \frac{\mathrm{tr}\left(\mathbf{e}_p^{n+1}\right)}{3}\mathbf{I}$.

   (c) $\mathbf{e}_{p,dev}^{n+1} = \mathbf{e}_p^{n+1} - \mathbf{e}_{p,vol}^{n+1}$.

   (d) $\boldsymbol{\sigma}_p^{n+1} = 2G\mathbf{e}_{p,dev}^{n+1} + 3K\mathbf{e}_{p,vol}^{n+1}$.

3.6 Key Advances in MPM's Development

Since its original development MPM has been under continual development with many new insights and improvements to the method over the years. Bardenhagen and Kober developed the generalized interpolation material point method (GIMP), which improves upon the original method by generalizing the formation of the grid weighting functions such that

$$\phi_i(\mathbf{x}) = \frac{1}{V_p}\int \chi_p(\mathbf{x}) S_i(\mathbf{x}) dV, \tag{3.64}$$

where $\chi_p(\mathbf{x})$ is the particle domain and $S_i(\mathbf{x})$ is the grid basis function located at node i [87]. If $S_i(\mathbf{x})$ is a linear basis function and $\chi_p(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_p)$ then $\phi_i(\mathbf{x})$ resolves to the original MPM weighting function.

The convected particle domain interpolation technique (CPDI) developed by Sadeghirad and Brannon represents particle domains as parallelograms in 2d and parallelepipeds in 3d. The particle domain shape is updated according to the particle deformation gradient and the deformation gradient is considered constant across the particle domain [86]. The second-order convected particle domain interpolation (CPDI2) developed by Sadeghirad, Brannon, and Guilkey eliminates the overlaps or gaps that can occur with CPDI [88].

B-splines have become a popular choice for grid weighting functions. Steffen et al. explored the use of B-splines in their analysis work on MPM [89]. Stomakhin et al. [90] and Gan et al. [91] have presented b-spline based MPM formulations. Least squares approaches have also been adopted for MPM. Wallstedt and Guilkey developed a weighted least squares variation of MPM [92] and moving least variations of MPM have been developed by Edwards and Bridson [93], Hu et al. [94], Tran et al. [95] and Song and Kim [96].

The approach to how information is moved between the particles and grid is another area of focus in MPM research. Wallstedt and Guilkey incorporated the velocity gradient values, which are computed for each particle, when computing velocity values at the grid nodes [97]. The affine-particle-in-cell (APIC) method developed by Jiang et al. [98] and the angular conserving affine-particle-in-cell method [99], which improves upon the original APIC method, uses an affine transfromation in the particle to grid mapping that has been shown to conserve angular momentum.

From this small survey of the literature, it can be seen that MPM continues to be a topic of interest to the research community and because of its unique strengths it is continuuing to gaining greater adoption.

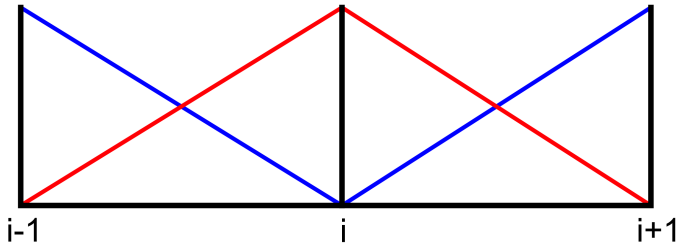

Fig. 3.1. One-dimension linear basis functions.

CHAPTER 4

NULLSPACE FILTERS AND CORRECTED DERIVATIVES FOR MPM

The particle-in-cell (PIC) [14] method, fluid implicit particle method (FLIP) [15], and the material point method (MPM) [16], as described in Chapter 3, use moving particles whose dynamics are defined by calculating quantities such as acceleration on a fixed background grid. The main idea in such methods is to make use of basis functions at particles (originally delta functions) and basis functions at nodal grid points (originally linear basis "hat" functions). MPM has been much studied since it was introduced, and there are many examples of papers that improve the performance of the method. Examples of such papers are the improved basis functions for MPM derived by [87, 86, 88, 100] and the higher order basis functions derived by [21, 101]. Many of these papers produce improved results by developing methods that help reduce the grid crossing error that occurs when particles cross a grid cell boundary. In general the combination of moving Lagrangian particles and a fixed Eulerian grid used by PIC and MPM is successful on many challenging problems; however, many theoretical issues to do with such methods remain at least partially unresolved in the areas of stability, accuracy, and convergence.

One such issue is that the PIC method is well-known, e.g. [102, 103], to have an aliasing error due to the difference between the degrees of freedom at the grid points of the spatial mesh cell compared to the degrees of freedom at the particles. This error may result in oscillatory solution values. For example Brackbill [103] states that, "Because the number of particles is finite, the number of Fourier modes is also finite. Thus, when there are n particles in each cell, there are n times as many Fourier modes as there are grid points." When values are mapped from nodes to particles the lack of resolution at the nodes compared to resolution at the particles can cause an aliasing error. Again to quote Brackbill, "Aliases occur because all Fourier modes with wavelengths shorter than the grid spacing are indistinguishable at

the grid points" [103]. This aliasing is exactly the null space error addressed in this work. Brackbill showed that the user of better interpolants associated with the nodes to particles mapping helped reduce these errors and conducted a Fourier analysis of the problem. Early attempts to address this instability started from the PIC jiggling work of Langdon [102] and Chen, Langdon and Birdsall [104] to Brackbill and Lapenta [105]. The jiggling approach artificially moves the points and results in reduced oscillations.

An alternative approach is to use the conservation of energy to improve the performance of PIC and MPM. While conservation of energy does not necessarily itself imply stability [106], it is a desirable property for a numerical method. Examples of work to improve the accuracy and stability of PIC methods by introducing energy conservation are Lapenta [107] and Brackbill [108] address energy and momentum conservation properties of PIC applied to plasma calculations. At the same time it is not entirely clear how the approach used with PIC methods applied to plasma calculations may be used with MPM. Energy conservation in MPM is addressed by Bardenhagen [109], and Love and Sulsky provide energy consistent and conserving approaches [110, 111]. Finally the work of Mast et al. [112] addresses the locking phenomenon observed originally in finite element methods in the context of MPM. Locking is a phenomenon that is different to the grid crossing error mentioned above and is addressed by multidimensional filtering of stresses and strains. However, while these papers greatly improve the accuracy and conservation properties of PIC and MPM and indirectly result in more stable methods they do not directly address the nullspace issue in MPM. Part of the challenge with undertaking analysis of MPM as Wallstedt and Guilkey point out is its nonlinearity [113] and the fact that moving particles may dynamically change the effective computational stencils frequently.

Similar issues of stability due to moving particles arise in smooth particle hydrodynamics (SPH) and in the many finite element formulations of particle methods. For examples of the related work on SPH [114, 115, 116] and finite element methods, see [117, 118, 119]. Belytschko and Xaio [119] address two sources of instability of their particle methods. The first one is a rank deficiency of the discrete equations that is similar to the ringing instability or the nullspace problem considered here and the second is a distortion of the material instability. The work on these methods deals with continually moving points without having a background grid as is used in MPM. For this reason it is not at all clear how the body

of work introduced above immediately relates to MPM with its mixture of Lagrangian particles and an Eulerian background grid. The general approaches adopted clearly have some potential application to MPM however.

There has been resent work on both linear and non-linear stability analysis for MPM. Bai and Schroeder [120] formulate a stability analysis of explicit MPM using Von Neumann stability analysis. Ni and Zhang [121], propose a stable time stepping criterion based on linear analysis for explicit MPM. Berzins [122] provides a nonlinear stability analysis for MPM based on the work of Spigler and Vianello [123] along with time stepping criterion for the update stress last version of MPM [109].

In previous work we began to address the general area of spurious oscillations in PIC [124, 125] based on a matrix approach that directly reflects the fact that there are nonzero values at the particles that cannot be "seen" at the nodes and the space of such values is oscillatory. The positive results that were obtained have provided encouragement to expand these ideas to MPM. As the intention here is to shed some light upon this issue for MPM, attention is focused on one-dimensional (1d) problems to add clarity. Sections 4.1 and 4.2 describes MPM. Section 4.3 addresses the nullspace that results from the difference between the number of particles and the nodal grid points points that they are mapped to. While a global approach based on a singular value decomposition for addressing this was given in [124, 125], Section 4.4 defines a method based upon a local calculation using values in the elements on either side of a grid node. Finally, Section 4.7 describes a numerical experiment that illustrates the benefits of this approach and compares it against the method of [100] as well as some of the challenges that remain.

4.1 Material Point Method

The model problem used here is a pair of equations connecting velocity v, displacement u and density ρ and is typical of straightforward MPM applications:

$$\frac{Du}{Dt} = v, \tag{4.1}$$

$$\rho\frac{Dv}{Dt} = \frac{\partial\sigma}{\partial x} + b, \tag{4.2}$$

with a linear stress model $\sigma = E\frac{\partial u}{\partial x}$ for which Young's modulus, E, is constant, a body force b and with appropriate boundary and initial conditions. In describing MPM the starting

point is to assume that there is a mesh of $N+1$ fixed nodes X_i on some fixed interval $[a,b]$ such that

$$a = X_0 < X_1 < ... < X_N = b. \tag{4.3}$$

The mesh has N elements with the interval I_i defined by

$$I_i = [X_i, X_{i+1}], \tag{4.4}$$

and the width of each interval is denoted by

$$h_i = X_i - X_{i-1}. \tag{4.5}$$

A fixed evenly spaced mesh is used here with $h = h_i \forall i$.

It is assumed that there are m particles between each pair of nodes, situated at x_p^n points where at each time step, $t^n = \delta t * n$, where n is the nth time step, and the computed solution at the pth particles will be written as $u_p^n = u(x_p^n, t^n)$. The superscript n and the dependence on t will often be dropped unless they are necessary to clarify timestep issues. The restriction of attention to equal numbers of particles in each interval is used solely to simplify the algebra in the analysis that follows and imposes no constraint on the approaches discussed. In this case the particles in interval i lie between X_i and X_{i+1} and have positions $x_{im+j}, j = 1, .., m$.

4.1.1 Mapping Matrix for Solution Values

In both the MPM and PIC grid-based particle methods it is necessary to map values from the particles to the nodes and from the nodes to the particles [16, 87]. For example, in mapping particles to the node X_i and back to particles, the linear basis functions centered at that point and given by $\phi_i(x)$

$$\phi_i(x) = \frac{x - X_i}{X_i - X_{i-1}}, X_{i-1} \leq x \leq X_i, \tag{4.6}$$

$$\phi_i(x) = \frac{X_i - x}{X_{i+1} - X_i}, X_{i+1} \geq x \geq X_i. \tag{4.7}$$

$$\phi_i(x) = 0, x \notin [X_{i-1}, X_{i+1}] \tag{4.8}$$

are often used. There are, of course, many alternative choices [87, 86, 88, 100, 21, 101].

In the case of the basis functions associated with particles, there are a number of possibilities ranging from delta functions to approaches such as [87]. In general given a

function value $u_p(t)$ at a particle x_p, the representation used in MPM is given by $u_p(t)\chi_p(x,t)$ where the basis functions $\chi_p(x,t)$ in the case of delta functions $\delta(x-x_p)$ are given by

$$\chi_p(x,t) = \delta(x-x_p)V_p \tag{4.9}$$

where V_p is the width in one dimension of the particle as defined below. In two dimensions V_p is the area of the particle and in three dimensions its volume. Bardenhagen points out [87] that this is not to be a partition of unity in that

$$\sum_p \chi_p(x,t) \neq 1. \tag{4.10}$$

As a result other representations such as piecewise constant functions

$$\chi_p(x,t) = 1, x \in V_p, \tag{4.11}$$

$$\chi_p(x,t) = 0, x \notin V_p, \tag{4.12}$$

or even more complex basis functions such as [86, 88, 21, 101] are used. In general the volume of the pth particle V_p is defined as

$$V_p = \int_a^b \chi_p(x,t)dx. \tag{4.13}$$

The particle volumes are initially defined to span the interval $[a,b]$ and are then modified as the particles move, as is shown in the next section.

The mapping from particles to nodes in MPM takes into account all the particles in adjacent intervals to the node in question (X_i for example) and may be written, regardless of the choice of particle basis, as

$$U_i(t) = \sum_p S_{ip}u_p(t). \tag{4.14}$$

where $U_i(t)$ is the value at the node X_i. The argument (t) will often be dropped unless needed for clarity. From [87, 89] the mapping constants are given by

$$S_{ip} = \frac{1}{\int_a^b \chi_p(x,t)dx}\int_a^b \chi_p(x,t)\phi_i(x)dx. \tag{4.15}$$

In the case of linear basis functions and particles represented by Equation (4.9), it follows that

$$S_{ip} = \phi_i(x_p). \tag{4.16}$$

This mapping may be modified in two ways. The first modification is that a mass weighting is often used when velocity is mapped, e.g. [113], see Equation (4.40) below. The second

modification is that the S_{ip} multipliers correspond to a partition of unity by dividing by the sum of all the components that contribute to node X_i using the modified mapping coefficients S^*_{ip} defined by

$$S^*_{ip} = \frac{\phi_i(x_p)}{W_i} \tag{4.17}$$

where $W_i = \sum_{p \in I_{i-1} \cup I_i} \phi_i(x_p)$, such that

$$\sum_p S^*_{ip} = 1. \tag{4.18}$$

While this mapping from particles to nodes can be expressed in terms of a system-wide matrix, which was adopted in [124, 125], for this chapter a local version of the mapping from the set of particles in the two elements on either side of a node to that node will be used. This local mapping is represented as

$$U_i = \mathbf{S}^*_{ip} \mathbf{u}_{ip}, \tag{4.19}$$

where U_i is the mapped value at the node X_i and $\mathbf{u}_{ip}$ is the vector of particle values in the two elements adjacent to X_i. Thus, as there are $2m$ particles in the two adjacent elements the matrix $\mathbf{S}^*_{ip}$ has only one row and $2m$ columns with the value in the jth column being given by $S^*_{i((i-1)m+j)}$. As the entries of the row vector $\mathbf{S}^*_{ip}$ are positive, it is straightforward to construct vectors consisting of non-zero entries, say $\mathbf{v}_{ip}$ that give rise to zero values at the nodes. In other words

$$\mathbf{S}^*_{ip} \mathbf{v}_{ip} = 0, \tag{4.20}$$

This is the null space problem in our previous work [124, 125] and is also, when considered for gradient mappings, termed a high-frequency instability that arises from the rank deficiency of the discrete divergence operator [117]. This topic will be examined in detail in below.

4.1.2 Internal Force Calculation

The calculation of the internal forces in MPM at the nodes requires the calculation of the volume integral of the divergence of the stress [113]. The divergence of the stress, σ, at a node X_i, as denoted by $\frac{\partial \sigma_i}{\partial x}$ given the stress at particles σ_p. After integration by parts, the force calculation is written as

$$f^{int}_i = -\sum_p D S_{ip} \sigma_p V_p \tag{4.21}$$

where in the case of linear functions at the nodes from [87] in the case when Delta functions are used to represent particles it follows that

$$DS_{ip} = \frac{d\phi(x_p)}{dx}. \tag{4.22}$$

In the same way that the mapping coefficients S^*_{ip} are modified to be a partition of unity, the coefficients DS_{ip} are amended to reproduce derivatives of constant and linear functions exactly. In order to reproduce linear functions exactly, a derivative correction is needed [126, 118]. Multiple methods have been proposed to correct for this deficiency [13, 114, 115, 127]. In one space dimension, the modified coefficients for the derivative of a constant to be zero are given by [13],

$$DS^{**}_{ip} = \frac{1}{\sum_{pp}\phi_i(x_{pp})}\frac{d\phi_i(x_p)}{dx} - \frac{\phi_i(x_p)}{(\sum_{pp}\phi_i(x))^2}\sum_{pp}\frac{d\phi_i(x_{pp})}{dx}, \tag{4.23}$$

where the subscript pp indicates the set of particles within the support of the basis functions for node i. From this it follows immediately that

$$\sum_p DS^{**}_{ip} = 0. \tag{4.24}$$

A constant C is then used to ensure that differentiating the function x at the particles give the correct value of one at the node X_i

$$C\sum_p DS^{**}_{ip}x_p = 1. \tag{4.25}$$

The correction thus is defined [119] by

$$C = \frac{1}{\sum_p x_p DS^{**}_{ip}}. \tag{4.26}$$

Therefore if the mapping coefficients are defined by

$$DS^*_{ip} = \frac{1}{\sum_{pp} x_{pp} DS^{**}_{ipp}}\cdot\left[\frac{1}{\sum_{pp}\phi_i(x_{pp})}\frac{d\phi_i(x_p)}{dx} - \frac{\phi_i(x_p)}{(\sum_{pp}\phi_i(x))^2}\sum_{pp}\frac{d\phi_i(x_{pp})}{dx}\right]. \tag{4.27}$$

The force calculation is given by

$$f^{int}_i = -\sum_p DS^*_{ip}\sigma_p V_p \tag{4.28}$$

It should also be noted that this is essentially a derivative approximation in that the derivative at a node $dy/dx(X_i)$ of a function whose values at particles are given by y_p is approximated by

$$\frac{dy}{dx}(X_i) = -\sum_p DS^*_{ip} y_p V_p \tag{4.29}$$

This form of computing the values of the internal force at the nodes varies from the traditional MPM approach, which is defined in the variational form. The above method for computing nodal forces is focused on ensuring that the gradients of weighting functions meet the criteria for the reproducibility of constant and linear functions. This method is only applied to the nodal force calculation in this chapter. A consistent approach would be to apply the method for all particle to node mappings as is done for moving least squares approaches to MPM [94].

4.1.3 Interpolating Back to Particles

Once the nodal values of either the solution or the derivative are calculated, then by using the piecewise linear basis functions, $\phi_i(x)$, associated with the mesh nodes, a linear approximation to the function $u(x)$ at a particle may be defined as

$$u_p = \sum_i U_i \phi_{ip}. \tag{4.30}$$

where $\phi_{ip} = \phi_i(x_p)$. It is convenient to write this mapping as a matrix operation. In the case of linear basis functions the individual mappings have the form given by

$$u_{m(i-1)+j} = U_{i-1}\alpha_{m(i-1)+j} + U_i(1 - \alpha_{m(i-1)+j}), \tag{4.31}$$

$$u_{mi+j} = U_i\alpha_{mi+j} + U_{i+1}(1 - \alpha_{mi+j}), j = 1, ..., m \tag{4.32}$$

where

$$\alpha_{m(i-1)+j} = \frac{X_i - x_{m(i-1)+j}}{h_i}, \tag{4.33}$$

$$\alpha_{mi+j} = \frac{X_{i+1} - x_{mi+j}}{h_{i+1}}, j = 1, ..., m. \tag{4.34}$$

The grid to particle mapping may be written as a matrix equation to define the values at all the points in $[X_{i-1}, X_{i+1}]$.

$$\mathbf{u}_{ip} = \hat{\mathbf{M}} \begin{bmatrix} U_{i-1} \\ U_i \\ U_{i+1} \end{bmatrix}, \tag{4.35}$$

where the mapping matrix $\hat{\mathbf{M}}$ is defined by

$$\hat{\mathbf{M}} = \begin{bmatrix} \alpha_{m(i-2)+1} & (1-\alpha_{m(i-2)+1}) & 0 \\ \vdots & \vdots & \vdots \\ \alpha_{m(i-2)+m} & (1-\alpha_{m(i-2)+m}) & 0 \\ 0 & \alpha_{m(i-1)+1} & (1-\alpha_{m(i-1)+1}) \\ \vdots & \vdots & \vdots \\ 0 & \alpha_{mi} & (1-\alpha_{mi}) \end{bmatrix}$$

and where the vector $\mathbf{u}_{ip}$ is defined as in Equation (4.19). The mapping to construct derivative values at the particle points is similar and is defined by

$$\frac{\partial u_{m(i-1)+j}}{\partial x} = (U_i - U_{i-1})/(X_i - X_{i-1}), \tag{4.36}$$

$$\frac{\partial u_{mi+j}}{\partial x} = (U_{i+1} - U_i)/(X_{i+1} - X_i), j = 1, ..., m, \tag{4.37}$$

and, in the case of evenly-spaced nodes, gives rise to a similar form of the mapping matrix.

$$\hat{\mathbf{M}}_{\mathbf{D}} = \frac{1}{h} \begin{bmatrix} -1 & 1 & 0 \\ \vdots & \vdots & \vdots \\ -1 & 1 & 0 \\ 0 & -1 & 1 \\ \vdots & \vdots & \vdots \\ 0 & -1 & 1 \end{bmatrix}$$

and a similar equation to calculate the derivatives at the particles as in Equation (4.35).

$$\frac{\partial}{\partial x}\mathbf{u}_{ip} = \hat{\mathbf{M}}_{\mathbf{D}} \begin{bmatrix} U_{i-1} \\ U_i \\ U_{i+1} \end{bmatrix}. \tag{4.38}$$

4.2 Material Point Method

The description of MPM given here follows that of a number of authors such as [113]. The first two steps in MPM are to compute the mass and velocity values at the nodes. Masses at the particles m_p are mapped to the nodes as follows:

$$m_i = \sum_p S^*_{ip} m_p. \tag{4.39}$$

Velocity is similarly mapped to the nodes as follows:

$$v_i^n = \sum_p \hat{S}_{ip} v_p^n, \tag{4.40}$$

where the coefficients of the mass-weighted mapping are $\hat{S}_{ip} = S_{ip}^* \frac{m_p}{m_i}$. Forces at the nodes are computed by using Equation (4.21)

$$f_i^n = -\sum_p D S_{ip}^* \sigma_p^n V_p^n + b_i \tag{4.41}$$

where σ_p^n is the stress at particle p, b_i is the body force at the node and V_p^n is the volume of that particle. The nodal acceleration is then computed using the nodal mass and force,

$$a_i^n = \frac{f_i^n}{m_i}. \tag{4.42}$$

With the acceleration computed at the node, nodal velocities can now be updated using a Euler-forward time stepping scheme,

$$v_i^{n+1} = v_i^n + a_i^n \delta t. \tag{4.43}$$

Using the updated nodal velocity, the velocity gradients are then computed at the particles using the gradient of the interpolating function,

$$\frac{\partial v_p^{n+1}}{\partial x} = \sum_p \frac{\partial \phi_{ip}}{\partial x} v_i^{n+1}. \tag{4.44}$$

The deformation gradient at particle p is updated as follows:

$$F_p^{n+1} = F_p^n + \frac{\partial v_p^{n+1}}{\partial x} F_p^n \delta t. \tag{4.45}$$

Stress is updated using the appropriate constitutive model. In this case using the velocity gradient, $\frac{\partial v_p^{n+1}}{\partial x}$, and Young's Modulus, E,

$$\sigma_p^{n+1} = \sigma_p^n + dt E \frac{\partial v_p^{n+1}}{\partial x}. \tag{4.46}$$

The particle velocity and position is updated by interpolating nodal values to the particles,

$$v_p^{n+1} = v_p^n + \sum_p \phi_{ip} a_i^n \delta t, \tag{4.47}$$

$$x_p^{n+1} = x_p^n + \sum_p \phi_{ip} v_i^{n+1} \delta t. \tag{4.48}$$

Finally the particle volumes are updated using the determinant of the deformation gradient, J, and the initial particle volume,

$$V_p^{n+1} = J V_p^0. \tag{4.49}$$

In one dimension the determinant of the deformation gradient is $J = F_p^{n+1}$.

4.3 Null Space of the Mapping Vector

The general mapping matrix $\mathbf{S}_{ip}$ maps from all particles to nodes and so has $N+1$ rows and $m \times N$ columns, [124, 125]. While the global SVD analysis used in [124, 125] applies to MPM considered here, it is prohibitively expensive to use as an algorithmic approach for large meshes and more than one space dimension. For this reason while a global SVD method is a useful comparison tool in one space dimension, it is important to consider local mappings from particles to each node and back again independently. In what follows the standard mapping matrix to the nodes defined by the coefficients S^*_{ip} is used. The approach could be applied equally well to any mapping matrix. In this local mapping case the mapping matrix reduces to a row-vector, but still has an SVD decomposition and a nontrivial null space. For example let $\mathbf{c}$ be a vector in $\mathbb{R}^{2m}$. If $\mathbf{S}^*_{ip}\mathbf{c} = \mathbf{0}$, then we say that $\mathbf{c}$ is in the null space of $\mathbf{S}^*_{ip}$. We can determine the null space of $\mathbf{S}^*_{ip}$ by making use of its singular value decomposition, SVD [128]. Taking the SVD of $\mathbf{S}^*_{ip}$ gives the following decomposition:

$$\mathbf{S}^*_{ip} = \hat{U}_{svd}\hat{\mathbf{S}}\mathbf{V}^T, \tag{4.50}$$

where $\hat{U}_{svd}$ is a scalar that is assumed to have value one, $\hat{\mathbf{S}}$ is 1 by $2m$, and $\mathbf{V}$ is $2m$ by $2m$.

In the global case in contrast $\hat{\mathbf{U}}_{svd}$ is a matrix of size $(N+1)x(N+1)$, $\hat{\mathbf{S}}$ is $N+1$ by $m \times N$, and $\mathbf{V}$ is $m \times N$ by $m \times N$. While all the remaining analysis will be for the local nodal case, the full global form of the SVD as used by [124, 125] will be used below to motivate the local approach used here.

In both local and global cases the matrix $\mathbf{V}$ is unitary, meaning that the columns are orthonormal [129]. In other words, if $\mathbf{v}_i$ and $\mathbf{v}_j$ are columns of the matrix V, then,

$$\mathbf{v}_i^T\mathbf{v}_j = \delta_{ij}, \tag{4.51}$$

where δ_{ij} is the Kroenecker delta. The columns of $\mathbf{V}$ are orthogonal, linearly independent and span the space $\mathbb{R}^{2m}$, [129]. The matrix $\hat{\mathbf{S}}$ is an 1 by $2m$ matrix of the form

$$\hat{\mathbf{S}} = \begin{bmatrix} \hat{\sigma}_i & 0 & \dots & 0 & 0 & \dots & 0 \end{bmatrix}. \tag{4.52}$$

Taking the matrix product of $\hat{\mathbf{S}}$ and $\mathbf{V}^T$ gives

$$\hat{\mathbf{S}}\mathbf{V}^T = \left(\mathbf{V}\hat{\mathbf{S}}^T\right)^T = \left(\left[\begin{array}{c|c|c|c} \hat{\sigma}_i\mathbf{v}_1 & 0 * \mathbf{v}_2 & \dots & 0 * \mathbf{v}_{2m} \end{array} \right] \right). \tag{4.53}$$

Consequently, the column vectors $\mathbf{v}_2$ to $\mathbf{v}_{2m}$ span the null space of $\mathbf{S^*_{ip}}$. Since the columns of $\mathbf{V}$ are orthogonal, they form a basis for $\mathbb{R}^{2m}$, which means that any vector $c \in \mathbb{R}^{2m}$ can be expressed as a linear combination of the columns of $\mathbf{V}$,

$$\mathbf{c} = c_1 \begin{bmatrix} \\ \mathbf{v}_1 \\ \\ \end{bmatrix} + \underbrace{c_2 \begin{bmatrix} \\ \mathbf{v}_2 \\ \\ \end{bmatrix} + c_3 \begin{bmatrix} \\ \mathbf{v}_3 \\ \\ \end{bmatrix} + \cdots + c_{2m} \begin{bmatrix} \\ \mathbf{v}_{2m} \\ \\ \end{bmatrix}}_{null(S_{ip^*})}. \tag{4.54}$$

A portion of $\mathbf{c}$ is thus in the null space of $\mathbf{S}^*_{ip}$ if $c_j \neq 0$ for some $j = 2, .., 2m$. Using the inner product [129] allows the components of a vector that are in the null space to be found. As the vectors $\mathbf{v_i}$ are a basis it follows that

$$\mathbf{u}_{ip} = \sum_{i=1}^{2m} (\mathbf{v}_i \cdot \mathbf{u}_{ip}) \mathbf{v}_i. \tag{4.55}$$

where $\mathbf{u}_{ip}$ is defined by Equation (4.19). A similar equation exists in the case of the full global form of the SVD matrix [124, 125]. In the local case considered here, the vectors $\mathbf{v_2}$ to $\mathbf{v_{2m}}$ span the null space of the mapping from particles to a node. From this it follows that a filtered form of the vector $\mathbf{u}_{ip}$ with the null space elements removed is denoted by $\mathbf{u_{ip}}^F$ and is defined by

$$\mathbf{u}^F_{ip} = (\mathbf{v}_1 \cdot \mathbf{u}_{ip}) \mathbf{v}_1, \tag{4.56}$$

or by using the SVD decomposition of $\mathbf{S}^*_{ip}$ to get

$$U_i = \hat{\sigma} \mathbf{v}_1^T \mathbf{u}_{ip}. \tag{4.57}$$

Equation (4.19) in terms of the nodal value U_I as

$$\mathbf{u}^F_{ip} = \frac{U_i}{\hat{\sigma}} \mathbf{v}_1. \tag{4.58}$$

The portion of any given vector that lies in the null space of $\mathbf{S}^*_{ip}$ can be found as follows: as the vector $\mathbf{S}^*_{ip}$ can be written in terms of the singular value components,

$$\begin{aligned} \mathbf{S}^*_{ip} &= \pm 1 \left[\hat{\sigma}, 0, 0, \cdots, 0 \right] \left[\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \cdots, \mathbf{v}_n \right]^T \\ &= \pm 1 (\hat{\sigma_i} \mathbf{v}_1 + 0\mathbf{v}_2 + 0\mathbf{v}_3 + \cdots + 0\mathbf{v}_n)^T \\ &= \pm 1 \hat{\sigma_i} \mathbf{v}_1^T. \end{aligned} \tag{4.59}$$

Taking the dot product of the vector $\mathbf{S}^*_{ip}$ with itself gives

$$\begin{aligned}\mathbf{S}^*_{ip} \cdot \mathbf{S}^*_{ip} &= \hat{U}_{svd}\hat{\mathbf{S}}\mathbf{V}^{\mathbf{T}}(\hat{U}_{svd}\hat{\mathbf{S}}\mathbf{V}^{\mathbf{T}})^{\mathbf{T}}, \\ &= \hat{U}_{svd}\hat{\mathbf{S}}\mathbf{V}^{\mathbf{T}}\mathbf{V}^{\mathbf{T}}\hat{\mathbf{S}}^{\mathbf{T}}\hat{U}^T_{svd}, \\ &= \hat{U}_{svd}\hat{\mathbf{S}}\hat{\mathbf{S}}^{\mathbf{T}}\hat{U}^T_{svd}, \\ &= (\hat{\sigma}_i)^2. \end{aligned} \tag{4.60}$$

From the vector dot product above it can be seen that the singular value for the mapping defined by $\mathbf{S}^*_{ip}$ is $\hat{\sigma} = \sqrt{(\mathbf{S}^*_{ip} \cdot \mathbf{S}^*_{ip})}$. Using the above two observations the column vector $\mathbf{v}_1$ can be found as follows:

$$\mathbf{v}_1^T = \frac{1}{\sqrt{(\mathbf{S}^*_{ip} \cdot \mathbf{S}^*_{ip})}}\mathbf{S}^*_{ip}. \tag{4.61}$$

Given the vector $\mathbf{v}_1$, the part of $\mathbf{u}_p$ that lies in the the null space of the mapping vector $\mathbf{S}^*_{ip}$ is then given by $\mathbf{u}_{ipnull}$ is then analogously as in the full SVD version and is given by

$$\mathbf{u}_{ipnull} = \mathbf{u}_{ip} - (\mathbf{v}_1 \cdot \mathbf{u}_{ip})\mathbf{v}_1 \tag{4.62}$$

or as

$$\mathbf{u}_{ipnull} = \mathbf{u}_{ip} - \frac{U_i}{\hat{\sigma}_i{}^2}(\mathbf{S}^*_{ip})^T \tag{4.63}$$

4.3.1 Numerical Example of Nullspace

The following example shows the null spaces for both the derivative mappings and the solution mappings, in the case of the original MPM with a linear basis at the nodes and delta functions at the particles. The null space vectors for both the derivative mappings and the solution mappings are shown in the case of two cells adjacent to a node have four particles per cell. The cell width is 1 and the spatial domain of the problem goes from -1 to 1. The values at the particles is defined by the function,

$$u(x) = e^x. \tag{4.64}$$

Using the function a vector of particle values is computed where $u_p = u(x_p)$. The vector of mapping weights are computed about the node at position 0.0, $S_{ip} = \phi_i(x_p)$. Fig. 4.1 shows the values at the particles and the portions of the particles values that lie in the null space of the nodal mapping (4.19) and of the derivative mapping (4.28) using linear basis functions at the nodes and delta functions at the particles.

Fig. 4.1 shows the single oscillation about the node. Other more oscillatory examples of null space components occur with symmetric positive S^*_{ip} values. In this case for each pair of S^*_{ip} values one solution component may be positive in one interval, and the other solution value in the other interval may be negative. In the case of the mapping DS^*_{ip}, associated with derivatives, as the mapping values will have different signs, the null space could still be oscillatory but may not necessarily change sign.

4.3.2 Mapping to Particles Reinforces the Nullspace

Equation (4.19) maps particle values to the nodes. At this point, the null space component of $\mathbf{u}_p$ has been removed by the nature of the mapping. It is at the next step in the computation that a null space component can be re-introduced. From Equations (4.55 to 4.62) it follows that if for any $\mathbf{v_i}$ for $i = 2, ..., 2m$ the following is true

$$\mathbf{v_i^T}\hat{\mathbf{M}} \begin{bmatrix} U_{i-1} \\ U_i \\ U_{i+1} \end{bmatrix}, \quad \mathbf{v_i} \neq 0, \quad i = 2, ..., 2m \tag{4.65}$$

then the vector $\mathbf{u}_{ip}$, as defined by the mapping in Equation (4.19), has a null space component.

As an example consider the case of one particle per cell with the particles numbered $0, 1, 2, 3$, etc. being midway between nodes, $i-1, i, i+1$ etc. Then, with linear basis functions the mapping to nodal values is given by

$$\begin{aligned} U_{i-1} &= (u_0 + u_1)/2, \\ U_i &= (u_1 + u_2)/2, \\ U_{i+1} &= (u_2 + u_3)/2. \end{aligned} \tag{4.66}$$

The new values at particles 1 and 2 are now given by the mapping

$$u^*_1 = (u_0 + 2u_1 + u_2)/4, \tag{4.67}$$

$$u^*_2 = (u_1 + 2u_2 + u_3)/4. \tag{4.68}$$

As the null space vector associated with the mapping from particles to nodes is

$$\mathbf{v} = [-1, 1]^T. \tag{4.69}$$

The mapped values u^*_1 and u^*_2 only have a zero null space component if they are equal,

$$[u^*_1, u^*_2].[-1, 1]^T = 0 \tag{4.70}$$

which requires that

$$(u_0 + 2u_1 + u_2) - (u_1 + 2u_2 + u_3) = 0 \tag{4.71}$$

or that

$$u_0 + u_1 - u_2 - u_3 = 0 \tag{4.72}$$

holds. A similar result is obtained if gradient values are calculated at the points 1 and 2 using the nodal values.

4.4 Local Removal of the Nullspace Noise

As was shown in [124] using a full singular value decomposition across the whole grid for the removal of null space noise works well for small one-dimensional problems, but is computationally expensive, as the computational complexity of generating the matrix $\mathbf{V}$ with a singular value decomposition is $O((mN)^3)$ [128]. This method calculates a filtered full particle vector across the whole grid as denoted by $\mathbf{U}_p^{filter}$ in terms of the full vector of all the particle values $\mathbf{U}_p^{full}$.

$$\mathbf{U}_p^{filter} = \sum_i (\mathbf{V_i} \cdot \mathbf{U}_p^{full})\mathbf{V}_i, \tag{4.73}$$

where the vectors $\mathbf{V}_i$ are the vectors that are not in the null space as calculated from the full SVD decomposition [124].

It is thus desirable to have a method for removing the null space noise that works locally at each node, X_i with less complexity than the full SVD method. In [124] a local nullspace filter was proposed, the following discussion provides an analysis of the proposed approach. If we consider the interval $[X_{i-1}, X_{i+1}]$, then using (4.56) it is possible to define a particle vector in this interval using the null space associated with node X_i by

$$\mathbf{u}_{ip}^{AF} = (\mathbf{v}_{1,i} \cdot \mathbf{u}_{ip})\mathbf{v}_{1,i}, \tag{4.74}$$

where $\mathbf{v}_{1,i}$ is the vector $\mathbf{v}_1$ associated with the mapping to node X_i. and a particle vector in the interval $[X_i, X_{i+2}]$ using the null space associated with node $i+1$ by

$$\mathbf{u}_{(i+1)p}^{AF} = (\mathbf{v}_{1,i+1} \cdot \mathbf{u}_{(i+1)p})\mathbf{v}_{1,i+1}, \tag{4.75}$$

or by using Equation (4.63) to get

$$\mathbf{u}_{ip}^{AF} = \frac{U_i}{(\mathbf{S}_{ip}^* \cdot \mathbf{S}_{ip}^*)}(\mathbf{S}_{ip}^*)^T. \tag{4.76}$$

$$\mathbf{u}_{(i+1)p}^{AF} = \frac{U_{i+1}}{(\mathbf{S}_{(i+1)p}^* \cdot \mathbf{S}_{(i+1)p}^*)}(\mathbf{S}_{(i+1)p}^*)^T. \tag{4.77}$$

From this and Equation (4.17) we see that the x_pth component obtained by adding Equations (4.76) and (4.77) is

$$\left[\mathbf{u}_{ip}^{AF} + \mathbf{u}_{(i+1)p}^{AF}\right]_p = \frac{U_i}{SN_i}\frac{\phi_i(x_p)}{W_i} + \frac{U_{i+1}}{SN_{i+1}}\frac{\phi_{i+1}(x_p)}{W_{i+1}} \tag{4.78}$$

where $x_p \in I_i$, $SN_i = (\mathbf{S}_{ip}^* \cdot \mathbf{S}_{ip}^*)$, $SN_{i+1} = (\mathbf{S}_{(i+1)p}^* \cdot \mathbf{S}_{(i+1)p}^*)$ and W_i, W_{i+1} are defined by Equation (4.17). Hence, the filtered values correspond to a modified form of linear interpolation. However, as the example in Section 4.1 illlustrates the linear interpolation mapping can still reintroduce null space errors. The reason for the new nullspace error is that parts of the vector $\mathbf{S}_{ip}$ may lie in the null space of $\mathbf{S}_{(i+1)p}$ and vice versa. For this reason a second mapping is done back to the grid nodes and then linear interpolation used again.

The local nullspace method [124] takes a different approach than the SVD method to removing the null space components. The key idea in the local method is to

- first map particle values to the nodes, using Equation (4.19) to compute U_i and U_{i+1}.
- and then interpolate values from nodes to particles as in Equations (4.31, 4.32). When this happens, a null space component is introduced by this calculation.
- The newly computed solution values are again mapped back to the nodes, and
- These new nodal values are interpolated back to the particles.

In the case of mapping velocities to the nodes and then forming velocity gradients at any point x_p in an interval I_i the value of the derivative instead of being

$$\frac{\partial v}{\partial x}(x_p) = \frac{V_{i+1} - V_i}{h} + (1 - 2\alpha_p)\frac{h}{2}V_{xx}(x_p) + \text{h.o.t} \tag{4.79}$$

is now defined for the new method by

$$\begin{aligned}\frac{\partial v}{\partial x}(x_p) &= (1-\alpha_p)S_i^+\left(\frac{(V_{i+1}-V_i)}{h} + S_i^-\frac{(V_i - V_{i-1})}{h}\right) \\ &+ \alpha_p\left(S_{i+1}^+(\frac{(V_{i+2}-V_{i+1})}{h} + S_{i+1}^-\frac{(V_{i+1}-V_i)}{h}\right)\end{aligned} \tag{4.80}$$

where α_p is defined by Equation (4.33) or (4.34) and

$$S_i^+ = \sum_{p \in I_i} S_{ip}^* \tag{4.81}$$

$$S_i^- = \sum_{p \in I_{i-1}} S_{ip}^*. \tag{4.82}$$

The error of this new method is given by a Taylor's series analysis as

$$\begin{aligned} \frac{\partial v}{\partial x}(x_p) &- (1-\alpha_p)\left(S_i^+ \frac{(V_{i+1}-V_i)}{h} + S_i^- \frac{(V_i - V_{i-1})}{h}\right) \\ &+ \alpha_p \left(S_{i+1}^+ \frac{(V_{i+2}-V_{i+1})}{h} + S_{i+1}^- \frac{(V_{i+1}-V_i)}{h}\right) \\ &= i((1-\alpha_p)(S_i^+ - S_i^-) + \alpha_p(S_{i+1}^+ - S_{i+1}^-))\frac{h_i}{2} V_{xx}(x_p). \end{aligned} \tag{4.83}$$

The effect of this approach is thus not to increase the underlying accuracy but to broaden out the stencil used to create the derivatives at the particles. This approach can be used with both solution and gradient values, but is used here with the velocity gradient calculation.

4.4.1 Multidimensional Extensions

As this approach makes use of the standard mappings from only particles to nodes and back again, there is no conceptual problem in extending it to the use of MPM in two and three space dimensions. While there is further work to be done, there is clearly room for combining the approaches suggested here with the improved grid crossing approaches such as [100] and the locking approach of Mast [112]. In particular the latter approach with its careful tensor-based decomposition ideas may have important implications for multi-dimensional extensions of the approach suggested here. What may well be the case, however, is that it is a combination of all these approaches that will be important.

4.5 Alternate Approaches to Global Nullspace Filter

In the previous section a local nullspace filtering method was presented which greatly reduces the computational cost of reducing the nullspace noise as compared to the SVD approach. As noted above, the above filter only approximates the nullspace filtering affects of the full SVD approach. Probabilistic approaches to matrix decompositions [130] present an interesting possibility for reducing the computational cost of the global nullspace filter. In general terms, the randomized SVD [131], as applied to a $m \times n$ matrix $\mathbf{A}$, involves two

steps [132]. The first step is to generate a random projection matrix $\mathbf{P}$ of size $n \times r$, where r is less than or equal to the rank of matrix $\mathbf{A}$, and multiply $\mathbf{A}$ by $\mathbf{P}$,

$$\mathbf{B} = \mathbf{AP}. \tag{4.84}$$

The QR factorization is then applied to the reduced matrix $\mathbf{B}$,

$$\mathbf{B} = \mathbf{QR}. \tag{4.85}$$

Matrix $\mathbf{A}$ is then projected onto $\mathbf{Q}$,

$$\mathbf{C} = \mathbf{Q}^T\mathbf{A}. \tag{4.86}$$

The second step computes the SVD of $\mathbf{A}$ by first computing the SVD of the reduced matrix $\mathbf{C}$,

$$\mathbf{C} = \mathbf{U}_C\mathbf{\Sigma}_C\mathbf{V}_C^T. \tag{4.87}$$

Note that both $\mathbf{A}$ and $\mathbf{C}$ share the same values of $\mathbf{\Sigma}_C$ and $\mathbf{V}_C$ [132], i.e.,

$$\mathbf{\Sigma}_C\mathbf{V}_C^T = \mathbf{\Sigma}_A\mathbf{V}_A^T. \tag{4.88}$$

The matrix $\mathbf{U}$ of the SVD of matrix $\mathbf{A}$ is then computed using matrix $\mathbf{Q}$ as follows:

$$\mathbf{U}_A = \mathbf{QU}_C. \tag{4.89}$$

The use of the randomized SVD as a possible nullspace filtering technique for MPM presents an interesting direction for possible future research. For a more complete discussion of the randomized SVD the reader is referred to [132].

4.6 New Work in Nullspace Filtering Methods

Since the initial publication of the work presented in this chapter [133] and previous work [125, 124], new approaches to addressing the issues associated with the nullspace errors have been developed. Hammerquist and Nairn [134] introduced the XPIC method for enhancing the stability of MPM. In their approach a filter is applied during the particle update phase of the MPM algorithm. For example if $\overline{\mathbf{v}}_i$ represents the vector of all particle

velocities, $\overline{\mathbf{a}}_i$ represents the vector of all nodal accelerations, and $\tilde{\mathbf{S}}_{ip}$ is the matrix that maps nodal values to particles, then the particle velocity update is represented as

$$\overline{\mathbf{v}}_p^{n+1} = \overline{\mathbf{v}}_p^n + \Delta t \tilde{\mathbf{S}}_{ip} \overline{\mathbf{a}}_i^n. \tag{4.90}$$

The XPIC filtering approach uses both the node to particle mapping matrix, $\tilde{\mathbf{S}}_{ip}$, and the particle to node mapping matrix, $\mathbf{S}_{ip}$. The particle velocity update using the XPIC method is defined as

$$\overline{\mathbf{v}}_p^{n+1} = \overline{\mathbf{v}}_p^n - (\mathbf{I} - \tilde{\mathbf{S}}_{ip}\mathbf{S}_{ip})^m \overline{\mathbf{v}}_p^n + \Delta t \tilde{\mathbf{S}}_{ip} \overline{\mathbf{a}}_i^n. \tag{4.91}$$

In the case where $m = 1$, the above equation reduces to

$$\overline{\mathbf{v}}_p^{n+1} = \tilde{\mathbf{S}}_{ip}\mathbf{S}_{ip} \overline{\mathbf{v}}_p^n + \Delta t \tilde{\mathbf{S}}_{ip} \overline{\mathbf{a}}_i^n. \tag{4.92}$$

The approach used for XPIC and the approaches discussed in this chapter and in previous work [125, 124] share some common features. For example the matrix matrix multiplication, $\tilde{\mathbf{S}}_{ip}\mathbf{S}_{ip}$, in the equation above is the same operation that is used for the local nullspace filter approach discussed in Section 4.4. Hammerquist and Nairn also note that as $m \to \infty$ the XPIC approach removes the nullspace component using the orthogonal nullspace vectors which is similar in approach, but different in application, to the SVD method discussed in [125, 124].

Nairn and Hammerquist [135] build upon their XPIC approach which proposes the use of an approximation to a full mass matrix to improve the stability and accuracy of MPM. Tran and Solowski [136] use a nullspace filter coupled with generalized - α integrator to improve the performance of MPM. Their proposed nullspace filter approach uses a QR decomposition to remove the nullspace component in the computed particle velocity gradients.

4.7 Computational Experiments

We compare the original MPM approach to those proposed above. We start by defining the one dimensional bar problem, the linear elastic constitutive model, associated initial conditions, and the analytic solution. Experiments are then run to compare the original MPM approach to those described above.

4.7.1 Model Problem

The constitutive model used for this experiment is defined as follows:

$$\sigma = P = E\frac{\partial u}{\partial X} = E(F-1), \tag{4.93}$$

where E is the Young's modulus. The rate of change of stress is then computed as

$$\dot{\sigma} = E(\dot{F}), \tag{4.94}$$

$$= E(lF), \tag{4.95}$$

where l is the velocity gradient in the spatial description.

The problem considered is a one dimensional bar problem, following similar examples in [89]. The analytic solutions for displacement and velocity defined in the material description are

$$u(X,t) = Asin(2\pi X)sin(c\pi t), \tag{4.96}$$

$$\frac{\partial u}{\partial t} = Ac\pi sin(2\pi X)cos(c\pi t), \tag{4.97}$$

where $c = \sqrt{E/\rho_o}$ and A is the maximum displacement. The constitutive model is defined in Equation 4.93 and the body force is

$$b(X,t) = 3A(c\pi)^2 u(X,t). \tag{4.98}$$

The initial spatial discretization uses two evenly spaced particles per cell with the spatial domain being $[0,1]$. The periodic nature of the analytic solution means that both periodic boundary conditions and zero Dirichlet boundary conditions are both appropriate. The initial conditions for the updated Lagrangian description of the particles are

$$F = 1, \tag{4.99}$$

$$x_p = X_p^0, \tag{4.100}$$

$$V_p = V_p^0. \tag{4.101}$$

For the 1d bar problem the cell width is $h = 10^{-2}$, the material density is $\rho_0 = 1$, Young's modulus is $E = 10^3$, maximum displacement is $A = 3\times10^{-3}$, and the time step is $dt = 10^{-5}$. It should be noted that with the use of the above parameters no particles will cross from one cell to another.

4.7.2 Versions MPM Used in Experiments

Initial experiments were undertaken with standard MPM using linear basis functions at the nodes and delta functions at the particles to provide a baseline calculation. Two modifications were then made to standard MPM in order to apply the new corrected derivatives. First, the mapping of mass and velocity to the nodes was modified to use the mapping function as defined by (4.19). Similarly, the modified derivative weight function defined by (4.27) replaces the standard derivative weight function when computing the internal forces as found in (4.41) of the MPM algorithm. From Fig. 4.2 it can be seen that the use of the corrected derivatives in computing the internal force calculation greatly increased the accuracy of the method. By the 2000th time step there is a noticeable difference in accuracy between those versions of MPM that use the corrected derivatives and the standard version.

To further increase the accuracy the use of the full SVD and approximate nullspace filters were used in combination with the corrected derivatives. The nullspace filters are applied after velocity gradients are computed as done in (4.80). Again from Fig. 4.2 it can be seen that the use of the nullspace filter in both the SVD form and the approximate forms appear to improve the accuracy of MPM when used in conjunction with the corrected derivatives. Here are a few observations from the experimental results. First, the example problem is a one dimensional bar where the material between the two end points is oscillating between compression and tension. Because of this oscillating behavior, the numerical solution and the analytic solution get closer to each other at different points in the cycle, which causes the up and down pattern in Fig. 4.2. The second observation is the sudden change in the error of the full SVD nullspace filter after 14000 time steps. In numerical experiments it was observed that in some cases small instabilities would be introduced that would eventually led to the solution becoming completely unstable. At this point the reason for this behavior is not understood and further work still needs to be done to explore the cause of this type of instability.

In these simulations it is observed that the nullspace filter removes the nullspace noise that is introduced when velocity gradients are computed at particles. The nullspace noise produces oscillations that are visible in the computed velocities of the particles. Fig. 4.3 shows the computed velocity of each particle. On the right hand side of the figure are

velocity plots at three time steps where corrected derivatives MPM was used and on the left are velocity values where the global null space filter was applied. Fig. 4.4 shows the same results but using the local filter. From these plots it can be seen that both the nullspace filter remove the oscillations that can arise due to the nullspace noise.

The dual domain MPM [100] increases the accuracy of MPM by improving the method of computing weight function gradients. The improved weight function gradients, as described in [100], are applied to the internal force calculation and the velocity gradient calculation steps of the MPM algorithm described in this paper. The updated version of MPM using the improved weight function gradient calculations is then used to solve the example problem described in this section. Fig. 4.5 shows a comparison of the dual domain method compared to the corrected derivatives version with the approximated nullspace filter as proposed in this paper. From the plot it can be seen that the two methods are comparable in accuracy.

Further work needs to be done on the methods proposed in this paper. Currently, the use of the different nullspace filters have been applied to problems where particle do not cross cell boundaries. The nullspace filters have not currently shown improvements in accuracy when cell crossings are involved.

4.8 Summary

MPM, like all variants of PIC methods, suffers from a mismatch in the dimensionality between grid nodes and particles. As has been shown when particle values are mapped to grid nodes, particle data are lost. We also demonstrate that when grid values are used to update particle values spurious noise can be introduced into the particle update due to the nullspace of the grid to particle mapping.

A global solution to the removal of the nullspace noise is through the use of an SVD approach, but such an approach can be computationally prohibitive for larger problems. In this chapter an analysis of the local approach proposed in [124] for approximating the nullspace noise removal has been presented and has been shown in preliminary experiments to help reduce numerical noise. Along with the local nullspace filter, the use of corrective derivatives have been shown in preliminary experiments to improve the accuracy of the method. As with the work on SPH [115], there is the potential for using the presented approaches in multidimensional production codes as part of future work.

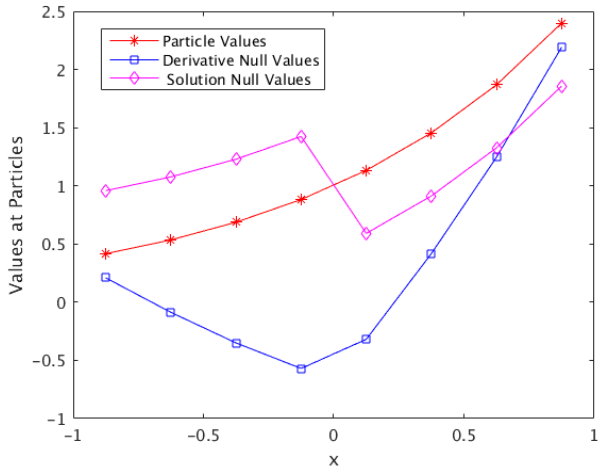

Fig. 4.1. Nullspaces of particle to node solution and derivative mappings.

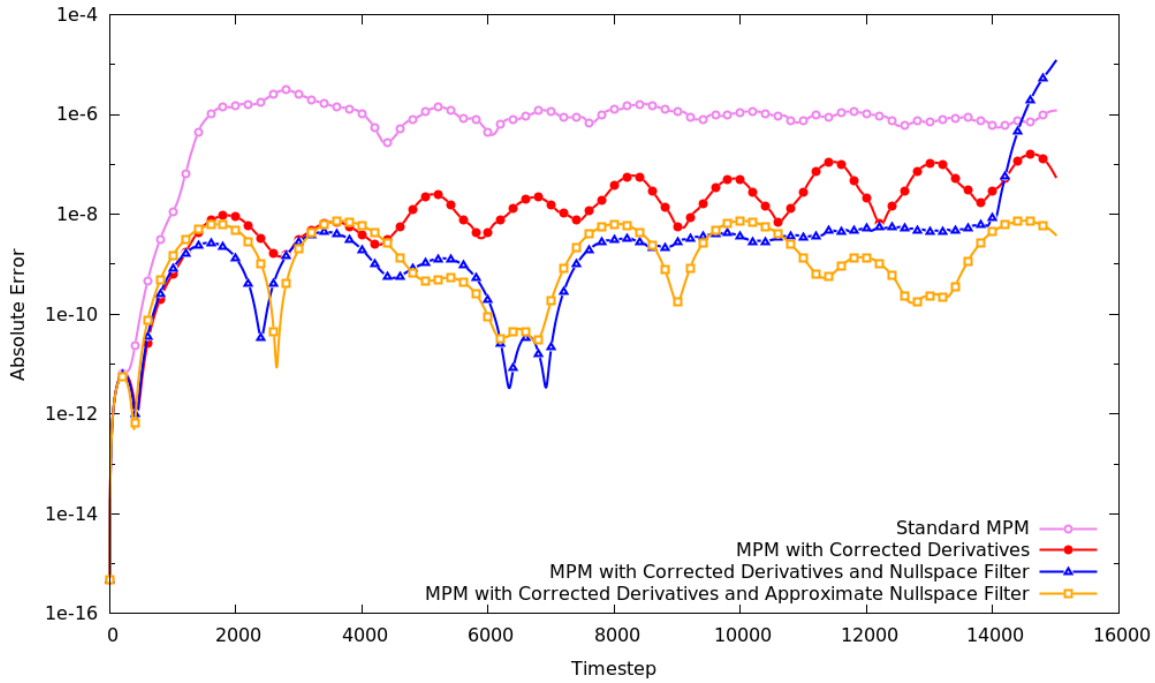

Fig. 4.2. One dimensional bar, error versus time steps for MPM, corrected derivatives MPM, and corrected derivatives MPM with full and approximate nullspace filter.

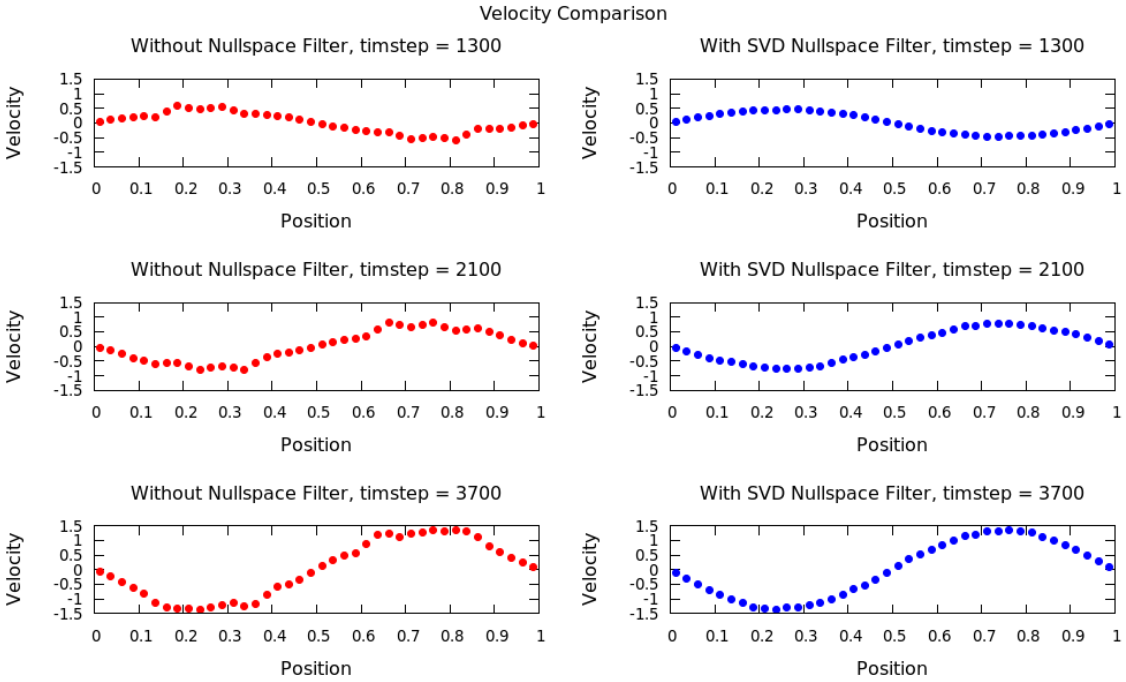

Fig. 4.3. One dimensional bar, particle velocities for corrected derivatives MPM, and corrected derivatives MPM with global nullspace filter.

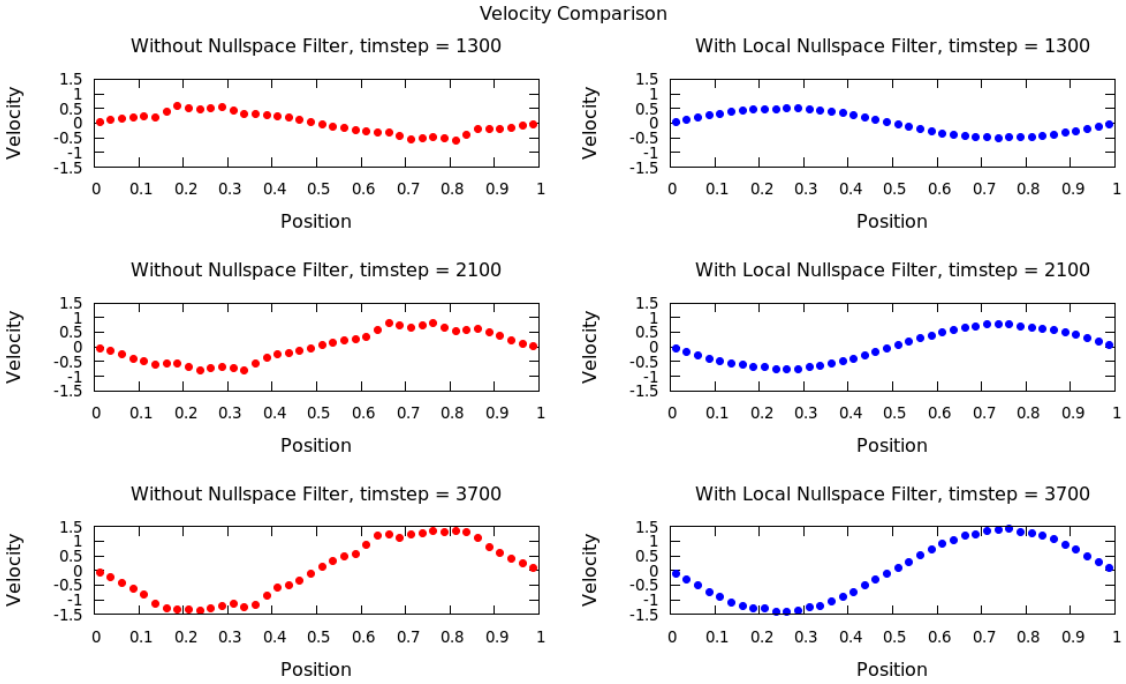

Fig. 4.4. One dimensional bar, particle velocities for corrected derivatives MPM, and corrected derivatives MPM with local nullspace filter.

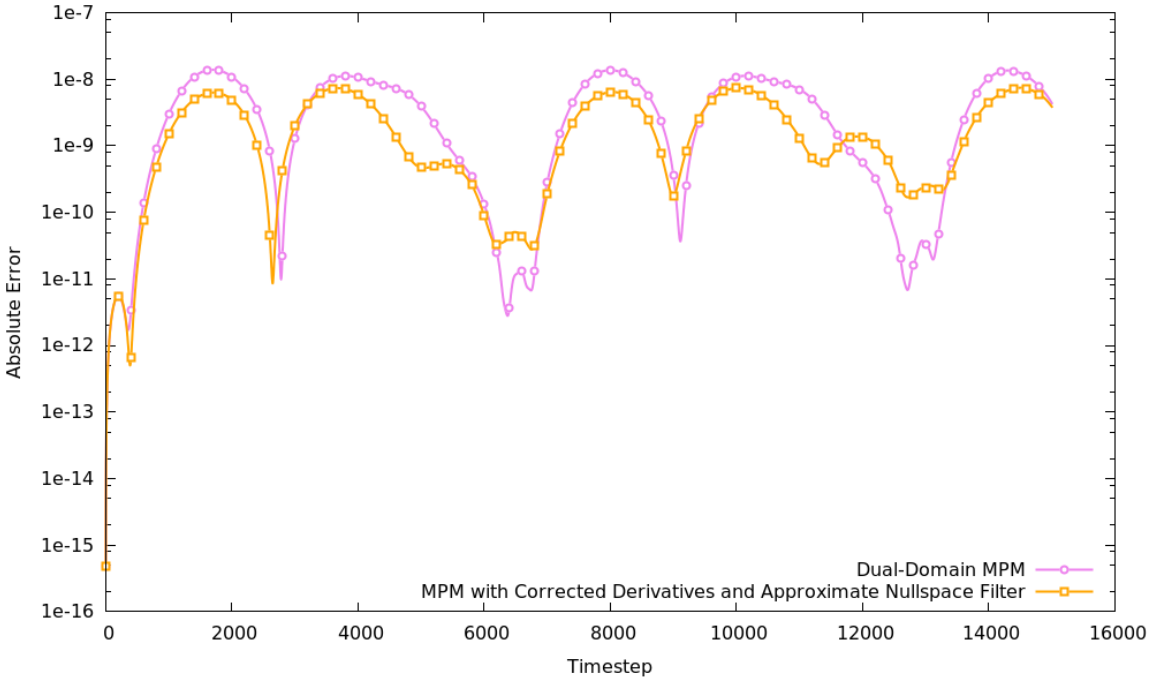

Fig. 4.5. One dimensional bar, error versus time steps for MPM, dual-domain MPM, and corrected derivatives MPM with approximate nullspace filter

CHAPTER 5

TIME INTEGRATION METHODS FOR MULTISCALE PROBLEMS USING MPM

Difficulties arise in modeling multiphysics problems when the coupled set of physical processes occurs at time scales that can be many orders of magnitude different. An example of this is the fully coupled chemical diffusion and mechanical deformation problem described in Chapter 7. In this example lithium ions diffuse through a silicon anode, which causes the anode to swell. The swelling induces mechanical waves to travel through the material. A simple comparison between diffusivity of lithium in amorphous silicon and speed of sound in the same material will show the differences in time scale.

The diffusivity of lithium within amorphous silicon is estimated to be within the range of $10^{-18}\,m^2/s$ to $10^{-14}\,m^2/s$ [137, 138, 139]. The equation for the material sound speed is [140]

$$c = \sqrt{\frac{K + 4/3G}{\rho}}, \tag{5.1}$$

where K is the bulk modulus, G is the shear modulus, and ρ is the mass density. For amorphous silicon, reasonable values for initial density, bulk modulus, and shear modulus are $2.33 \times 10^3\,kg/m^3$, $67\,GPa$, and $31\,GPa$, respectively, which is a sound speed of $6.8 \times 10^3\,m/s$. A simple comparison between diffusivity and sound speed shows the difference in magnitude between the two physical processes.

A comparison of the stability constraints for the explicit time integration of diffusion and mechanical wave propagation problems further demonstrates the point. For a diffusion problem the time step constraint is [141],

$$\Delta t < \frac{\Delta x^2}{2D} \tag{5.2}$$

where Δt is the time step size, Δx is the grid spacing, and D is the diffusivity. For the wave equation the time step size is restricted by the CFL condition [141], which is defined as,

To illustrate, if a second-order finite difference scheme coupled with an explicit Euler time integration method is used to model the diffusion process, then the limit on time step size, based on the stability constraint, is

$$\Delta t \leq \frac{\Delta x}{c}. \tag{5.3}$$

Note that in MPM a constraint of this form is not sufficient to ensure stability as was demonstrated in [122].

Using a uniform grid with spacing $\Delta x = 10^{-8}\, m$, a diffusivity value of $D = 10^{-15}\, m^2/s$, and the sound speed computed above, the stability limited time step sizes for diffusion and deformation are, respectively,

$$\Delta t \leq \frac{\Delta x^2}{2D} = 5.0 \times 10^{-2} s \tag{5.4}$$

$$\Delta t \leq \frac{\Delta x}{c} = 1.47 \times 10^{-12} s. \tag{5.5}$$

From this simple example it would take roughly 10^{10} deformation time steps to reach the length of one diffusion time step. Based on these numbers, it would be impossible to see any appreciable change in the diffusion process within a tractable time frame. An implicit time-integration scheme can be used to ease the stability constraint on the time step size for the deformation process. Using the implicit approach would reduce the number of time steps needed to reach the length of one diffusion based time step, but at the possible cost of a decrease in numerical accuracy. This simple examples demonstrates the care that is needed when modeling a coupled system of this type.

5.1 Introduction

An understanding of the different time integration approaches used in MPM is necessary when solving problems as the one described above. In its original implementation, MPM uses a symplectic Euler time integration scheme [122].

5.1.1 Related Work

Since the introduction of MPM by Sulsky et al. [16], a large body of work has been produced to address the accuracy, stability, performance, and understanding of the method. Research focused on the issues associated with time integration schemes is becoming a larger area of focus for the method. Bardenhagen [109] examined energy conservation issues associated with the MPM and the choice of time integration schemes. Wallstedt and Guilkey [113]

investigated the choices of explicit integration schemes and presented a Störmer-Verlet-like integration scheme for MPM. Berzins [122] performed the stability analysis and derived the stability bounds for the symplectic Euler-A time integration method, which is commonly used in MPM. Berzins also derived the time integration errors and examined the energy conservation properties of the Stormer-Verlet method as applied to MPM [142]. Gast et al. [143] introduced an optimization integrator that allows for the use of larger time step sizes in MPM. Berzins [144] looked at the accuracy and energy conservation properties of a variety of explicit time integration schemes for MPM. Von Neumann stability analysis has also been applied to PIC methods [124] and MPM [121, 120].

Implicit integration methods have also become an area of focus dynamic problems. There are a variety of approaches to using implicit MPM for dynamic problems. Cummins and Brackbill developed an implicit MPM approach for granular flow [145]. Guilkey and Weiss [146] implemented an implicit approach based on particle value updates using the trapezoidal rule. Sulsky and Kaul also explored an implicit approach to MPM [147]. Wang et al. developed an implicit MPM approach for geotechnical problems [148].

5.2 The Ordinary Differential Equations Based on MPM Discretization

To better examine the implementation of an appropriate time integration scheme for MPM, as applied to the coupled multiphysics problem, it is appropriate to write the MPM formulation as a system of ordinary differential equations (ODEs). As is often done in finite element methods (FEM), the momentum balance equation can be written as

$$\mathbf{M}\underline{\mathbf{a}} = \underline{\mathbf{f}} + \underline{\mathbf{b}} \tag{5.6}$$

where $\mathbf{M}$ is the mass matrix, $\underline{\mathbf{a}}$ is the vector of nodal accelerations, $\underline{\mathbf{f}}$ is the vector of internal force values, and $\underline{\mathbf{b}}$ is the vector of nodal body forces. The nodal internal force calculation in MPM, as described in Section 3.5.2 but repeated here for clarity, is

$$\mathbf{f}_i = -\sum_p \boldsymbol{\sigma}_p \cdot \phi_{ip} V_p. \tag{5.7}$$

The constitutive model for stress is a function of the deformation gradient along with model specific parameters such as velocity and concentration, e.g.,

$$\boldsymbol{\sigma}_p = \boldsymbol{\sigma}(\mathbf{F}_p, \mathbf{v}_p, c_p). \tag{5.8}$$

MPM is an updated Lagrangian calculation in that at the beginning of each time step the background grid is reset and the new material and reference configurations are the same. Particle quantities are mapped to the grid and nodal velocities are then updated. Using the updated velocities incremental displacement, $\mathbf{u}^*$, of the node can be computed. The updated the deformation gradient can be treated as a decomposition of a new incremental deformation gradient, $\mathbf{F}^*$, based on the incremental displacement and the deformation gradient at the beginning of the time step, i.e.,

$$\mathbf{F}^{n+1} = \mathbf{F}^*\mathbf{F}^n. \tag{5.9}$$

The new incremental deformation gradient is computed using the incremental displacement as follows:

$$\mathbf{F}^* = (\mathbf{I} + \nabla\mathbf{u}^*). \tag{5.10}$$

Because this is an updated Lagrangian calculation and particles have yet to be moved, gradients with respect to material and reference coordinates are the same. Using the incremental displacement values at the nodes the displacement gradients are calculated at the particles as follows:

$$\nabla\mathbf{u}_p^* = \sum_i \mathbf{u}_i^* \otimes \nabla\phi_{ip}. \tag{5.11}$$

Combining (5.9) and (5.10) the updated deformation gradient at the particle is

$$\mathbf{F}_p^{n+1} = (\mathbf{I} + \nabla\mathbf{u}_p^*)\mathbf{F}_p^n. \tag{5.12}$$

Observe that the updated particle stress is dependent upon the updated deformation gradient as defined by (5.8) and the updated particle deformation gradient is dependent upon the nodal displacement as defined by (5.11) and (5.12). Based on these two observations the nodal internal force, as defined by (5.7), can be written as a function of the nodal displacements plus other model dependent variables, i.e,

$$\mathbf{f}_i = \mathbf{f}_{int}(\mathbf{u}_1, \ldots, \mathbf{u}_n). \tag{5.13}$$

A more common approach in MPM is to update the deformation gradient in terms of rate of deformation.

$$\dot{\mathbf{F}} = \mathbf{l}\mathbf{F} \tag{5.14}$$

where $\mathbf{l}$ is the velocity gradient. The deformation gradient updated using a forward Euler time integration method as follows:

$$\mathbf{F}_p^{n+1} = \mathbf{F}_p^n + \Delta t\, \mathbf{l}_p \mathbf{F}_p^n. \tag{5.15}$$

The particle velocity gradient is calculated in a manner similar to (5.11),

$$\mathbf{l}_p = \sum_i \mathbf{v}_i \otimes \nabla \phi_{ip}. \tag{5.16}$$

As was noted previously, the updated particle stress is dependent upon the updated deformation gradient. Observe that in this approach the updated particle deformation gradient is dependent upon the nodal velocities as defined by (5.15) and (5.16). Based on this observation the nodal internal force is a function of the nodal velocities, the timestep, plus other model dependent variables, i.e.,

$$\mathbf{f}_i = \mathbf{f}_{int}(\mathbf{v}_1, \ldots, \mathbf{v}_n, \Delta t). \tag{5.17}$$

Let $\underline{\mathbf{u}} = [\mathbf{u}_1, \ldots, \mathbf{u}_n]^T$, $\underline{\mathbf{v}} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]^T$, and $\underline{\mathbf{x}} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T$ be the vectors of nodal displacements, velocities, and positions, respectively. The system of ODEs for MPM is then written in a similar manner to that which is done in FEM. In the case where the deformation gradient is updated according nodal displacements, the system is written as

$$\mathbf{M}\underline{\mathbf{a}} = \mathbf{f}(\underline{\mathbf{u}}) + \mathbf{b}(\underline{\mathbf{x}}). \tag{5.18}$$

In the case where deformation gradients are updated according to nodal velocities the system is defined as

$$\mathbf{M}\underline{\mathbf{a}} = \mathbf{f}_{int}(\underline{\mathbf{v}}) + \mathbf{b}(\underline{\mathbf{x}}). \tag{5.19}$$

The relationship between nodal velocity and nodal acceleration is

$$\dot{\mathbf{v}} = \mathbf{a} \tag{5.20}$$

and the relationship between nodal displacement and nodal velocity is

$$\dot{\mathbf{u}} = \mathbf{v}. \tag{5.21}$$

The nodal values for external forces are computed as a function of time and nodal location, i.e,

$$\mathbf{b}_i^n = \mathbf{b}(\mathbf{x}_i, t). \tag{5.22}$$

5.3 Explicit MPM

One of the initial decisions that needs to be made when choosing a integration scheme is where in the time step algorithm is the stress update performed. Bardenhagen [109] proposed two approaches, update stress first (USF) or update stress last (USL). In the USF approach after the initial steps of mapping mass and velocity values to the grid, velocity gradients are then computed at the particles,

$$\nabla \mathbf{v}_p^n = \sum_i \mathbf{v}_i^n \otimes \nabla \phi_{ip}. \tag{5.23}$$

From here updates to particle deformation gradients, volumes, and stresses are computed. The USL approach uses the updated nodal velocity values to compute the velocity gradient at the particles, followed by updating particle deformation gradients, volumes and stresses. The original MPM uses USL.

A variety of explicit time integration schemes have been proposed for MPM. The original MPM time uses a Symplectic Euler scheme [149]. Other explicit time integration approaches that have been adopted to MPM have involved a central difference scheme [113, 150] and the use of the Stormer-Verlet time integration method [149, 144]. In the central difference approach developed by Wallsteadt and Guilkey [113], velocities values are shifted by half a time step prior to the start of the simulation with the algorithmic steps then being similar to the traditional USL MPM approach. Berzins [144] developed an alternate time integration for MPM based on the Stormer-Verlet integration scheme. This approach involves additional mappings from grid to particles in order to account for the changes in particle position the occur midway through the time stepping algorithm. Table 5.1 shows a comparison between the traditional USL, Stormer-Verlet, and central difference approaches. Note that in the table $\boldsymbol{\sigma}()$ denotes the function that computes the Cauchy stress and $\Delta\boldsymbol{\sigma}()$ is the function to compute the objective stress rate which, in both cases, are defined by the chosen constitutive model.

The method for updating particle values from grid values has been investigated. There are two approaches to updating particle values that fall under either the PIC approach or the FLIP approach. In the PIC approach updated node values are mapped directly to the particles,

$$\mathbf{v}_p^{n+1} = \sum_i \phi_{ip} \mathbf{v}_i^{n+1}. \tag{5.24}$$

In the FLIP approach [15] the update to the particle value is mapped to the particle and then the particle value is updated,

$$\mathbf{v}_p^{n+1} = \mathbf{v}_p^n + \Delta t \sum_i \phi_{ip} \mathbf{a}_i^n. \tag{5.25}$$

The drawback of the PIC method is the excess of numerical diffusion and for the FLIP method it is the introduction of numerical noise [98]. Zhu and Bridson [84] use a hybrid of the two methods,

$$\mathbf{v}_p^{n+1} = \beta \left(\mathbf{v}_p^n + \Delta t \sum_i \phi_{ip} \mathbf{a}_i^n \right) + (1-\beta) \sum_i \phi_{ip} \mathbf{v}_i^{n+1}, \tag{5.26}$$

which uses the diffusive nature of the PIC method to control the excess noise of the FLIP method.

5.4 Implicit MPM

A brief discussion of the method developed by Guilkey and Weiss [146] will be presented for context in order to better understand the approach to error analysis of the time integration method. In the implicit method discussed in [146], nodal displacements and velocities are updated using a trapezoidal integration method as defined by

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \frac{\Delta t}{2}(\mathbf{a}_i^n + \mathbf{a}_i^{n+1}) \tag{5.27}$$

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n + \frac{\Delta t}{2}(\mathbf{v}_i^n + \mathbf{v}_i^{n+1}). \tag{5.28}$$

This method of time integration is in the family of Newmark-β methods [151], i.e.,

$$\mathbf{v}^{n+1} = \mathbf{v}^n + \frac{\Delta t}{2}(\mathbf{a}^n + \mathbf{a}^{n+1})$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, \mathbf{v}^n + \frac{1-2\beta}{2}\, \Delta t^2\, \mathbf{a}^n + \beta\, \Delta t^2\, \mathbf{a}^{n+1}$$

with $\beta = 1/4$. With MPM being an updated Lagrangian method, nodal displacements are reset at the beginning of each time step, i.e., $\mathbf{u}_i^n = \mathbf{0}$, (5.28) is reduced to the following:

$$\mathbf{u}_i^{n+1} = \frac{\Delta t}{2}(\mathbf{v}_i^n + \mathbf{v}_i^{n+1}). \tag{5.29}$$

Combining (5.29) and (5.27) produces the equation for the updated acceleration,

$$\mathbf{a}_i^{n+1} = \frac{4}{\Delta t^2}\mathbf{u}_i^{n+1} - \frac{4}{\Delta t}\mathbf{v}_i^n - \mathbf{a}_i^n. \tag{5.30}$$

Noting the observations found in (5.13), the implicit MPM nodal form for the momentum balance equation is written as

$$m_i^n \mathbf{a}_i^{n+1} = \mathbf{f}_i(\mathbf{u}_1^{n+1}, \ldots, \mathbf{u}_m^{n+1}) + \mathbf{b}(\mathbf{x}_i, t + \Delta t). \tag{5.31}$$

Substituting (5.30) into (5.31) produces the following:

$$m_i^n \left(\frac{4}{\Delta t^2} \mathbf{u}_i^{n+1} - \frac{4}{\Delta t} \mathbf{v}_i^n - \mathbf{a}_i^n \right) = \mathbf{f}(\mathbf{u}_1^{n+1}, \ldots, \mathbf{u}_m^{n+1}) + \mathbf{b}(\mathbf{x}_i, t + \Delta t). \tag{5.32}$$

Written as a system of equations for all nodes, the implicit formulation is

$$\mathbf{M}^n \left(\frac{4}{\Delta t^2} \underline{\mathbf{u}}^{n+1} - \frac{4}{\Delta t} \underline{\mathbf{v}}^n - \underline{\mathbf{a}}^n \right) = \mathbf{f}(\underline{\mathbf{u}}^{n+1}) + \mathbf{b}(\underline{\mathbf{x}}, t + \Delta t). \tag{5.33}$$

The matrix $\mathbf{M}^n$ is the lumped mass matrix that is commonly used in MPM formulations.

In order to solve for $\underline{\mathbf{u}}^{n+1}$, the system of equations defined by (5.33) is rewritten as

$$F(\underline{\mathbf{u}}^{n+1}) = \mathbf{M}^n \left(\frac{4}{\Delta t^2} \underline{\mathbf{u}}^{n+1} - \frac{4}{\Delta t} \underline{\mathbf{v}}^n - \underline{\mathbf{a}}^n \right) - \mathbf{f}(\underline{\mathbf{u}^{n+1}}) - \mathbf{b}(\underline{\mathbf{x}}, t + \Delta t) = \mathbf{0}. \tag{5.34}$$

In this form, Newton's method is applied to (5.34) in order to determine the updated nodal displacements.

5.4.1 Steps for Computing One Timestep of Implicit MPM Algorithm

The steps for computing a complete time step of the implicit algorithm are given below. Note that during the computation process for each time step the particles do not move, as a result the values for ϕ_{ip} and $\nabla\phi_{ip}$ are computed only once at the beginning of each time step. The superscript k is used to denote the kth Newton iteration. The superscript n indicates values computed at time t or the beginning of the time step and superscript $n+1$ indicates the time at the end of the time step, i.e., $t + \Delta t$.

1) Reset grid and set nodal displacement to zero.

$$\mathbf{u}_i^n = \mathbf{0} \tag{5.35}$$

2) Map particle values to nodes.

$$m_i^n = \sum_p \phi_{ip} m_p \tag{5.36}$$

$$\mathbf{p}_i^n = \sum_p \phi_{ip} m_p \mathbf{v}_p^n \tag{5.37}$$

$$\mathbf{f}_i^n = \sum_p \boldsymbol{\sigma}_p \nabla \phi_{ip} \mathbf{v}_p \tag{5.38}$$

$$\mathbf{v}_i^n = \frac{1}{m_i^n} \mathbf{p}_i^n \tag{5.39}$$

$$\mathbf{a}_i^n = \frac{1}{m_i^n} (\mathbf{f}_i^n + \mathbf{b}_i^n) \tag{5.40}$$

3) Compute predicted values for the initial displacement and update velocity according to (5.29).

$$\mathbf{u}_i^k = \mathbf{u}_i^n + \alpha \Delta t \mathbf{v}_i^n \tag{5.41}$$

$$\mathbf{v}_i^k = \frac{2}{\Delta t} \mathbf{u}_i^k - \mathbf{v}_i^n \tag{5.42}$$

The variable α is a user defined parameter in the range $(0 \leq \alpha \leq 1)$.

4) Compute the Jacobian of (5.34) with respect to the nodal displacements.

$$\mathbf{J}_{ij}(\underline{\mathbf{u}}^k) = \frac{\partial F_i}{\partial \underline{u}_j}(\underline{\mathbf{u}}^k) \tag{5.43}$$

5) Solver linear system of equations to determine the displacement increment.

$$\mathbf{J}(\underline{\mathbf{u}}^k) \Delta \underline{\mathbf{u}}^k = -F(\underline{\mathbf{u}}^k) \tag{5.44}$$

6) Update the nodal displacements and velocities.

$$\mathbf{u}_i^{k+1} = \mathbf{u}_i^k + \Delta \mathbf{u}_i^k \tag{5.45}$$

$$\mathbf{v}_i^{k+1} = \frac{2}{\Delta t} \mathbf{u}_i^{k+1} - \mathbf{v}_i^n \tag{5.46}$$

7) Update particle values.

$$\nabla \mathbf{u}_p^{k+1} = \sum_i \mathbf{u}_i^{k+1} \otimes \nabla \phi_{ip} \tag{5.47}$$

$$\mathbf{F}_p^{new} = (\mathbf{I} + \nabla \mathbf{u}_p^{k+1}) \tag{5.48}$$

$$\mathbf{F}_p^{k+1} = \mathbf{F}_p^{new} \mathbf{F}_p^n \tag{5.49}$$

$$\mathbf{l}_p^{k+1} = \sum_i \mathbf{v}_i^{k+1} \otimes \nabla \phi_{ip} \tag{5.50}$$

$$V_p^{k+1} = \det(\mathbf{F}_p^{k+1}) V_p^0 \tag{5.51}$$

8) Update particle stress based on constitutive model.

$$\boldsymbol{\sigma}_p^{k+1} = \boldsymbol{\sigma}(\mathbf{F}_p^{k+1}) \tag{5.52}$$

9) Check for convergence. If the convergence criteria is not met return to step 4.

10) Save nodal and particle state.

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^{k+1} \tag{5.53}$$

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^{k+1} \tag{5.54}$$

$$\mathbf{F}_p^{n+1} = \mathbf{F}_p^{k+1} \tag{5.55}$$

11) Compute updated the nodal acceleration according to (5.30).

$$\mathbf{a}_i^{n+1} = \frac{4}{\Delta t^2}\mathbf{u}_i^{n+1} - \frac{4}{\Delta t}\mathbf{v}_i^n - \mathbf{a}_i^n \tag{5.56}$$

12) Update particle velocity, position, and stress.

$$\mathbf{v}_p^{n+1} = \mathbf{v}_p^n + \frac{\Delta t}{2}\sum_i \phi_{ip}(\mathbf{a}_i^{n+1} + \mathbf{a}_i^n) \tag{5.57}$$

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \frac{\Delta t}{2}\sum_i \phi_{ip}(\mathbf{v}_i^{n+1} + \mathbf{v}_i^n) \tag{5.58}$$

$$\sigma_p^{n+1} = \sigma(\mathbf{F}_p^{n+1}). \tag{5.59}$$

5.5 Formulating the Jacobian

As has been noted earlier, there are different approaches to implementing implicit MPM. Along with differences in approach, there are also different means of determining the Jacobian. In the case of [145] and [147] the Jacobian was formed using an approximation to the Jacobian. Because MPM uses both a grid and particles in its calculations the method for computing the Jacobian of (5.34) with respect to the nodal displacement is not necessarily straightforward. Numerical approaches to approximating the Jacobian using finite difference are a common approach that is used in common ODE libraries such as SUNDIALS [152]. In this section a brief discussion on both analytic and numerical approaches to formulating the Jacobian will be given.

5.5.1 Analytically Determined Jacobian

In [146] the Jacobian was determined analytically. To better help understand to process for determining the Jacobian analytically, a simple one dimensional (1D) example will be given. In this example the constitutive model will be defined by the following equation:

$$\boldsymbol{\sigma} = \mathbf{P} = E(\mathbf{F} - \mathbf{I}). \tag{5.60}$$

In the 1D case, the ith row of (5.34) is written as

$$F_i(\underline{\mathbf{u}}^k) = \frac{4m_i}{\Delta t^2}u_i - \frac{4m_i}{\Delta t}v_i - m_i a_i - \mathrm{f}_i(\underline{\mathbf{u}}^k) - m_i \mathrm{g} = 0, \tag{5.61}$$

where $\underline{\mathbf{u}}^k$ is the vector of nodal displacement values

$$\underline{\mathbf{u}}^k = [u_1, u_2, \ldots, u_{n-1}, u_n]^T \tag{5.62}$$

and g is the acceleration due to gravity. The internal force is evaluated as

$$\mathrm{f}_i(\underline{\mathbf{u}}^k) = -\sum_p \boldsymbol{\sigma}_p(\mathbf{F}_p(\underline{\mathbf{u}}^k))V_p(\mathbf{F}_p(\underline{\mathbf{u}}^k))\nabla\phi_{ip}. \tag{5.63}$$

Taking the partial derivative of (5.61) with respect to the jth element of $\underline{\mathbf{u}}^k$ leads to the following value for the ijth element of the Jacobian:

$$\frac{\partial F_i}{\partial u_j} = \frac{4m_i}{\Delta t^2}\delta_{ij} - \frac{\partial \mathrm{f}_i}{\partial u_j} \tag{5.64}$$

where δ_{ij} is the Kronecker delta function. The partial derivative of (5.63) with respect to the jth element of $\underline{\mathbf{u}}^k$ is defined by

$$\frac{\partial \mathrm{f}_i}{\partial u_j} = -\sum_p \left[\frac{\partial \boldsymbol{\sigma}_p}{\partial \mathbf{F}_p}V_p + \boldsymbol{\sigma}_p\frac{\partial V_p}{\partial \mathbf{F}_p}\right]\frac{\partial \mathbf{F}_p}{\partial u_j}\nabla\phi_{ip}. \tag{5.65}$$

The relationship between the current particle volume and the initial particle value is defined by (5.51). Taking the partial derivative of (5.51) with respect to the particle deformation gradient leads to the following:

$$\frac{\partial V_p}{\partial \mathbf{F}_p} = \frac{\partial(\det(\mathbf{F}_p))}{\partial \mathbf{F}_p}V_p^0. \tag{5.66}$$

In the 1D case the above partial derivative reduces to

$$\frac{\partial V_p}{\partial \mathbf{F}_p} = V_p^0 \tag{5.67}$$

and for the multi-dimension case the partial derivative is

$$\frac{\partial V_p}{\partial \mathbf{F}_p} = \det(\mathbf{F}_p)\mathbf{F}_p^{-T}V_p^0. \tag{5.68}$$

Taking the partial derivatives of the particle stress based on the constitutive model defined by (5.60) leads to the following:

$$\frac{\partial \boldsymbol{\sigma}_p}{\partial \mathbf{F}_p} = E. \tag{5.69}$$

Using the above relationships, (5.65) is reduced to

$$\frac{\partial \mathrm{f}_i}{\partial u_j} = -\sum_p \left[E \det(\mathbf{F}_p^k) + \boldsymbol{\sigma}_p \right] V_p^0 \frac{\partial \mathbf{F}_p^k}{\partial u_j} \nabla \phi_{ip}. \tag{5.70}$$

The deformation gradient for the kth Newton iteration is defined by

$$\mathbf{F}_p^k = (\mathbf{I} + \nabla \mathbf{u}_p^k)\mathbf{F}_p^n, \tag{5.71}$$

$$= \left(\mathbf{I} + \sum_l \mathbf{u}_l^k \otimes \nabla \phi_{lp} \right) \mathbf{F}_p^n. \tag{5.72}$$

In the 1D case the partial derivative of (5.72) with respect to the jth element of $\underline{\mathbf{u}}$ is

$$\frac{\partial \mathbf{F}_p^k}{\partial u_j} = \nabla \phi_{jp} \mathbf{F}_p^n. \tag{5.73}$$

Substituting (5.73) into (5.70) leads to the final form of (5.63), given as

$$\frac{\partial \mathrm{f}_i}{\partial u_j} = -\sum_p \left[E \det(\mathbf{F}_p^k) + \boldsymbol{\sigma}_p \right] V_p^0 \nabla \phi_{jp} \nabla \phi_{ip} \mathbf{F}_p^n. \tag{5.74}$$

Applying (5.74) to (5.61) leads to the final form of the Jacobian as defined by

$$\frac{\partial F_i}{\partial u_j} = \frac{4 m_i}{\Delta t^2} \delta_{ij} + \sum_p \left[E \det(\mathbf{F}_p^k) + \boldsymbol{\sigma}_p \right] V_p^0 \nabla \phi_{jp} \nabla \phi_{ip} \mathbf{F}_p^n. \tag{5.75}$$

5.5.2 Approximating the Jacobian

As can be seen from the example above, for each constitutive model there is a new analytic solution to determining the Jacobian. The analytic approach to determining the Jacobian for each constitutive model can be a difficult and error prone task. An alternate approach to this method is to compute an approximation to the Jacobian using finite differences. This is a common approach to Jacobian formation and a complete discussion on the topic can be found in [153] and [154].

The approximation of each entry of the Jacobian is defined by

$$\mathbf{J}_{ij}(\underline{\mathbf{u}}^k) = \frac{F_i(\underline{\mathbf{u}}^k + \epsilon_j \underline{\mathbf{e}}_j) - F_i(\underline{\mathbf{u}}^k)}{\epsilon_j} \tag{5.76}$$

where F is the function 5.34, ϵ_j is perturbation parameter, and $\underline{\mathbf{e}}_j$ is the column vector defined by

$$\mathbf{e}_j = [0, 0, \ldots, 1, \ldots, 0, 0]^T \tag{5.77}$$

with the 1 being located at the jth location. There exist multiple approaches to to determining the size of ϵ_j. A survey of the different approaches to determining the size of ϵ_j can

be found in [154]. For the purposes of this paper, ϵ_j is computed using the method defined by [155], which is used in the SUNDIALS package [152].

The evaluation of (5.34) given a set of nodal displacements $\underline{\mathbf{u}}^k$ is a relatively strait forward calculation with the exception of the internal force calculation defined by $\mathbf{f}(\underline{\mathbf{u}}^k)$. Because of the hybrid nature of MPM, the internal force calculation involves moving information from grid to particle and then back. The steps to do this are as follows:

1) update grid velocities according to (5.46);
2) update deformation and velocity gradients at the particles using (5.47), (5.48), (5.49), and (5.50);
3) use updated deformation and velocity gradients to compute the particle stresses as defined by the given constitutive model;
4) compute the internal forces at the nodes using (5.7).

5.6 Local Error Analysis for Time Integration Methods

To understand the error analysis frame work used to examine the time integration schemes used in MPM, a brief discussion of error analysis as applied to ordinary differential equations will be presented. Given the following initial value problem,

$$y' = f(t, y(t)), \quad t > t^0 \tag{5.78}$$

$$y(t^0) = \eta. \tag{5.79}$$

The numerical approximation to the function $y(t)$ using an multi-stage explicit method is defined by

$$y^{n+1} = y^n + \Delta t \psi(t^n, y^n, \Delta t), \tag{5.80}$$

where $y^n \approx y(t^n)$ for time steps $t^n > t^0$ and $y^0 = \eta$ at time step $t^n = t^0$. The true or global error of the numerical approximation at a given time t^n is defined by [156, 157],

$$ge^n = y^n - y(t^n). \tag{5.81}$$

Based on Equation 5.78 the local solution at t^n is $g(t)$ where

$$g' = f(t, g(t)), \tag{5.82}$$

$$g(t^n) = y^n. \tag{5.83}$$

The local error at t^n is the accumulated error over one time step based on the local solution [156, 157],

$$le^n = y^{n+1} - g(t^n + \Delta t). \tag{5.84}$$

The relationship between the local error and the global error, based on Equation 5.84 and Equation 5.81, is defined as

$$ge^{n+1} = [g(t + \Delta t) - y(t + \Delta t)] + le^n. \tag{5.85}$$

There exist multiple approaches to approximate global errors, but generally in numerical integrators it is the local error that is controlled. In numerical codes, such as SUNDIALS [152], a local error is computed and then compared to a given tolerance, i.e.,

$$\|le^n\| \leq tol. \tag{5.86}$$

If the local error exceeds the tolerance, the time step size is reduced in order bring the local error below the tolerance level.

5.7 Local Error Analysis of Implicit MPM

A variety of local error estimators for Newmark methods have been proposed [158, 159, 160]. These methods seek to find a local error estimator for dynamics problems of the form,

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}(t)\dot{\mathbf{u}}(t) + \mathbf{K}(t)\mathbf{u}(t) = \mathbf{b}(t), \tag{5.87}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{C}(t)$ is the dampening matrix, and $\mathbf{K}(t)$ is the stiffness matrix. A detailed survey of some of the error estimators for dynamics problems of this type can be found in [161]. While the published work on local error estimators can provide fruitful insights, care needs to be taken when examining time integration methods as applied to MPM. The method by which MPM moves information from particles to grid and then back provides unique challenges when examining local time integration errors. Local error estimation as applied to MPM is relatively new. Berzins [144] used local error estimates in his examination of the Stormer-Verlet time integration scheme as applied to MPM.

5.7.1 Local Error of Nodal Displacement and Velocity

Let $\mathbf{u}_i(t^n)$, $\mathbf{v}_i(t^n)$, and $\mathbf{a}_i(t^n)$ be the local solutions for displacement, velocity, and acceleration at node i, respectively, where

$$\mathbf{u}_i^n = \mathbf{u}_i(t^n), \quad \mathbf{v}_i^n = \mathbf{v}_i(t^n), \quad \text{and} \quad \mathbf{a}_i^n = \mathbf{a}_i(t^n). \tag{5.88}$$

The local errors for the nodal displacements and velocities at $t^n + \Delta t$ are, respectively,

$$le\_\mathbf{u}_i^{n+1} = \mathbf{u}_i^{n+1} - \mathbf{u}_i(t^n + \Delta t) \quad \text{and} \quad le\_\mathbf{v}_i^{n+1} = \mathbf{v}_i^{n+1} - \mathbf{v}_i(t^n + \Delta t). \tag{5.89}$$

The time integration of the nodal displacement is

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n + \frac{\Delta t}{2}(\mathbf{v}_i^n + \mathbf{v}_i^{n+1}), \tag{5.90}$$

and for the velocity update,

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \frac{\Delta t}{2}(\mathbf{a}_i^n + \mathbf{a}_i^{n+1}). \tag{5.91}$$

The Taylor series expansions of nodal displacement and velocity for $t + \Delta t$ about t are

$$\mathbf{u}_i(t + \Delta t) = \mathbf{u}_i(t) + \Delta t \mathbf{v}_i(t) + \frac{\Delta t^2}{2}\mathbf{a}_i(t) + \frac{\Delta t^3}{6}\dot{\mathbf{a}}(\xi_u), \tag{5.92}$$

$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \Delta t \mathbf{a}_i(t) + \frac{\Delta t^2}{2}\dot{\mathbf{a}}_i(t) + \frac{\Delta t^3}{6}\ddot{\mathbf{a}}_i(\xi_v). \tag{5.93}$$

Using (5.89), (5.90), and (5.92) and noting the relationship defined in (5.88), the local error for displacement can be rewritten as

$$le\_\mathbf{u}_i^{n+1} = \frac{\Delta t}{2}(\mathbf{v}_i^{n+1} - \mathbf{v}_i^n) - \frac{\Delta t^2}{2}\mathbf{a}_i^n - \frac{\Delta t^3}{6}\dot{\mathbf{a}}_i(\xi_u). \tag{5.94}$$

Substituting (5.91) in to (5.94) results in the following:

$$le\_\mathbf{u}_i^{n+1} = \frac{\Delta t^2}{4}(\mathbf{a}_i^{n+1} - \mathbf{a}_i^n) - \frac{\Delta t^3}{6}\dot{\mathbf{a}}_i(\xi_u). \tag{5.95}$$

The final term in the above equation can be approximated as

$$\dot{\mathbf{a}}_i(\xi_u) \approx \frac{\mathbf{a}_i^{n+1} - \mathbf{a}_i^n}{\Delta t}. \tag{5.96}$$

Using the approximation above the local error term for the nodal displacements can then be written entirely in terms of the nodal accelerations,

$$le\_\mathbf{u}_i^{n+1} \approx \frac{\Delta t^2}{12}(\mathbf{a}_i^{n+1} - \mathbf{a}_i^n). \tag{5.97}$$

Using (5.89), (5.91), and (5.93) and following a similar approach as above the local error for the velocity update is

$$le\_\mathbf{v}_i^{n+1} = \frac{\Delta t^2}{2}\left[\frac{(\mathbf{a}_i^{n+1} - \mathbf{a}_i^n)}{\Delta t} - \dot{\mathbf{a}}_i\right] - \frac{\Delta t^3}{6}\ddot{\mathbf{a}}_i(\xi_v). \tag{5.98}$$

The final term of the above equation can be approximated using

$$\ddot{\mathbf{a}}_i(\xi_v) \approx \frac{\mathbf{a}_i^{n+1} - 2\mathbf{a}_i^n + \mathbf{a}_i^{n-1}}{\Delta t^2}. \tag{5.99}$$

By using the approximations for $\dot{\mathbf{a}}_i$ and $\ddot{\mathbf{a}}_i$ the local error for the nodal velocity can be reduced to

$$le\_\mathbf{v}_i^{n+1} \approx \frac{\Delta t^3}{6}\left(\frac{\mathbf{a}_i^{n+1} - 2\mathbf{a}_i^n + \mathbf{a}_i^{n-1}}{\Delta t^2}\right). \tag{5.100}$$

At this point we have local errors for both the nodal displacement and velocity.

5.8 Local Error Analysis of Particle Displacements and Velocities

Let $\mathbf{u}_p(t)$ and $\mathbf{v}_p(t)$ be the local solutions for displacement and velocity, respectively, where

$$\mathbf{u}_p^n = \mathbf{u}_p(t) \quad \text{and} \quad \mathbf{v}_p^n = \dot{\mathbf{u}}_p(t). \tag{5.101}$$

The local errors for displacement and velocity at $t + \Delta t$ are, respectively,

$$le\_\mathbf{u}_p^{n+1} = \mathbf{u}_p^{n+1} - \mathbf{u}_p(t + \Delta t) \quad \text{and} \quad le\_\mathbf{v}_p^{n+1} = \mathbf{v}_p^{n+1} - \dot{\mathbf{u}}_p(t + \Delta t). \tag{5.102}$$

The particle update for displacement is

$$\mathbf{u}_p^{n+1} = \mathbf{u}_p^n + \frac{\Delta t}{2}\sum_i \phi_{ip}^n(\mathbf{v}_i^n + \mathbf{v}_i^{n+1}), \tag{5.103}$$

and for velocity,

$$\mathbf{v}_p^{n+1} = \mathbf{v}_p^n + \frac{\Delta t}{2}\sum_i \phi_{ip}^n(\mathbf{a}_i^n + \mathbf{a}_i^{n+1}). \tag{5.104}$$

Noting the relationships defined in (5.101), the Taylor series expansions of nodal displacement and velocity for $t + \Delta t$ about t are

$$\mathbf{u}_p(t + \Delta t) = \mathbf{u}_p^n + \Delta t\mathbf{v}_p^n + \frac{\Delta t^2}{2}\mathbf{a}_p^n + \frac{\Delta t^3}{6}\dot{\mathbf{a}}(\xi_u), \tag{5.105}$$

$$\mathbf{v}_p(t + \Delta t) = \mathbf{v}_p^n + \Delta t\mathbf{a}_p^n + \frac{\Delta t^2}{2}\dot{\mathbf{a}}_p^n + \frac{\Delta t^3}{6}\ddot{\mathbf{a}}_p(\xi_v). \tag{5.106}$$

Combining (5.102), (5.103) and (5.105) the local error for displacement can be written as

$$le\_\mathbf{u}_p^{n+1} = \frac{\Delta t}{2}\left(\sum_i \phi_{ip}^n(\mathbf{v}_i^n + \mathbf{v}_i^{n+1})\right) - \Delta t\mathbf{v}_p^n - \frac{\Delta t^2}{2}\mathbf{a}_p^n - \frac{\Delta t^3}{6}\dot{\mathbf{a}}_p(\xi_u). \tag{5.107}$$

Noting the following relationships:

$$\mathbf{v}_p^n = \sum_i \phi_{ip}^n \mathbf{v}_i^n, \tag{5.108}$$

$$\mathbf{a}_p^n = \sum_i \phi_{ip}^n \mathbf{a}_i^n, \tag{5.109}$$

$$\dot{\mathbf{a}}_p^n = \sum_i \phi_{ip}^n \dot{\mathbf{a}}_i^n, \tag{5.110}$$

$$\ddot{\mathbf{a}}_p^n = \sum_i \phi_{ip}^n \ddot{\mathbf{a}}_i^n, \tag{5.111}$$

then (5.107) can be written as

$$le\_\mathbf{u}_p^{n+1} = \frac{\Delta t^2}{2}\left(\sum_i \phi_{ip}^n \frac{(\mathbf{v}_i^{n+1} - \mathbf{v}_i^n)}{\Delta t} - \mathbf{a}_i^n\right) - \frac{\Delta t^3}{6}\sum_i \phi_{ip}^n \dot{\mathbf{a}}_i(\xi_u). \tag{5.112}$$

Substituting an approximation of $\dot{\mathbf{a}}_i$ into (5.112) results in the following approximation of the local error for the updated particle displacement:

$$le\_\mathbf{u}_p^{n+1} \approx \frac{\Delta t^2}{2}\left(\sum_i \phi_{ip}^n \frac{(\mathbf{v}_i^{n+1} - \mathbf{v}_i^n)}{\Delta t} - \mathbf{a}_i^n\right) - \frac{\Delta t^3}{6}\sum_i \phi_{ip}^n \frac{(\mathbf{a}_i^{n+1} - \mathbf{a}_i^n)}{\Delta t}. \tag{5.113}$$

Determining the local error for the particle velocity update follows a similar approach to that of the particle displacement. Combining (5.102), (5.104), and (5.106) leads to the following equation for the local error,

$$le\_\mathbf{v}_p^{n+1} = \frac{\Delta t}{2}\sum_i \phi_{ip}^n(\mathbf{a}_i^{n+1} + \mathbf{a}_i^n) - \Delta t \mathbf{a}_p^n - \frac{\Delta t^2}{2}\dot{\mathbf{a}}_p^n - \frac{\Delta t^3}{6}\ddot{\mathbf{a}}_p(\xi_u). \tag{5.114}$$

Updating the above equation using (5.109), (5.110), and (5.111) leads to the following:

$$le\_\mathbf{v}_p^{n+1} = \frac{\Delta t^2}{2}\left(\sum_i \phi_{ip}^n \frac{(\mathbf{a}_i^{n+1} - \mathbf{a}_i^n)}{\Delta t} - \dot{\mathbf{a}}_i^n\right) - \frac{\Delta t^3}{6}\sum_i \phi_{ip}^n \ddot{\mathbf{a}}_p(\xi_u). \tag{5.115}$$

Substituting an approximation of $\dot{\mathbf{a}}_i$ and $\ddot{\mathbf{a}}$ into (5.115) results in the following approximation of the local error for the updated particle velocity:

$$le\_\mathbf{v}_p^{n+1} \approx \frac{\Delta t^3}{6}\sum_i \phi_{ip} \frac{(\mathbf{a}_i^{n+1} - 2\mathbf{a}_i^n + \mathbf{a}_i^{n-1})}{\Delta t^2}. \tag{5.116}$$

5.9 Local Error Analysis of Deformation Gradient

Let $\mathbf{F}_p(t)$ be the local solution where

$$\mathbf{F}_p^n = \mathbf{F}_p(t), \quad \mathbf{l}_p^n \mathbf{F}_p^n = \dot{\mathbf{F}}_p(t). \tag{5.117}$$

The local error for the deformation gradient at $t + \Delta t$ is

$$le\_\mathbf{F}_p^{n+1} = \mathbf{F}_p^{n+1} - \mathbf{F}_p(t + \Delta t). \tag{5.118}$$

The deformation update is computed in the following manner:

$$\mathbf{F}_p^{n+1} = \mathbf{F}_p^* \mathbf{F}_p^n, \tag{5.119}$$

where $\mathbf{F}_p^*$ is defined in terms of the incremental displacement gradient,

$$\mathbf{F}_p^* = \mathbf{I} + \sum_i \mathbf{u}_i^{n+1} \otimes \nabla \phi_{ip}^n. \tag{5.120}$$

Noting that

$$\mathbf{u}_i^{n+1} = \frac{\Delta t}{2}(\mathbf{v}_i^{n+1} + \mathbf{v}_i^n), \tag{5.121}$$

as defined by (5.29) in the algorithm, (5.120) can updated to,

$$\mathbf{F}_p^* = \mathbf{I} + \frac{\Delta t}{2}\left(\sum_i \mathbf{v}_i^{n+1} \otimes \nabla \phi_{ip}^n + \sum_i \mathbf{v}_i^n + \otimes \nabla \phi_{ip}^n\right). \tag{5.122}$$

Then noting that the approximate velocity gradient may be written as

$$\tilde{\mathbf{l}}_p^n = \sum_i \mathbf{v}_i^n \otimes \nabla \phi_{ip}^n,$$
$$\tilde{\mathbf{l}}_p^{n+1} = \sum_i \mathbf{v}_i^{n+1} \otimes \nabla \phi_{ip}^n$$

so that (5.122) can be rewritten as

$$\mathbf{F}_p^* = \mathbf{I} + \frac{\Delta t}{2}(\tilde{\mathbf{l}}_p^{n+1} + \tilde{\mathbf{l}}_p^n). \tag{5.123}$$

Substituting the above into (5.119) the deformation gradient is then defined as

$$\mathbf{F}_p^{n+1} = \mathbf{F}_p^n + \frac{\Delta t}{2}(\tilde{\mathbf{l}}_p^{n+1} + \tilde{\mathbf{l}}_p^n)\mathbf{F}_p^n. \tag{5.124}$$

The Taylor series expansion of the deformation gradient local error is defined as follows:

$$\begin{aligned}\mathbf{F}_p(t + \Delta t) &= \mathbf{F}_p(t) + \dot{\mathbf{F}}_p(t)\Delta t + \ddot{\mathbf{F}}_p(\xi_F)\frac{\Delta t^2}{2}, \\ &= \mathbf{F}_p^n + \tilde{\mathbf{l}}_p^n \mathbf{F}_p^n \Delta t + \ddot{\mathbf{F}}_p(\xi_F)\frac{\Delta t^2}{2}.\end{aligned}$$

Combining (5.118) and (5.124) with the above Taylor series expansion gives the local error for the deformation gradient update,

$$le\_\mathbf{F}_p^{n+1} = \frac{\Delta t}{2}\left(\tilde{\mathbf{l}}_p^{n+1} - \tilde{\mathbf{l}}_p^n\right)\mathbf{F}_p^n - \ddot{\mathbf{F}}_p(\xi_F)\frac{\Delta t^2}{2}. \tag{5.125}$$

By the product rule,

$$\ddot{\mathbf{F}}_p^n = \dot{\mathbf{l}}_p^n \mathbf{F}_p^n + \tilde{\mathbf{l}}_p^n \dot{\mathbf{F}}_p, \tag{5.126}$$

and thus the local error can be written as

$$le\_\mathbf{F}_p^{n+1} = \frac{\Delta t^2}{2}\left(\frac{\tilde{\mathbf{l}}_p^{n+1} - \tilde{\mathbf{l}}_p^n}{\Delta t} - \dot{\mathbf{l}}_p^n\right)\mathbf{F}_p^n - \frac{\Delta t^2}{2}\tilde{\mathbf{l}}_p^n\dot{\mathbf{F}}_p. \tag{5.127}$$

Noting that

$$\frac{\tilde{\mathbf{l}}_p^{n+1} - \tilde{\mathbf{l}}_p^n}{\Delta t} = \dot{\mathbf{l}}_p^n + O(\Delta t), \tag{5.128}$$

then (5.127) reduces to,

$$le\_\mathbf{F}_p^{n+1} = -\frac{\Delta t^2}{2}\tilde{\mathbf{l}}_p^n\dot{\mathbf{F}}_p^n + O(\Delta t^3). \tag{5.129}$$

5.10 Summary

The time scales at which diffusion and deformation occurs can be very different. In the opening section of this chapter, a case was presented that demonstrated the large difference in time step size, based on stability constraints, when solving the diffusion and deformation equations using an explicit time integration approach. In order to resolve the stability issues, either an implicit or semi-implicit approach needs to be used.

The use of an implicit method for solving the deformation portion of the coupled deformation and diffusion equations will allow for time steps sizes that are on the same order as those used for the diffusion equation. The drawback to using larger time steps is that the error associated with the time stepping scheme will also increase. If care is not taken in the selection of the time step size, then the resultant error may produce results that are no longer valid for the modeled problem. It is for this reason that an analysis of time integration approach needs to be done in order to understand the implications of using a larger time step size.

In this chapter a local error analysis was done on the implicit MPM scheme introduced by Guilkey and Weiss [146]. For each of the nodal and particle values, a local error approximation has been derived. The approximations found in Equations (5.97), (5.100), (5.113), (5.116), and (5.129) can be used to compute an error estimate that can be used to determine if the time step size is within the allowed tolerances needed for the model. This approach thus provides an alternative to the stability-only approach frequently adopted in MPM time integration [39, 162, 163].

Table 5.1. Explicit time integration scheme comparison.

Traditional	**Stormer-Verlet**	**Centered-Difference**
$m_i = \sum_p \phi_{ip}^n m_p$	$m_i = \sum_p \phi_{ip}^n m_p$	$m_i = \sum_p \phi_{ip}^n m_p$
$\mathbf{p}_i^n = \sum_p \phi_{ip}^n \mathbf{v}_p^n m_p$	$\mathbf{p}_i^n = \sum_p \phi_{ip}^n \mathbf{v}_p^n m_p$	$\mathbf{p}_i^{n-\frac{1}{2}} = \sum_p \phi_{ip}^n \mathbf{v}_p^{n-\frac{1}{2}} m_p$
$\mathbf{v}_i^n = \frac{1}{m_i}\mathbf{p}_i^n$	$\mathbf{v}_i^n = \frac{1}{m_i}\mathbf{p}_i^n$	$\mathbf{v}_i^{n-\frac{1}{2}} = \frac{1}{m_i}\mathbf{p}_i^{n-\frac{1}{2}} m_p$
		$\boldsymbol{\sigma}_p = \boldsymbol{\sigma}(\mathbf{F}_p^n)$
		$V_p^n = det(F_p^n)V_p^0$
$\mathbf{f}_i^n = -\sum_p \nabla\phi_{ip}^n \cdot \boldsymbol{\sigma}_p^n V_p^n$	$\mathbf{f}_i^n = -\sum_p \nabla\phi_{ip}^n \cdot \boldsymbol{\sigma}_p^n V_p^n$	$\mathbf{f}_i^n = -\sum_p \nabla\phi_{ip}^n \cdot \boldsymbol{\sigma}_p^n V_p^n$
$\mathbf{a}_i^n = \frac{1}{m_i}\mathbf{f}_i^n$	$\mathbf{a}_i^n = \frac{1}{m_i}\mathbf{f}_i^n$	$\mathbf{a}_i^n = \frac{1}{m_i}\mathbf{f}_i^n$
$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \Delta t \mathbf{a}_i^n$	$\mathbf{v}_i^{n+\frac{1}{2}} = \mathbf{v}_i^n + \frac{\Delta t}{2}\mathbf{a}_i^n$	$\mathbf{v}_i^{n+\frac{1}{2}} = \mathbf{v}_i^{n-\frac{1}{2}} + \Delta t \mathbf{a}_i^n$
$\mathbf{l}_p^{n+1} = \sum_p \mathbf{v}_i^{n+1} \otimes \nabla\phi_{ip}^n$	$\mathbf{l}_p^n = \sum_p \mathbf{v}_i^{n+\frac{1}{2}} \otimes \nabla\phi_{ip}^n$	$\mathbf{l}_p^{n+\frac{1}{2}} = \sum_p \mathbf{v}_i^{n+\frac{1}{2}} \otimes \nabla\phi_{ip}^n$
$\mathbf{v}_p^{n+1} = \sum_i \phi_{ip}^n \mathbf{v}_i^{n+1}$	$\mathbf{v}_p^{n+\frac{1}{2}} = \sum_i \phi_{ip}^n \mathbf{v}_i^{n+\frac{1}{2}}$	$\mathbf{v}_p^{n+\frac{1}{2}} = \sum_i \phi_{ip}^n \mathbf{v}_i^{n+\frac{1}{2}}$
$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \mathbf{v}_p^{n+1}$	$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \mathbf{v}_p^{n+\frac{1}{2}}$	$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \mathbf{v}_p^{n+\frac{1}{2}}$
	$\mathbf{l}_p^{n+1} = \sum_p \mathbf{v}_i^{n+\frac{1}{2}} \otimes \nabla\phi_{ip}^{n+1}$	
$\dot{\mathbf{F}}_p^* = \mathbf{l}_p^{n+1}\mathbf{F}_p^n$	$\dot{\mathbf{F}}_p^* = \frac{1}{2}(\mathbf{l}_p^n\mathbf{F}_p^n + \mathbf{l}_p^{n+1}\mathbf{F}_p^{n+1})$	$\dot{\mathbf{F}}_p^* = \mathbf{l}_p^{n+\frac{1}{2}}\mathbf{F}_p^n$
$\mathbf{F}_p^{n+1} = \mathbf{F}_p^n + \Delta t \dot{\mathbf{F}}_p^*$	$\mathbf{F}_p^{n+1} = \mathbf{F}_p^n + \Delta t \dot{\mathbf{F}}_p^*$	$\mathbf{F}_p^{n+1} = \mathbf{F}_p^n + \Delta t \dot{\mathbf{F}}_p^*$
$\dot{\boldsymbol{\sigma}}_p^* = \Delta\boldsymbol{\sigma}(\mathbf{l}^{n+1})$	$\dot{\boldsymbol{\sigma}}_p^* = \frac{1}{2}(\Delta\boldsymbol{\sigma}(\mathbf{l}^n) + \Delta\boldsymbol{\sigma}(\mathbf{l}^{n+1}))$	
$\boldsymbol{\sigma}_p^{n+1} = \boldsymbol{\sigma}_p^n + \Delta t \dot{\boldsymbol{\sigma}}_p^*$	$\boldsymbol{\sigma}_p^{n+1} = \boldsymbol{\sigma}_p^n + \Delta t \dot{\boldsymbol{\sigma}}_p^*$	
	$\mathbf{f}_i^{n+1} = -\sum_p \nabla\phi_{ip}^{n+1} \cdot \boldsymbol{\sigma}_p^{n+1} V_p^{n+1}$	
	$\mathbf{a}_i^{n+1} = \frac{1}{m_i}\mathbf{f}_i^{n+1}$	
	$\mathbf{a}_p^* = \frac{1}{2}(\sum_i \phi_{ip}^n \mathbf{a}_i^n + \sum_i \phi_{ip}^{n+1} \mathbf{a}_i^{n+1})$	
	$\mathbf{v}_p^{n+1} = \mathbf{v}_p^n + \Delta t \mathbf{a}_p^*$	

CHAPTER 6

DIFFUSION MODELING IN MPM

The coupled processes of deformation and diffusion can be found in a variety of forms such as thermal-induced stress, concentration-induced stress, and material phase change. Numerical methods such as FEM, SPH, and discontinuous-Galerkin have all been used successfully to solve these types of coupled problems. In some cases the introduction of a new chemical species in a host material or the transfer of heat that induce a phase change in the material can lead to large deformations of the host material. Examples of this type of physical behavior can be seen in melting plastic or a dry sponge dropped in a bucket of water.

MPM is well suited for modeling large deformations and has already been applied to diffusion problems [90, 164, 165, 166, 167, 168, 169]. The heat equation in the spatial configuration is modeled by [170],

$$\rho C \frac{DT}{Dt} + \nabla \cdot \mathbf{q} = 0, \tag{6.1}$$

where C is the heat capacity and $\mathbf{q}$ is the heat flux. Chemical diffusion in the spatial configuration is modeled by [41]

$$\frac{Dc}{Dt} + \nabla \cdot \mathbf{j} = 0, \tag{6.2}$$

where c is chemical species concentration and $\mathbf{j}$ is the flux.

There are a couple of different approach to modeling diffusion with MPM. One common approach [166, 171, 164] follows a similar pattern by which MPM is used to solve the momentum balance equation. Using the heat equation as the case example, the first step is to compute the nodal temperature based on particle values in the following manner:

$$C_i^* = \sum_p m_p C_p^n \phi_{ip}, \tag{6.3}$$

$$\tau_i^n = \sum_p m_p C_p^n T_p^n \phi_{ip}, \tag{6.4}$$

$$T_i^n = \frac{\tau_i^n}{C_i^*}, \tag{6.5}$$

where m_p, C_p, T_p, and T_i are the particle mass, particle heat capacity, particle temperature, and grid temperature, respectively. The next step is to compute the nodal heat load,

$$Q_i^n = \sum_p V_p^n (\mathbf{q}_p^n \cdot \nabla \phi_{ip}), \tag{6.6}$$

where $\mathbf{q}_p^n$ is the particle heat flux and ϕ_{ip} is the nodal basis function. The nodal temperature rate is then computed,

$$\dot{T}_i^n = \frac{Q_i^n}{C_i^*}. \tag{6.7}$$

Nodal temperature values are updated,

$$T_i^{n+1} = T_i^n + \Delta t \dot{T}_i \tag{6.8}$$

and boundary conditions are applied. Particle values are then updated based on the updated nodal temperatures,

$$T_p^{n+1} = T_p^n + \Delta t \sum_i \dot{T}_i^n. \tag{6.9}$$

Using the updated nodal temperature the particle heat flux can then be computed,

$$\mathbf{q}_p^{n+1} = \mathbf{k}_p \sum_i \nabla \phi_{ip} T_i^{n+1}, \tag{6.10}$$

were $\mathbf{k}_p$ is the thermal conductivity tensor. For a complete discussion on this approach to modeling diffusion, the reader is referred to [167, 171, 40].

An alternate approach is to use a mesh base method such as FVM or FEM to solve the heat equation using the values stored at the MPM particles. An example of this type of approach has been proposed by Stomakhin et al. [90]. In their approach they use a cell-centered finite volume method (FVM) to solve the heat equation. First temperature is mapped to cell-centers,

$$C_c^* = \sum_p m_p C_p \phi_{cp}, \tag{6.11}$$

$$T_c^n = \frac{1}{C_c^*} \sum_p m_p C_p T_p \phi_{cp}, \tag{6.12}$$

where the subscript c represents the cell-center index, followed by mapping thermal conductivity to face-centers,

$$m_c = \sum_p m_p \phi_{cp}, \tag{6.13}$$

$$k_c^n = \frac{1}{m_c} \sum_p m_p k_p^n \phi_{cp}, \tag{6.14}$$

where subscript f is the face-center index.

Once values have been mapped to the grid an implicit FVM solver is used to compute the updated temperature values on the grid. Particle values can then be updated using the old and new cell-center values,

$$T_p^{n+1} = T_p^n + \sum_c (T_c^{n+1} - T_c^n)\phi_{cp}. \tag{6.15}$$

An alternate approach used by [90] is to use a weighted combination of the updated cell-center temperature and the change in the cell-center temperature, this is similar in approach to the PIC/FLIP method for updating particle velocities [84] as described in Chapter 5.3,

$$T_p^{n+1} = \beta \left(T_p^n + \sum_c (T_c^{n+1} - T_c^n) S_{cp} \right) + (1 - \beta) \sum_c T_c^{n+1}. \tag{6.16}$$

6.1 MPM in the Reference Configuration

The traditional approach to MPM is to solve the model equations in the spatial configuration. Alternates to this approach, such as moving-mesh MPM [172] and total lagrangian MPM (TLMPM) [173], solve the equations in the reference configuration and then update the particles positions in the spatial configuration. For purposes of simplification the term reference configuration MPM (REFMPM) will be used when discussing MPM in the reference configuration, generally, and moving-mesh MPM or TLMPM when discussing specifics of their respective approaches. The process of solving the model problems in the reference configuration has been used with particle methods [174, 175] and FEM [176, 177] for decades. As noted by [173], there are advantages to solving the model equation in the reference configuration. First, because the particles positions in the reference configurations don't change the weights, ϕ_{ip}, and gradients, $\nabla\phi_{ip}$, of the basis functions remain the same for each timestep and therefore only need to be computed once. Second, given that mass is conserved, the nodal mass needs only to be compute once. Computing nodal masses

once becomes especially advantageous when using the full versus lumped mass matrix [16]. Third, because particles don't move, numerical fracture is avoided. In traditional approaches to MPM, numerical fracture occurs when the number of particles in a cell or cells is not capable of modeling large strains correctly [178]. Methods used to avoid this nonphysical behavior are the use particle splitting [178] or deforming the particle domain [86, 88].

The conservation of linear momentum equation in the reference configuration is defined as [58]

$$\rho_0 \frac{\partial \mathbf{v}}{\partial t} = \nabla_0 \cdot \mathbf{P} \tag{6.17}$$

where ∇_0 represents the gradient with respect to the reference coordinates and $\mathbf{P}$ is the first Piola-Kirchhoff stress tensor. The relationship between density in the reference and spatial configuration is defined by [58]

$$\rho_0 = J\rho, \tag{6.18}$$

where $J = det(\mathbf{F})$ and the relationship between the first Piola-Kirchhoff and Cauchy stress tensors is defined as [58],

$$\mathbf{P} = J\boldsymbol{\sigma}\mathbf{F}^{-T}. \tag{6.19}$$

6.1.1 The Algorithm

The methods used by Steffen et al. [172] and De Vaucorbeil et al. [173], while similar, do have slight variations. A description of the general of REFMPM, along with discussion on differences between moving-mesh MPM and TLMPM, are presented below. REFMPM follows similar steps to those of MPM. For an explicit stress last approach the first step is to map particle values to the grid.

$$m_i = \sum_p \phi_{ip} m_p, \tag{6.20}$$

$$\mathbf{v}_i^n = \frac{1}{m_i} \sum_p \phi_{ip} m_p \mathbf{v}_p, \tag{6.21}$$

$$\mathbf{f}_i^{int,n} = -\sum_p \mathbf{P} \nabla_0 \phi_{ip} V_p^0, \tag{6.22}$$

$$\mathbf{f}_i^{ext,n} = \sum_p m_p \mathbf{b}_p, \tag{6.23}$$

$$\mathbf{a}_i^n = \frac{1}{m_i}(\mathbf{f}_i^{int,n} + \mathbf{f}_i^{ext,n}). \tag{6.24}$$

In this set of equations the mapping function, ϕ_{ip}, is in terms of the reference position of the particle, i.e., $\phi_{ip} = \phi_i(\mathbf{X}_p)$.

At this point in the algorithm there are variety of time integration approaches that can be used as demonstrated by Berzins [144]. moving-mesh MPM uses a symplectic Euler time integration, where nodal velocity and displacements are updated as follows:

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \Delta t \mathbf{a}_i^n, \tag{6.25}$$

$$u_i^{n+1} = \mathbf{u}_i^n + \Delta t \mathbf{v}_i^{n+1}. \tag{6.26}$$

Particle velocity and displacements are then updated,

$$\mathbf{v}_p^{n+1} = \mathbf{v}_p^n + \Delta t \sum_i \phi_{ip} \mathbf{a}_i^n, \tag{6.27}$$

$$\mathbf{u}_p^{n+1} = \mathbf{u}_p^n + \Delta t \sum_i \phi_{ip} \mathbf{v}_i^{n+1}. \tag{6.28}$$

TLMPM also uses symplectic Euler scheme, but with a few modifications. First the nodal velocity is updated,

$$\mathbf{v}_i^* = \mathbf{v}_i^n + \mathbf{a}_i^n \Delta t, \tag{6.29}$$

but the updated nodal velocity, $\mathbf{v}_i^*$, is a temporary nodal value. Temporary, particle values for velocity and acceleration are computed based on the nodal values and the particle position is updated,

$$\mathbf{v}_p^* = \sum_i \phi_{ip} \mathbf{v}_i^*, \tag{6.30}$$

$$\mathbf{a}_p^* = \sum_i \phi_{ip} \mathbf{a}_i^n, \tag{6.31}$$

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \mathbf{v}_p^*. \tag{6.32}$$

Particle velocities are then updated using the hybrid PIC/FLIP update [84, 98],

$$\mathbf{v}_p^{n+1} = (1-\beta)\mathbf{v}_p^* + \beta(\mathbf{v}_p^n + \Delta t \mathbf{a}_p^*). \tag{6.33}$$

The nodal velocity is then updated by mapping the updated particle velocity back to the node,

$$\mathbf{v}_i^{n+1} = \frac{1}{m_i} \sum_p m_p \mathbf{v}_p^{n+1}. \tag{6.34}$$

The process for updating particle deformation gradients varies between moving-mesh MPM and TLMPM in execution, but as will be shown later, are equivalent. The moving-

mesh MPM approach utilizes the nodal displacement to compute the deformation gradient directly,

$$\mathbf{F}_p^{n+1} = \mathbf{I} + \sum_i \mathbf{u}_i^{n+1} \otimes \nabla_0 \phi_{ip}. \tag{6.35}$$

TLMPM updates the particle deformation gradient using the reference velocity gradient,

$$\dot{\mathbf{F}}_p = \sum_i \mathbf{v}_i^{n+1} \otimes \nabla_0 \phi_{ip}, \tag{6.36}$$

$$\mathbf{F}_p^{n+1} = \mathbf{F}_p^n + \Delta t \dot{\mathbf{F}}_p. \tag{6.37}$$

A close examination of the two approaches will show that they are equivalent. Substituting (6.26) into (6.35) leads to

$$\mathbf{F}_p^{n+1} = \mathbf{I} + \sum_i \mathbf{u}_i^n \otimes \nabla_0 \phi_{ip} + \Delta t \sum_i \mathbf{v}_i^{n+1} \otimes \nabla_0 \phi_{ip}. \tag{6.38}$$

Noting the relationships in (6.35) and (6.36) then the above equation is the same as (6.37).

Using the updated particle deformation gradient particle volumes and stresses are then computed,

$$V_p^{n+1} = J V_p^0, \tag{6.39}$$

$$\mathbf{P}_p^{n+1} = \mathbf{P}(\mathbf{F}_p^{n+1}), \tag{6.40}$$

where $\mathbf{P}(\cdot)$ is dependent upon the chosen constitutive model.

6.1.2 Stability Analysis

Von Neumann stability analysis has been applied to PIC methods [124] and to MPM [120, 121]. Using Von Neumann analyis for developing stability criteria for MPM is not always fruitful. Because particles move from cell to cell and the distribution of particles does not remain evenly distributed, as compared to finite difference methods, it becomes difficult to correctly formulate the equations analyis. In the analysis provided by Ni and Zhang [121], they show that in the case of cell crossing, terms within the formulation become complex, which violates the requirements of the analysis. Bai and Schroeder [120] provide an analysis for MPM in multiple dimension, but their work requires uniformity of particle mass and volume, even distribution of particles, and that particle displacements remain near zero.

Stability analysis for REFMPM has an advantage over variants of traditional MPM in that the evaluation of the basis functions is done in relation to the reference configuration

of the particle and as such the values do not change from time step to time step. Because the particles don't move the complexity of the analysis is reduced and does not require the assumption that particle displacements be near zero [120]. The analysis provide here follows a similar approach to that done in [124, 121]. For this analysis the following assumptions are made:

1. All particles are evenly spaced over the entire domain.
2. The background grid is uniform and Cartesian, in that the spacing between nodes in each cardinal direction is the same and $\Delta x = \Delta y = \Delta z$.
3. The distribution of particles is such that there is symmetry of particle about a grid node.
4. No body forces are applied.
5. The domain of the problem is infinite or has periodic boundary conditions.

From assumption 1 all particle have the same mass and volume, and as a result the nodal mass for each time step is

$$m_i = m_p \sum_p \phi_{ip}. \tag{6.41}$$

One observation about the computing of the nodal mass, is that for a 1D grid using linear basis functions, with evenly spaced particles that are symmetric about node i, the above equation reduces to

$$m_i = N_i m_p, \tag{6.42}$$

where N_i is the number of particle in a cell. For this analysis linear basis functions on a one dimensional grid, with two particles per cell, and a linear elastic constitutive model,

$$\mathbf{P} = E(\mathbf{F} - \mathbf{I}), \tag{6.43}$$

will be used.

As was noted earlier, the approaches for computing the deformation gradient are equivalent for moving-mesh MPM and TLMPM. For the purposes of this analysis the moving-mesh MPM approach will be used. The nodal displacement defined as

$$\mathbf{u}_i^{n+1} = \mathbf{x}_i^{n+1} - \mathbf{X}_i. \tag{6.44}$$

The particle deformation gradient for time step $n+1$ is then computed as follows:

$$\nabla_0 \mathbf{u}_p^{n+1} = \sum_i \nabla_0 \phi_{ip} \mathbf{u}_i^{n+1}, \tag{6.45}$$

$$\mathbf{F}_p^{n+1} = \nabla_0 \mathbf{u}_p^{n+1} + \mathbf{I}. \tag{6.46}$$

Note that for linear basis function on a one dimensional grid the gradient of the basis function is $\nabla_0 \phi_{ip} = \pm 1/\Delta x$, thus in the one dimensional case the particle displacement gradient, for a particle between nodes i and $i+1$, is then defined as

$$\nabla_0 \mathbf{u}_p^{n+1} = \frac{\mathbf{u}_{i+1}^{n+1} - \mathbf{u}_i^{n+1}}{\Delta x}. \tag{6.47}$$

The particle stress can then be computed using (6.43), (6.46), and (6.47),

$$\mathbf{P}_p^{n+1} = E \frac{\mathbf{u}_{i+1}^{n+1} - \mathbf{u}_i^{n+1}}{\Delta x}. \tag{6.48}$$

The force calculation defined by (6.22) can now be rewritten as

$$\mathbf{f}_i^n = -\sum_p^i \nabla_0 \phi_{ip} \mathbf{P}_p^n V_p^0 - \sum_p^{i+1} \nabla_0 \phi_{ip} \mathbf{P}_p^n V_p^0, \tag{6.49}$$

$$= -\sum_p^i E \frac{\mathbf{u}_i^n - \mathbf{u}_{i-1}^n}{\Delta x^2} V_p^0 + \sum_p^{i+1} E \frac{\mathbf{u}_{i+1}^n - \mathbf{u}_i^n}{\Delta x^2} V_p^0, \tag{6.50}$$

$$= -2E \frac{\mathbf{u}_i^n - \mathbf{u}_{i-1}^n}{\Delta x^2} V_p^0 + 2E \frac{\mathbf{u}_{i+1}^n - \mathbf{u}_i^n}{\Delta x^2} V_p^0, \tag{6.51}$$

$$= 2EV_p^0 \frac{\mathbf{u}_{i+1}^n - 2\mathbf{u}_i^n + \mathbf{u}_{i-1}^n}{\Delta x^2}, \tag{6.52}$$

where the summation notation, $\sum_p^i$, indicates summing over the particles between nodes $i-1$ and i. Noting the observation in (6.42) the nodal acceleration as defined by (6.24)

$$\mathbf{a}_i^n = \frac{1}{m_i} \mathbf{f}_i^n, \tag{6.53}$$

$$= \frac{2EV_p^0}{m_i} \frac{(\mathbf{u}_{i+1}^n - 2\mathbf{u}_i^n + \mathbf{u}_{i-1}^n)}{\Delta x^2}, \tag{6.54}$$

$$= \frac{c^2}{\Delta x^2} (\mathbf{u}_{i+1}^n - 2\mathbf{u}_i^n + \mathbf{u}_{i-1}^n), \tag{6.55}$$

where $c = \sqrt{\frac{E}{\rho_0}}$.

The updates for the nodal velocity and the displacement updates for n and $n+1$ are defined as

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \Delta t \mathbf{a}_i^n, \tag{6.56}$$

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n + \Delta t \mathbf{v}_i^{n+1}, \tag{6.57}$$

$$\mathbf{u}_i^n = \mathbf{u}_i^{n-1} + \Delta t \mathbf{v}_i^n. \tag{6.58}$$

Using the above equations, the velocity update can be rewritten as

$$\mathbf{u}_i^{n+1} - 2\mathbf{u}_i^n + \mathbf{u}_i^{n-1} = \Delta t^2 \mathbf{a}_i^n. \tag{6.59}$$

Combining the above equation with (6.55) leads to

$$\mathbf{u}_i^{n+1} - 2\mathbf{u}_i^n + \mathbf{u}_i^{n-1} = k^2(\mathbf{u}_{i+1}^n - 2\mathbf{u}_i^n + \mathbf{u}_{i-1}^n). \tag{6.60}$$

where $k = \frac{c\Delta t}{\Delta x}$.

At this point, (6.60) is in the form of the standard finite difference stencil for the wave equation which has the stability constraint $k \leq 1$ [179].

6.2 Diffusion in the Reference Configuration

The heat equation in the reference configuration is defined as,

$$\rho_0 c \frac{\partial T}{\partial t} + \nabla_0 \cdot \mathbf{Q} = 0, \tag{6.61}$$

where $\mathbf{Q}$ is the Piola-Kirchhoff or nominal heat flux [58].

6.2.1 Constitutive Relationship

The nominal heat flux is defined as [58],

$$\mathbf{Q} = -\mathbf{k}_0 \nabla_0 T \tag{6.62}$$

where $\mathbf{k}_0$ is the conductivity tensor in the reference configuration. Heat flux in spatial configuration, also known as the Cauchy heat flux [58], is defined as,

$$\mathbf{q} = -\mathbf{k}\nabla T \tag{6.63}$$

where $\mathbf{k}$ is the conductivity tensor in the spatial configuration. The total flux of heat entering the body, Ω, is the same in either configuration,

$$\int_{\partial\Omega_0} \mathbf{Q} \cdot \mathbf{N} dA = \int_{\partial\Omega} \mathbf{q} \cdot \mathbf{n} da. \tag{6.64}$$

Using Nanson's formula (6.64) can be re-written as

$$\int_{\partial\Omega_0} \mathbf{Q} \cdot \mathbf{N} dA = \int_{\partial\Omega_0} \mathbf{q} \cdot J\mathbf{F}\mathbf{N} dA, \tag{6.65}$$

$$= \int_{\partial\Omega_0} J\mathbf{F}^{-1}\mathbf{q} \cdot \mathbf{N} dA, \tag{6.66}$$

which then leads to the following relationship between the heat flux in the two configurations:

$$\mathbf{Q} = J\mathbf{F}^{-1}\mathbf{q}. \tag{6.67}$$

The relationship between a gradient of a scalar field in the reference configuration, $\Phi(\mathbf{X}, t)$, and spatial configuration, $\Phi(\mathbf{x}, t)$, is defined as [58]

$$\nabla_0 \Phi = \mathbf{F}^T \nabla \Phi. \tag{6.68}$$

By combining (6.62), (6.63), and (6.67) along with noting the relationship defined by (6.68) leads to the following:

$$\mathbf{k}_0 \nabla_0 T = J\mathbf{F}^{-1}\mathbf{k}\nabla T, \tag{6.69}$$

$$= J\mathbf{F}^{-1}\mathbf{k}\mathbf{F}^{-T}\nabla_0 T. \tag{6.70}$$

From the above equation we get the mapping between the conductivity in the reference and spatial configurations,

$$\mathbf{k}_0 = J\mathbf{F}^{-1}\mathbf{k}\mathbf{F}^{-T}. \tag{6.71}$$

In the case that the conductivity is isotropic, then (6.71) can be simplified to

$$k_0 = Jk\mathbf{C}^{-1}, \tag{6.72}$$

where $\mathbf{C}$ is the right Cauchy-Green tensor, $\mathbf{F}^T\mathbf{F}$ [58]. For a more complete discussion on the relationship between the heat flux in the reference and spatial configuration, the reader is referred to Holzapfel [58].

6.3 Modeling Diffusion with REFMPM

Two approaches were discussed for modeling diffusion using MPM. The hybrid approach [90] using FVM coupled with REFMPM could be advantageous in the reference configuration, but the focus here will be on developing an approach similar to [164, 167].

6.3.1 REFMPM Diffusion

The process of modeling diffusion with REFMPM follows steps similar to those described in the intro to this chapter. The first step is to map mass and temperature to nodal values as is done in (6.3), (6.4), and (6.5). The difference is that the nodal basis function are

evaluated with respect to the particles reference position and not spatial. The nodal heat load is then computed,

$$Q_i^n = \sum_p V_p^0 (\mathbf{Q}_p^n \cdot \nabla_0 \phi_{ip}) \tag{6.73}$$

where V_p^0 is the particle volume in the reference configuration and $\mathbf{Q}_p^n$ is the particle nominal heat flux. The process for computing the nodal temperature rate, updating nodal temperatures, applying boundary conditions, and updating particle temperatures is the same as (6.7), (6.8), and (6.9). The particle heat flux can then be computed using the updated nodal temperature and the updated particle deformation gradients,

$$\nabla_0 T_p^{n+1} = \sum_i \nabla_0 \phi_{ip} T_i^{n+1}, \tag{6.74}$$

$$\mathbf{Q}_p^{n+1} = J \mathbf{F}_p^{-1} \mathbf{k}_p \mathbf{F}_p^{-T} \nabla_0 T_p^{n+1}. \tag{6.75}$$

6.3.2 Stability Analysis of REFMPM Diffusion

For this analysis the same assumptions apply that were used in section 6.1.2, in terms of particle placement, number of particles per cell, nodal spacing, and boundary conditions. Also, the problem will be one dimensional and linear basis functions will be used. Note that in the one dimensional case, the deformation gradient, $\mathbf{F}$, and the Jacobian, $J = det(\mathbf{F})$, are equivalent,

$$\mathbf{F} = J = det(\mathbf{F}), \tag{6.76}$$

and thus the thermal conductivity term defined by (6.72) can be reduced,

$$k_0 = Jk(\mathbf{F}^T \mathbf{F})^{-1}, \tag{6.77}$$

$$k_0 = \frac{Jk}{J^2}, \tag{6.78}$$

$$k_0 = \frac{k}{J}. \tag{6.79}$$

The nodal temperature update for REFMPM diffusion is defined as

$$T_i^{n+1} = T_i^n + \Delta t \dot{T}_i^n, \tag{6.80}$$

where $\dot{T}_i^n$ is defined as

$$\dot{T} = \frac{1}{m_p C_p \sum_p \phi_{ip}} \sum_p \nabla_0 \phi_{ip} \cdot \mathbf{Q}_p^n V_p^0. \tag{6.81}$$

Using linear basis functions the heat flux is defined as

$$\mathbf{Q}_p^n = \frac{k}{J_p} \frac{T_{i+1}^n - T_i^n}{\Delta x}. \tag{6.82}$$

Note that J_p is the same for all particles in the nodal interval i to $i+1$, because the gradient of the basis function is a constant. Using the same logic that was applied for in (6.49), (6.50), (6.51), and (6.52) the nodal temperature rate can be rewritten as

$$\dot{T}_i^n = \frac{1}{2m_pC_p}\left[\sum_p^i \nabla_0\phi_{ip}\mathbf{Q}_p^n V_p^0 - \sum_p^{i+1}\nabla_0\phi_{ip}\mathbf{Q}_p^n V_p^0\right], \tag{6.83}$$

$$= \frac{1}{2m_pC_p}\left[\sum_p^i \frac{k}{J_p}\frac{T_i^n - T_{i-1}^n}{\Delta x^2}V_p^0 - \sum_p^{i+1}\frac{k}{J_p}\frac{T_{i+1}^n - T_i^n}{\Delta x^2}V_p^0\right], \tag{6.84}$$

$$= \frac{kV_p^0}{m_pC_p}\left[\frac{1}{J_{pi}}\frac{T_i^n - T_{i-1}^n}{\Delta x^2} - \frac{1}{J_{pi+1}}\frac{T_{i+1}^n - T_i^n}{\Delta x^2}\right], \tag{6.85}$$

$$= \frac{K}{\Delta x^2}\left[\frac{1}{J_{pi}}(T_i^n - T_{i-1}^n) - \frac{1}{J_{pi+1}}(T_{i+1}^n - T_i^n)\right], \tag{6.86}$$

where $K = k/(\rho_p^0 C_p)$ and J_{pi} represents the Jacobian for the interval i. Substituting in the newly derived nodal temperature rate into (6.80) leads to,

$$T_i^{n+1} = T_i^n - B(\alpha_i T_{i-1}^n - (\alpha_i + \alpha_{i+1})T_i^n + \alpha_{i+1}T_{i+1}^n) \tag{6.87}$$

where $\alpha_i = 1/J_{pi}$, $\alpha_{i+1} = 1/J_{pi+1}$, and $B = K\Delta t/\Delta x^2$.

Using Von Neumann analysis, let

$$T_i^n = G^n \mathbf{e}^{i\beta j\Delta x} \tag{6.88}$$

where G^n is the amplification factor and in this particular case i is the imaginary number and j represents the node index. Substituting (6.88) into (6.87) leads to

$$G^{n+1}\mathbf{e}^{i\beta j\Delta x} = G^n\mathbf{e}^{i\beta j\Delta x} - B(\alpha_i G^n \mathbf{e}^{i\beta j-1\Delta x} - (\alpha_i + \alpha_{i+1})G^n\mathbf{e}^{i\beta j\Delta x} + \alpha_{i+1}G^n\mathbf{e}^{i\beta j+1\Delta x}). \tag{6.89}$$

The above equation can be further reduced to

$$G = 1 - B(\alpha_i \mathbf{e}^{-i\beta\Delta x} - (\alpha_i + \alpha_{i+1}) + \alpha_{i+1}\mathbf{e}^{i\beta\Delta x}). \tag{6.90}$$

As compared to the analysis done for the REFMPM solution of the linear momentum equation, the REFMPM diffusion solution does not resolve to a nice finite difference stencil. From this analysis it can be see that stability is dependent upon both $K\Delta t/\Delta x^2$ and values

of the particle Jacobians on either side of node i. For the sake of discussion let us assume that $\alpha_i \approx \alpha_{i+1}$. If this is the case, then (6.90) is written as

$$G = 1 - 2B\alpha(\cos(\beta\Delta x) - 1), \tag{6.91}$$

which is similar to that of a standard finite difference solution for the diffusion equation with the added α term. From (6.91) it can be seen that stability is contingent upon the particle deformation gradients.

6.4 Nullspace Issues Associated with MPM Diffusion

Modeling diffusion using the methods described at the beginning of this chapter and in Section 6.3.1 can introduce numerical oscillations in temperature field defined by values at the particles. These oscillations are most prominent at interface and boundaries. Nairn [180] has pointed out these oscillations are a result of the null-space noise that is introduced as particle values are updated from nodal values.

Fig. 6.1 is an example of this type of nullspace induced numerical noise. In this figure there are three images. The one on the left is the analytic solution evaluated at the particles. The middle and right images are the temperature values at the particles and nodes, respectively, as computed using REFMPM diffusion. An observation of the middle image will show that there is a layer of particles around the boundary that is at a lower temperature than those the next layer in from the boundary. This oscillation in the temperature field near the boundaries is not represented in the nodal solution, as shown by the image on the right. These oscillations are an example of the numerical noise that is introduced at the particle as a result of the mismatch in dimensionality between particles and nodes. A full description of the numerical experiment that resulted in these images is discussed in the experimental results section below.

One method for overcoming the nullspace issue is to use the PIC/FLIP approach that was presented in the introduction to this chapter and is posted here for convenience,

$$T_p^{n+1} = \beta\left(T_p^n + \sum_c (T_c^{n+1} - T_c^n) S_{cp}\right) + (1-\beta)\sum_c T_c^{n+1}. \tag{6.92}$$

When modeling mechanics problems it is well known that the PIC approach for updating particle velocities,

$$\mathbf{v}_p^{n+1} = \sum_i \mathbf{v}_i^{n+1}, \tag{6.93}$$

is much more stable than the FLIP approach,

$$\mathbf{v}_p^{n+1} = \mathbf{v}_p^n + \Delta t \sum_i \mathbf{a}_i^{n+1}, \tag{6.94}$$

but at the cost of excess numerical diffusion [99]. A hybrid of the two combines the positives of both approaches while mitigating some of the negatives. The same principle applies to the particle temperature update [40]. By using the PIC/FLIP approach the nullspace noise is not apparent, but even for values of $\beta = 0.995$ there is some excess numerical diffusion that which reduces the accuracy of the temperature update, as will be seen in the experiment results section below.

6.5 Experimental Results

6.5.1 One Dimensional Heat Transfer Problem

A one dimensional heat transfer problem is presented to validate the REFMPM diffusion approach. In this example a standard heat transfer problem is used with density, specific heat capacity, and conductivity of unit value, i.e., $\rho_0 C = 1$ / (m °C) and $k = 1$ J/(m^3 °C). The length of the domain is, $l = 1$m, and is subdivided into 20 cells of equal width. GIMP basis functions are used with two particles per cell. The initial temperature is $T_0 = 0$ °C with the left boundary condition being, $T_l = 1$ °C, and the right, $T_r = 0$ °C. The time step size was 1e-5 s. The analytic solution for this problem is [181],

$$T(x,t) = T_l + \frac{T_r - T_l}{l}x + 2\sum_j^{\infty} \frac{(T_0 - T_l) - (T_0 - T_r)(-1)^j}{j\pi} \exp(-K\lambda^2 t)\sin(x\lambda) \tag{6.95}$$

where $K = k/(\rho_0 C)$ and $\lambda = j\pi/l$.

This numerical study looks at the accuracy of the REFMPM diffusion along with the accuracy of using the PIC/FLIP temperature update. Two simulations where run, the first set the weighting value to $\beta =$, which means that the particle temperature update uses just the FLIP type update,

$$T_p^{n+1} = T_p^n + \Delta t \sum_i \dot{T}_i^{n+1}. \tag{6.96}$$

The second simulation used a value of $beta = .99$. Fig. 6.2 shows the results of the simulation at time 0.025 s. In the plot the use of the PIC/FLIP update, even by a small

percentage, decreases the accuracy of the approach through the introduction of excess numerical diffusion. Table 6.1 shows the error results at times 0.005 s, 0.025s, and 0.045s. The errors for each chosen time step are computed using the root mean square,

$$err = \sqrt{\frac{1}{N}\sum_{p}^{N}(T_p^n - T(x_p, t^n))}. \tag{6.97}$$

6.5.2 Two Dimensional Heat Transfer Problem

The second set of numerical experiments simulates heat transfer on a two dimensional square plate. The dimension of the plate is 1m x 1m and the cells is 20 x 20 with four particles per cell. The values for density, specific heat capacity, and conductivity are of unit value, i.e., $\rho_0 C = 1$ / (m °C) and $k = 1$ J/(m^3 °C). The initial temperature of the plate is $T_0 = 0$ °C. At the start of the simulation, a boundary condition of 1 °C is applied along the boundaries, i.e., the top, left, bottom, and right boundary conditions being, $T_t = T_l = T_b = T_r = T_1 = 1$ °C. The analytic solution for this problem is [171],

$$T(x, y, t) = T_1 + 16\frac{T_0 - T_1}{\pi^2}\sum_{i=1,3,\ldots}^{\infty}\sum_{j=1,3,\ldots}^{\infty}\frac{1}{ij}\exp\left(-\pi^2 t\left(\frac{i^2}{l^2} + \frac{j^2}{h^2}\right)\right)\sin\left(\frac{i\pi x}{l}\right)\sin\left(\frac{j\pi y}{h}\right). \tag{6.98}$$

Two simulations were run in a similar fashion to the previous example. The first used the full FLIP version, i.e., $\beta = 1$, and the second used a weighting value of $\beta = 0.995$. Both simulations were run using a time step size of 0.005s. The errors for each chosen time step are computed using the root mean square. Table 6.2 shows the error results at times 0.005s, 0.025s, and 0.045s.

From the results it can been see that full FLIP version is less accurate than the PIC/FLIP approach when looking at the particle values, but looking at the nodal values the FLIP approach is the most accurate. The nullspace noise causes a decrease in accuracy in the particles. On the other hand the excess diffusion from the PIC/FLIP approach, while preventing the numerical noise at the particles, does lead to a decrease in accuracy when looking at the nodal values.

Fig. 6.3 shows results of the computed temperature values at the particles for the two approaches along with the analytic solutions at times 0.005s, 0.025s, and 0.045s. As was discussed in section 6.4, the images of the full FLIP, $\beta = 1$, results show the prominence of the nullspace noise near the plate boundaries. In the PIC/FLIP version the nullspace noise

does not exist, but from visual examination it can be seen that temperature is diffusing into the center of the plate at a higher rate than that of both the FLIP and analytic solutions. The higher rate of diffusion comes from the excess numerical diffusion the is introduced by the PIC/FLIP approach.

Fig. 6.4 shows the temperature results at the nodes. Looking at the nodal values for $\beta = 1.0$, there is no noise and the results are close to the analytic solutions values. The $\beta = .995$ appear to be more diffuse than those of the both $\beta = 1.0$ and the analytic results.

6.6 Conclusion

A discussion on modeling diffusion using MPM along with a review of the different approaches has been presented. All of the approaches presented model diffusion in the spatial configuration. For mechanics problems, modeling in the reference configuration is not new and has been done using particle methods and FEM. Using MPM to model problem in the reference configuration, while not common, has been explored. The Moving Mesh MPM and TLMPM are both examples of this approach.

In this chapter a stability analysis of REFMPM showed that, under the correct assumptions, the nodal displacements stencil structure is the same as the finite difference stencil for the wave equation. The equality in stencil structure between REFMPM and a finite difference approach comes from the fact that all computation is done in the reference configuration, which means that particle positions do not change and all particle/node weights and gradients remain the same for each time step. Because of the relationship between REFMPM nodal displacements and the finite difference structure, the stability constraints on the time step size are already known.

This chapter also introduces a REFMPM diffusion approach that follows the same process as the traditional MPM approach to modeling diffusion. A stability analysis of this approach is presented. The analysis shows that the stability criteria is not simply a function of the conductivity or diffusivity and grid spacing, but is also dependent upon particle deformation gradients.

Nullspace noise in MPM diffusion is also discussed along with use of the PIC/FLIP approach as a mitigation approach. Numerical examples in both one and two dimensions compares the use of the PIC/FLIP approach to the full FLIP approach.

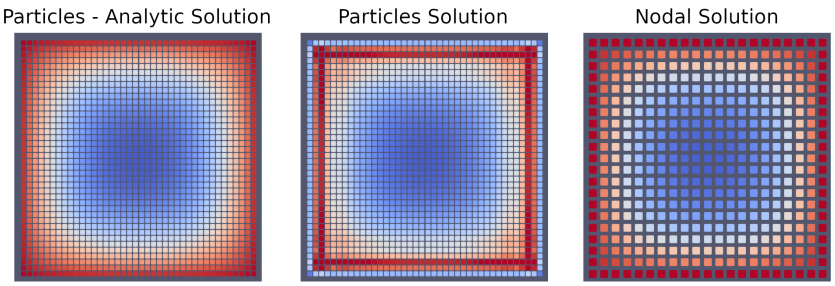

Fig. 6.1. Nullspace example.

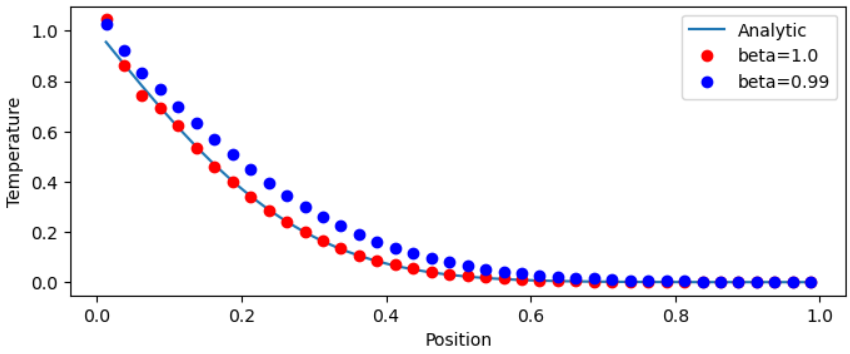

Fig. 6.2. PIC/FLIP approach using $\beta = 1.0$ and $\beta = 0.99$ at 0.025s.

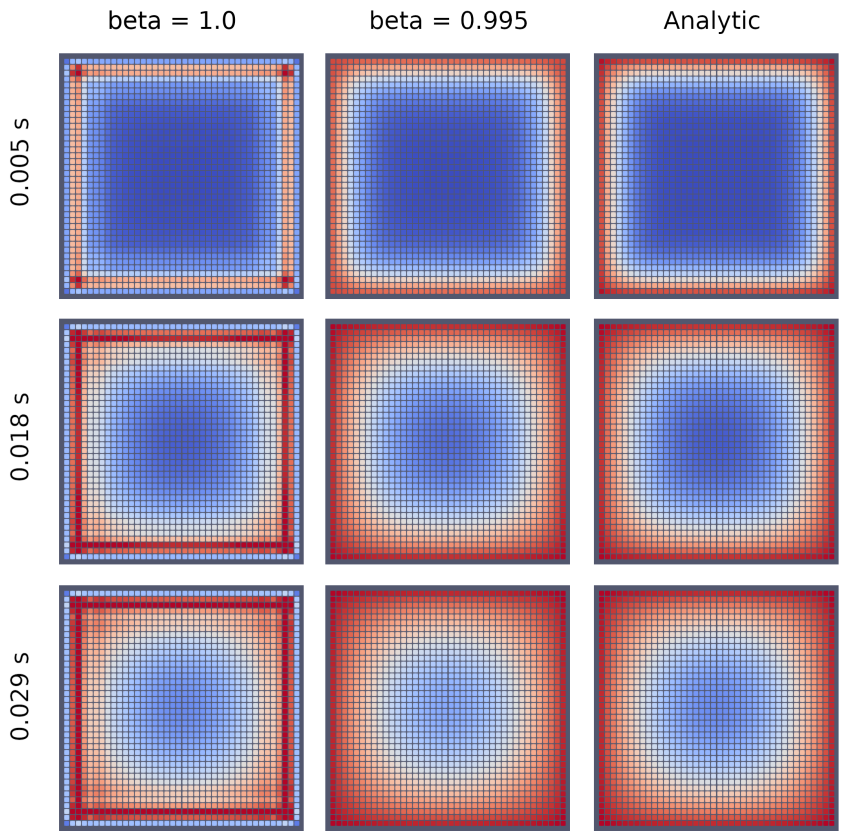

Fig. 6.3. Particle temperatures for different beta values and true solution at different time steps.

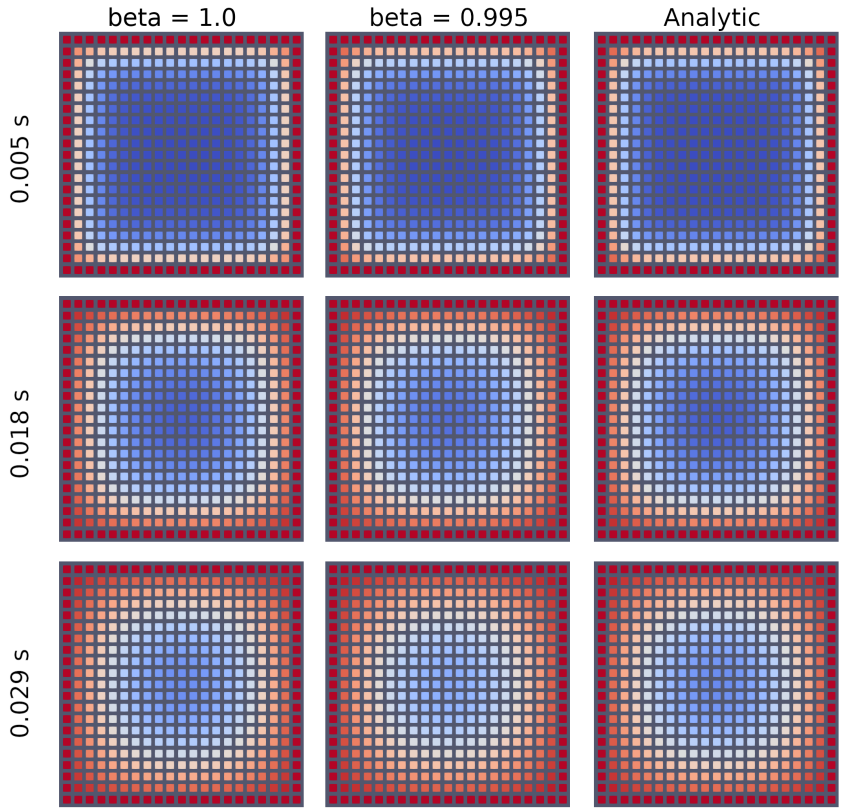

Fig. 6.4. Nodal temperatures for different beta values and true solution at different time steps.

Table 6.1. One dimensional diffusion example comparing PIC/FLIP updates.

Time (s)	beta=1.0	beta = 0.99
0.005	0.0375	0.0536
0.025	0.0156	0.0615
0.045	0.0114	0.0675

Table 6.2. Two dimensional diffusion heat transfer error results.

Time (s)	particles beta=1.0	nodes beta=1.0	particles beta = 0.995	nodes beta = 0.995
0.005	0.22411582	0.02122968	0.03027875	0.03027875
0.025	0.22238649	0.00717621	0.04500869	0.04500869
0.045	0.22228594	0.00455933	0.05400095	0.05400095

CHAPTER 7

USING THE MATERIAL POINT METHOD TO MODEL CHEMICAL/MECHANICAL COUPLING IN THE DEFORMATION OF A SILICON ANODE

Electrochemical storage devices are becoming pervasive in today's society. Electronic devices ranging from smart phones to electric vehicles all depend on efficient, robust, and high-capacity electrochemical storage. Driven by this demand for greater energy storage capacity, new materials are being sought to improve the performance of batteries. One promising material that is currently being explored to improve performance is silicon. Silicon has a theoretical charge capacity in the range of 3500 mAhg^{-1} to 4200 mAhg^{-1} [182, 183], which is 10 times larger than that of current anode materials such as graphite at 350 mAhg^{-1} [184]. The large theoretical charge capacity comes from silicon's ability to accommodate multiple lithium ions per silicon atom. As a silicon anode approaches full lithiation, the ratio of lithium ions to silicon atoms is 3.75 to 1 [56]. This ratio of lithium ions to silicon leads to a large volume change of up to $\sim$ 280% [20]. This large change in volume over multiple charge/discharge cycles can lead to mechanical failures that decrease the charge capacity, thus reducing the effectiveness of using silicon as an anode material.

A key observable trait of the lithiation process is the notable lithiation front that occurs as lithium ions diffuse through the silicon material. Experimental results presented by Wang et al. [20] show that as lithium diffuses through the silicon material, there exists a sharp phase transition between the lithiated silicon and the pure amorphous silicon. This sharp phase transition produces large strains on the lithium rich side of the phase transition. Wang et al. [20] report strains in the range of 160% in the region around the phase transition which produce a large percentage of the total volume expansion. The normalized concentration of lithium in the region of the phase transition is approximately 67% or 2.5 lithiums to

1 silicon [20]. Outside the region of the phase transition, the concentrations gradients for lithium are relatively small in the lithiated regions of the lithiation front.

By gaining an understanding of the chemical and mechanical changes that occur during the lithiation and delithiation phases of a charge cycle, new anode configurations and geometries can be explored that reduce the mechanical failure points within an anode. Numerous physical experiments have been performed testing different anode configurations, ranging from thin films [185], to nano pillars [186], to honeycomb structures [187]. Through the use of computational simulations, new anode configurations can be explored more easily, which can lead to better insights into the configurations used in physical experiments.

The study of diffusion-induced stress can be found in work dating back to the 1960s. Prussin [67] formulated a model that couples the diffusion of a solute in a single-crystal silicon wafer to the generation of stresses and the resultant crystal dislocation distributions. The work by Li et al. [51] involved the derivation of models for the chemical potentials of both the mobile guest species and immobile host material where the host material is in a stressed state. Beginning in the 1970's Larché and Cahn [52, 53, 54, 55] developed a framework for modeling diffusion induced stress. The model uses a theoretical embedded network within a material to track deformations. The diffusion process is driven by the chemical potential of the guest species within the host material. The derived chemical potential is a function of both the guest species concentration and hydrostatic stress. The work of Larché and Cahn now forms the basis for much of the current research being done to model the lithiation of silicon anodes [66, 57, 56].

Alternatives to the Larché and Cahn model for diffusion induced stress have also been proposed. For example in the paper by Wu [188] the argument is made that the Eshelby stress tensor should be used in the stress dependent chemical potential as opposed to the hydrostatic Cauchy stress. Cui et al. [63] extends the work of Wu to applications in modelling the stress-dependent chemical potential of lithium ion batteries. Other methods avoid the derivation of a chemical potential and instead derive an empirically based model for the diffusion process [20, 189, 190, 191].

As there have been multiple models proposed for the lithiation of a silicon anode, there have also been multiple methods used to simulate the lithiation/delithiation process. One method has been to reduce the dimensionality of the problem to a spherical geometry

for the anode [71, 64]. While the reduction of dimensionality can lead to insights into the lithiation/delithation process, without the higher computational costs of higher dimensional simulations, key features associated with anode geometries will be lost. For numerical simulations that cannot be reduced to one dimension, the finite element method (FEM) has been a popular choice. A large number of the published simulations involving the lithiation process of silicon have been done using the finite element method [192, 193, 20, 189, 194]. FEM is well understood and there exists a large number of different code bases, both commercial and open sourced, upon which a researcher can draw.

Key experimental observations of a silicon anode undergoing the lithiation process show the large deformations that it will experience. A variety of numerical methods have been used to model large deformations. Mesh-based methods such as FEM [195] and XFEM [196] have found success along with meshless methods such as SPH [197] and RPKM [198]. An alternative to these approaches that has been found to be successful in modeling large deformations is the material point method (MPM) [199]. MPM uses aspects of both FEM and particle methods to carry out computations.

MPM along with the multiple variants of the method are well described in the literature [16, 17, 89, 21, 172, 87]. For the the purpose of this paper the convected particle domain interpolation (CPDI) variant, as described by Sadeghirad and Brannon [86], will be used. The MPM algorithm as described in [89, 86] has been implemented in the Uintah compuational framework [200], and serves as the code base for this work. The *Uintah User Guide* [201] includes a further description of the MPM component, including additional feature that have been implemented therein. The primary contribution of this paper is the chemical-mechanical coupling, including the concentration diffusion solver working in concert with the momentum balance solver.

The chapter proceeds as follows. Section 7.1 will be a brief overview of MPM followed by a description of the method used to model diffusion within MPM. Results from verification tests of the diffusion method will also be presented. Section 7.2 will be a description of the two way coupling of the chemical process of lithium diffusion and mechanical process of momentum balance. Section 7.3 will present simulation results along with a discussion of the issues and simulation outcomes associated with time integration, constitutive model selection, boundary conditions, and parameter choices. Section 7.3.6 will look at the results

of bonding an anode to a fixed or deformable substrate. Section 7.3.7 will present the simulation results of a silicon anode undergoing a full lithiation/delithiation cycle. Lastly, Section 7.4 will draw conclusions and discuss further work.

7.1 The Material Point Method

As decribed in Chapters 3, 4, 5, and 6 the material point method was developed in the 1990s by Sulsky et al. [16] as numerical solution to the momentum balance equation,

$$\rho \boldsymbol{a} = \nabla \cdot \boldsymbol{\sigma} + \rho \boldsymbol{b}, \tag{7.1}$$

where ρ is density, $\boldsymbol{a}$ is acceleration, $\boldsymbol{\sigma}$ is stress, and $\boldsymbol{b}$ is the acceleration that comes as a result of an external force such as gravity. The basic premise of MPM is that the domain of an object is discretized into material points. Properties such as mass and momentum are assigned to each material point. The material points are Lagrangian in nature and are advected within the domain of the problem. The second component of MPM is a background Eulerian grid. Particle masses and momentums are mapped to the grid nodes using an appropriate choice of grid and particle basis functions. The grid is then used to solve the momentum balance equation. Particles values are updated by interpolating changes in the state from the grid nodes to the particles and then integrating forward in time. Variations in MPM come as a result of the choices made in grid and particle basis functions.

7.1.1 Modeling Diffusion in MPM

During the processes of lithiation and delithiation, lithium is transported by two processes. The first process is that of advection which arises as a result of the deformation of the host material. As the host material undergoes deformation the guest material embedded in the material is carried along with the deforming host material. The modeling of this process comes as result of lithium being assigned to a material point. The mass of the host material of each material point does not change. The advection of the guest material is accounted for naturally when material points are advected.

The second form of transport for lithium ions comes in the form of diffusion. For the purposes of this paper, the lithium concentration is normalized, $c = c_g/c_{max}$, where c_{max} is the molar concentration representative of a ratio of 3.75 lithium to 1 silicon and c_g is the current molar concentration. Diffusion is modeled in MPM using many of the same

numerical methods that are used in solving the momentum balance equations. The standard diffusion equation is

$$\frac{\partial c}{\partial t} = -\nabla \cdot \boldsymbol{J}, \tag{7.2}$$

where $\boldsymbol{J}$ is the normalized concentration flux. The equation for the normalized concentration flux is

$$\boldsymbol{J} = -D\nabla c \tag{7.3}$$

where D is the diffusivity constant.

A full description of using MPM to model diffusion is given in Chapter 6, but is applied here as follows. First, particle masses are mapped to the grid nodes,

$$m_i = \sum_p \varphi_{ip} m_p, \tag{7.4}$$

where m_i is the mass at the nodes, m_p is the mass at the particles, and φ_{ip} is the weighting function. For a general description of the use of the weighting function used in MPM, the reader is referred to Chapter 3 and [87], and for a specific description of the weighting functions used in the CPDI variant of MPM see [86]. The mapping of masses from particles to grid nodes is also a step found in the MPM solution to the momentum balance equation and thus grid values for m_i have already been computed.

The second step is to map the normalized particle concentration values to grid nodes. The mapping of concentration from particles to nodes is done in a similar fashion to mapping velocity values from particles to the grid and is done as follows:

$$c_i = \frac{\sum_p \varphi_{ip} m_p c_p}{m_i}, \tag{7.5}$$

where c_i is the normalized grid concentration value and c_p is the normalized particle concentration value. The third step is to calculate the flux values at the particles. The flux calculation is done by taking the gradient of the weighting function that interpolates normalized concentration values at the nodes to the particles,

$$\boldsymbol{J}_p = -D \sum_i \nabla \varphi_{ip} c_i. \tag{7.6}$$

The fourth step is to compute the divergence of the flux term. This is done in a manner similar to that used in computing the divergence of stress,

$$\frac{\partial c_i}{\partial t} = \frac{\sum_p (\boldsymbol{J}_p \cdot \nabla \varphi_{ip}) m_p}{m_i}. \tag{7.7}$$

The fifth step is to update normalized concentration values at the particles,

$$c_p^{n+1} = c_p^n + dt \sum_i \varphi_{ip} \frac{\partial c_i}{\partial t}. \tag{7.8}$$

This method of modeling concentration diffusion within MPM is similar to methods used previously in modeling heat diffusion [164], [166], and [165].

7.1.2 Verification of MPM Diffusion

A one-dimensional system with an analytical solution is used to test the method described above. The test problem is defined with the initial and boundary conditions on the spatial domain $0 \leq x \leq 1$ with a discontinuity existing between the initial condition and the boundary conditions and is written as follows:

$$\frac{\partial c}{\partial t} = -\frac{\partial}{\partial x} J, \tag{7.9}$$

$$c(x,0) = 0.0, \quad c(0,t) = 1.0, \quad c(1,t) = 0.0. \tag{7.10}$$

The flux term is defined as

$$J = -D \frac{\partial c}{\partial x}, \tag{7.11}$$

where D is the diffusivity constant. Combining (7.9) and (7.11) produces the following equation:

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2}. \tag{7.12}$$

The solution to Equation (7.12), subject to the initial and boundary conditions given in Equation (7.10), is well known [181],

$$c(x,t) = (1-x) - \frac{2}{\pi} \sum_{n=1}^{\infty} \frac{1}{n} e^{-Dt\pi^2 n^2} \sin(n\pi x). \tag{7.13}$$

For the purposes of determining the accuracy of the numerical method, the first 1000 terms of the Fourier expansion should be sufficient.

The numerical solution to Equation (7.12) is found using the MPM diffusion method described in Chapter 6. The problem set-up is such that there are two particle per cell. A set of seven simulations were run with grid spacings of 0.1, 0.05, 0.02, 0.01, 0.005, 0.002

and 0.001, with the data again being captured at simulation time 0.25. The error for the method was calculated taking the L2 norm of the relative error in concentration,

$$e_p = \begin{cases} (c_p - \hat{c}_p)/c_p & \text{if } c_p \neq 0 \\ 0 & \text{if } c_p = 0 \end{cases} \tag{7.14}$$

$$\text{error} = \frac{\sqrt{\sum_{p=1}^{N} e_p^2}}{N}, \tag{7.15}$$

where e_p is the relative error, c_p is the true solution, and $\hat{c}_p$ is the computed value at each particle p and N is the total number of particles. Fig. 7.1a shows a comparison of the actual results from the MPM diffusion solution with the true solution for a grid size of 0.02. Fig. 7.1b shows the convergence of the error as grid resolution increases. From the data it can be seen experimentally that the order of accuracy of MPM appears to be $O(h)$. For reference purposes the red line indicates a slope of 1.0, and the red error marker of Fig. 7.1b correlates with the results shown in Fig. 7.1a.

7.1.3 Advected Flux Boundary Conditions

In the calculations carried out here, a flux was applied to the surface of the object under investigation that was intended to mimic the flux induced by applying a voltage difference between the object and a far away cathode. Due to coupling between the concentration and mechanical state, increasing concentration results in deformation of the object. Hence, the location of the surface upon which the flux is prescribed is changing with time. For the simulations here, initial surface particles were identified, and the initial external area was computed and recorded. It was assumed here (and inspection of the results below validates this assumption), that while the surface moves and stretches, there is no change to what would be considered a surface particle, or to which face of the particle would be considered a surface. A current assumption that is probably less valid, and will be fixed in future work, is that the surface area associated with each particle remains constant. Inspection of the subsequent results show that this is not a valid assumption, but given the relatively uniform expansion of the object that we are currently interested in, this is similar to prescribing a slightly lower flux than intended. In future work, the particle area will be evolved according to the particle deformation gradient, which is already part of the particle data that is integrated in time.

A time varying flux is applied to the surface particles by computing the rate of change in concentration that this flux would be associated with, namely,

$$\left(\frac{\partial c_{pv}}{\partial t}\right)_{bc} = J(t)_{bc}\frac{A_p}{V_p}, \tag{7.16}$$

where $J(t)_{bc}$ is the user defined flux boundary condition, A_p is the surface of the particle upon which the boundary flux condition is being applied, V_p is the particle volume, and $(\partial c_{pv}/\partial t)_{bc}$ is the rate of change of concentration in the particle. Note that to avoid exceeding a normalized concentration of 1, a flux restriction factor is used, such that

$$\left(\frac{\partial c_{pv}}{\partial t}\right)'_{bc} = \left(\frac{\partial c_{pv}}{\partial t}\right)_{bc} \cdot (1 + .25\log(1 - c_p)). \tag{7.17}$$

Thus, once the concentration starts to saturate the substrate, the flux at that surface is gradually reduced. While the equation above is ad hoc, this reflects physically meaningful behavior. This rate of change is mapped to the grid and, in the case of boundary particles, functions as an additional source term in (7.7),

$$\frac{\partial c_i}{\partial t} = \frac{\sum_p (\boldsymbol{J}_p \cdot \nabla\varphi_{ip})m_p}{m_i} + \frac{\sum_p (\partial c_{pv}/\partial t)'_{bc}\,\varphi_{ip}m_p}{m_i}. \tag{7.18}$$

7.2 Chemical/Mechanical Coupling in MPM

7.2.1 Coupling Stress to Diffusion

Multiple models have been formulated that couple the affects of stress with the diffusion process [20, 64, 202, 203, 189, 190, 191]. The existing models can be broken up into two basic groups. The first group, [64, 202, 203], follows the framework that was developed by Larché and Cahn [52, 53, 54, 55] and computes the concentration flux based on a formulation of the chemical potential of lithium within the host silicon anode. The Larché and Cahn chemical potential is defined as

$$\mu_{Li} = \mu^0_{Li} + RT\ln\left(\gamma\frac{c}{1-c}\right) + \kappa\frac{d\epsilon^c_{vol}}{dc}p - \frac{\kappa}{2}B_{ijkl}\sigma_{ij}\sigma_{kl}, \tag{7.19}$$

where μ^0_{Li} is the reference chemical potential, R is the ideal gas constant T is temperature, γ is the activation coefficient, κ is the molar volume of silicon, $\epsilon^c_{vol} = \epsilon^c_{ii}/3$ is the volumetric portion of the stress-free strain due to the insertion of lithium, $p = -\sigma_{ii}/3$, and

$B_{ijkl} = dS_{ijkl}/dc$ is the rate of change of the elastic compliance tensor, S_{ijkl}, with respect to concentration. The flux based on the chemical potential is defined as

$$\boldsymbol{J} = -D\frac{c}{RT}\nabla\mu. \tag{7.20}$$

If the constitutive model has compliance tensor that is not dependent upon concentration, as is the case for the model discussed below, then B_{ijkl} reduces to zero and in cases where the relationship between the stress-free strain, ϵ^c_{ij}, and concentration is linear then $d\epsilon^c_{vol}/dc$ is a constant. Assuming that the compliance tensor is non-concentration dependent and the linear concentration relationship, then (7.20) can be written as a combination of the concentration and pressure gradients,

$$\boldsymbol{J} = -\frac{D\gamma}{(1-c)}\nabla c - \frac{D\kappa c}{RT}\frac{d\epsilon^c_{vol}}{dc}\nabla p. \tag{7.21}$$

The second group, [20, 189, 190, 191], uses an empirical based approach where a model is formulated that takes as input concentration and in some cases stress and then the user adjusts a set of parameters to produce the desired behavior. In the empirical-based approach, flux is defined as follows:

$$\boldsymbol{J} = -D(c,\sigma)\nabla c. \tag{7.22}$$

While both methods can be implemented within the MPM diffusion framework described above, for the purposes of this paper the approach of the second group will be used. The selection of the empirical-based approach is based upon the observation that the greatest changes in pressure occur in regions where lithium is diffusing into the nonlithiated amorphous silicon, in the areas behind this region the changes in pressure are minimal and the contribution of the pressure gradient to the concentration flux is small. The empirical-based approach is able to approximate the behavior of the lithiation of nonlithiated amorphous silicon and provides a good first approximation of the diffusion process.

The normalized concentration and pressure-dependent function for diffusivity that will be used for this paper is defined as,

$$D(c,p) = \begin{cases} D_0 e^{\alpha c} & \text{if } p \leq 0, \\ D_0 e^{\alpha c - \beta p} & \text{if } 0 < p < p_{max}, \\ D_0 e^{\alpha c - \beta p_{max}} & \text{if } p \geq p_{max}, \end{cases} \tag{7.23}$$

where D_0 is the initial diffusivity, α and β are tuning parameters, and p_{max} is the capped value for pressure. It is through the diffusivity function that stress is coupled to the diffusion

process. By way of comparison, Berla et al. [191] and Wang et al. [190] use a cubic polynomial to compute diffusivity as solely function of concentration. The diffusivity function is defined as

$$D(c) = D_0(1-c)^3 + D_{max}c^3, \tag{7.24}$$

where D_{max} is the diffusivity at full concentration. Using the following parameters $D_0 = 10^{-17}m^2/s$, $D_{max} = 10^{-15}m^2/s$, $\alpha = 6.0$, $\beta = .5$ and $p_{max} = 3$ GPa for (7.23) and (7.24) diffusivity curves were evaluated over the normalized concentration range $[0, 1]$. Fig. 7.2a shows the results for (7.23) and (7.24) with (7.23) being evaluated at 0 GPa and 3 GPa. Fig. 7.2b shows the surface plot of (7.23) over the normalized concentration range of $[0, 1]$ and the pressure range from $[0, 3]$ GPa with $\alpha = 6.0$, $\beta = .5$ and $p_{max} = 3$ GPa.

7.2.2 Coupling Concentration to Stress

The modeling of the silicon anode material response to concentration diffusion is done using a hypoelastic-plastic constitutive model. The hypoelastic-plastic model allows for an additive decomposition of the rate of strain tensor into elastic, plastic, and stress-free volumetric components,

$$\mathbf{d} = \mathbf{d}^e + \mathbf{d}^p + \mathbf{d}^c, \tag{7.25}$$

where $\mathbf{d}^e$ is the elastic rate of strain, $\mathbf{d}^p$ is the plastic rate of strain, and $\mathbf{d}^c$ is the stress-free volumetric rate of strain due to the insertion of lithium into the silicon material.

Objectivity is maintained in the stress calculation by rotating stress and the rate of strain back to its unrotated state where the stress update calculation is then performed, and then subsequently this updated stress is rotated back to its previous state. The translation between the two states is done using the rotation tensor obtained from the polar decomposition of the deformation gradient,

$$\mathbf{F} = \mathbf{R}\mathbf{U}, \tag{7.26}$$

where $\mathbf{F}$ is the deformation gradient, $\mathbf{R}$ is the rotation tensor, and $\mathbf{U}$ is the right stretch tensor. The transformation of stress from its rotated to unrotated state is performed as follows:

$$\bar{\boldsymbol{\sigma}} = \mathbf{R}^T\boldsymbol{\sigma}\mathbf{R}, \tag{7.27}$$

and the transformation for the rate of strain is done similarly,

$$\bar{\boldsymbol{d}} = \mathbf{R}^T \boldsymbol{d} \mathbf{R}, \tag{7.28}$$

where the rate of strain is the symmetric part of the velocity gradient, $\boldsymbol{l}$,

$$\boldsymbol{d} = \frac{1}{2}(\boldsymbol{l} + \boldsymbol{l}^T). \tag{7.29}$$

The unrotated symmetric part of the velocity gradient is chosen as a rate of strain because of the relatively low computational cost relative to other possible choices for rates of strain and under conditions where the rotation of the principle referential directions is small the unrotated symmetric part of the velocity gradient is a good approximation for Hencky strain rate [204].

In the unrotated state the stress/strain relationship is defined as

$$\bar{\boldsymbol{\sigma}} = 2G\boldsymbol{\epsilon}^e_{dev} + 3K\boldsymbol{\epsilon}^e_{vol}, \tag{7.30}$$

where G is the shear modulus, K is the bulk modulus, $\boldsymbol{\epsilon}^e_{dev}$ is the deviatoric elastic strain and, $\boldsymbol{\epsilon}^e_{vol}$ is the volumetric elastic strain. The derivation of the material stress rate then easily follows,

$$\dot{\boldsymbol{\sigma}} = 2G\bar{\boldsymbol{d}}^e_{dev} + 3K\bar{\boldsymbol{d}}^e_{vol}, \tag{7.31}$$

where $\bar{\boldsymbol{d}}^e_{dev}$ is the deviatoric elastic strain rate and $\bar{\boldsymbol{d}}^e_{vol}$ is the volumetric elastic strain.

Due to the amorphous nature of the silicon in our investigation, the concentration dependent stress-free portion of strain may be regarded as isotropic and follows the same model used in [54]. This is analogous to the expressions used for isotropic thermal expansion [58],

$$\boldsymbol{\epsilon}^c = \eta(c - c_0)\boldsymbol{I}, \tag{7.32}$$

where η is the volume expansion coefficient and $\boldsymbol{I}$ is the identity tensor. The material rate of stress-free volumetric strain is then

$$\bar{\boldsymbol{d}}^c = \eta \frac{\partial c}{\partial t} \boldsymbol{I}. \tag{7.33}$$

7.2.3 Perfect Plasticity and Linear Strain Hardening

Two different plasticity models will be used to determine the rate of plastic strain, $\bar{\boldsymbol{d}}^p$. The first is perfect plasticity with the yield surfaces defined by

$$\sqrt{\frac{3}{2}\bar{\boldsymbol{\sigma}}_{dev} : \bar{\boldsymbol{\sigma}}_{dev}} - \sigma_Y = 0, \tag{7.34}$$

and the second is linear isotropic strain hardening with the yield surface defined by

$$\sqrt{\frac{3}{2}\bar{\boldsymbol{\sigma}}_{dev} : \bar{\boldsymbol{\sigma}}_{dev}} - (\sigma_Y + K\epsilon^p_{equiv}) = 0, \tag{7.35}$$

where σ_Y is the initial yield stress, $\bar{\boldsymbol{\sigma}}_{dev} = \bar{\boldsymbol{\sigma}} - 1/3tr(\bar{\boldsymbol{\sigma}})$ is the deviatoric stress, K is the plastic modulus, and ϵ^p_{equiv} is the equivalent plastic strain. A description of the method for computing $\bar{\boldsymbol{d}}^p$ is beyond the scope of this paper, but the interested reader is referred to [205] for a complete description of the algorithm. Both the perfectly plastic and linear isotropic strain hardening models are implemented within the Uintah MPM component [200] and are used for the simulations presented in this paper.

7.2.4 Stress Update

Using (7.29), (7.33), and the rate of plastic strain, $\bar{\boldsymbol{d}}^p$, the rate of elastic strain can then be calculated,

$$\bar{\boldsymbol{d}}^e = \bar{\boldsymbol{d}} - \bar{\boldsymbol{d}}^c - \bar{\boldsymbol{d}}^p. \tag{7.36}$$

The rate of elastic strain is then decomposed into its deviatoric and volumetric component,

$$\bar{\boldsymbol{d}}^e_{vol} = \frac{1}{3}tr(\bar{\boldsymbol{d}}^e)\boldsymbol{I}, \qquad \bar{\boldsymbol{d}}^e_{dev} = \bar{\boldsymbol{d}}^e - \bar{\boldsymbol{d}}^e_{vol}. \tag{7.37}$$

Using the deviatoric and volumetric elastic strain rates the updated stress is then computed,

$$\bar{\boldsymbol{\sigma}}^{n+1} = \bar{\boldsymbol{\sigma}}^n + (2G\bar{\boldsymbol{d}}^e_{dev} + 3K\bar{\boldsymbol{d}}^e_{vol})dt, \tag{7.38}$$

where $\bar{\boldsymbol{\sigma}}^n$ is the current stress, $\bar{\boldsymbol{\sigma}}^{n+1}$ is the updated stress, and dt is the timestep size. The last step in the stress update calculation is to rotate the updated stress to it rotated state,

$$\boldsymbol{\sigma}^{n+1} = \boldsymbol{R}\bar{\boldsymbol{\sigma}}^{n+1}\boldsymbol{R}^T. \tag{7.39}$$

7.3 Numerical Solution of the Model Problem

As was mentioned in the introduction, there have been a multitude of numerical simulations performed to study the affects of lithiation on silicon anodes. In these studies different anode geometries have been used. Three common choices have been a sphere, a nanowire, and a pillar. For the purposes of this paper the pillar geometry will be used, and while the pillar geometry lends itself to a 2D axisymmetric numerical solution, for the purposes of this paper full 3D simulations will be performed. Two different choices in pillar sizes will be used through out this section. The smaller pillar has a diameter of $0.1\,\mu m$ and

a height of $0.1\,\mu m$. The larger pillar has a diameter of $0.5\,\mu m$ and a height of $0.05\,\mu m$. Multiple simulations were run using the smaller pillar size to explore choices in constitutive model, parameter selection, and boundary conditions. With the results from the simulations performed using the smaller diameter pillar, decisions were then made as to constitutive model choice, parameter selection, and boundary conditions and then applied to simulations using the larger diameter pillar. The use of a substrate material was then explored using the larger diameter pillar.

7.3.1 Time Scales of Governing Equations

The governing equations for momentum balance and concentration diffusion are represented by (7.1) and (7.2). For the purposes of this paper the external forces acting on the nanopillar will be neglected, $\boldsymbol{b} = 0$, and (7.1) can be reduced to

$$\rho \boldsymbol{a} = \nabla \cdot \boldsymbol{\sigma}. \tag{7.40}$$

The time scales of these two physical processes differ greatly in magnitude. As was discussed in Section 7.2.1 the range of values for diffusivity of lithium within amorphous silicon range between $10^{-17} m^2/s$ and $10^{-15} m^2/s$. On the other hand, the material response to changes in deformation happens at a much smaller time scale. The speed of sound within a given material is defined by the following equation:

$$c_s = \sqrt{\frac{K + 4/3G}{\rho}}, \tag{7.41}$$

where K is the bulk modulus, G is the shear modulus, and ρ is density. For amorphous silicon reasonable values for density, bulk modulus, and shear modulus are $2.33 \times 10^3\, kg/m^3$, $67\, GPa$, and $31\, GPa$, respectively. Based on these values the sound speed for amorphous silicon is $6.8 \times 10^3 m/s$. A simple comparison of diffusivity and sound speed shows the large differences in time scale by which each of the physical processes occur. Because of this difference, the momentum balance equation reaches a quasi-steady state,

$$\nabla \cdot \boldsymbol{\sigma} = 0, \tag{7.42}$$

within the time scale that diffusion occurs [57, 206, 207].

7.3.2 Time Integration and Time Step Selection

At the end of each iteration of the mpm algorithm a new time step, dt, is calculated. The size of the time step is limited by the numerical solutions for the diffusion equation

and the momentum balance equation. The time step for the momentum balance equation is computed in the following manner based on stability,

$$dt_{mb} = \min\left(\frac{dx_{cell}}{c_s + v_x}, \frac{dy_{cell}}{c_s + v_y}, \frac{dz_{cell}}{c_s + v_z}\right), \tag{7.43}$$

where dx_{cell}, dy_{cell}, and dz_{cell} are the cell dimension and v_x, v_y, and v_z are the components of the particle velocity vector in the x, y, and z directions respectively.

The time step size based on the numerical solution for the diffusion equation is calculated as follows:

$$dt_{diff} = \min\left(\frac{dx_{cell}^2}{2D}, \frac{dy_{cell}^2}{2D}, \frac{dz_{cell}^2}{2D}\right), \tag{7.44}$$

where D is the diffusivity coefficient. The time step that is then used for the overall simulation is the smallest of the two time steps,

$$dt = \min(dt_{mb}, dt_{diff}). \tag{7.45}$$

7.3.2.1 Time Stepping Issues

The time step criterion stated above is used to maintain a stable numerical solution, but if the time step needed for a stable solution becomes too small, then computation time needed for a viable solution will become intractable. To illustrate this point, a stable time step is computed using (7.43). The sound speed of $6.8 \times 10^3 m/s$ computed in Section 7.3.1 and a grid cell dimension of $dx_{cell} = 10^{-8}\, m$ is used in the calculation. For simplicity the particle velocity is neglected in the calculation because it is small in comparison to the computed sound speed. The resulting time step is

$$dt_{mb} = \frac{dx_{cell}}{c_s} = 1.47 \times 10^{-12} s. \tag{7.46}$$

Using the above computed time step size, 10^{11} time step integrations would be needed to reach 6.8s of simulation time. The number of time steps for this calculation would be intractable. By way of comparison, for a given diffusivity of $D = 10^{-15}\, m^2/s$ the calculated time step size is

$$dt_{diff} = \frac{dx_{cell}^2}{2D} = 0.05 s, \tag{7.47}$$

which is a tractable time step size.

From the above example it can be seen that for a simulation to run in a reasonable time a stable means of increasing dt_{mb} to a tractable time step size needs to be found. Ideally, the time step size would be limited by the diffusion calculations and not momentum balance.

In order to increase the limiting time step size needed for the numerical solution to the momentum balance equation, the density of silicon is artificially raised to a value such that the limiting time step criterion is dependent upon the numerical solution of the diffusion equation and not the solution to the momentum balance equation. The argument for the validity of artificially raising the density stems from the observations made in Section 7.3.1 that within the time scale of chemical diffusion the momentum balance equation reaches a quasi-static state. Using constitutive relationship found in (7.30) and (7.40) can be rewritten as

$$\rho \boldsymbol{a} = 2G\nabla \cdot \boldsymbol{\epsilon}^e_{dev} + 3K\nabla \cdot \boldsymbol{\epsilon}^e_{vol}, \tag{7.48}$$

and dividing through by density produces the equation for acceleration,

$$\boldsymbol{a} = \frac{2G}{\rho}\nabla \cdot \boldsymbol{\epsilon}^e_{dev} + \frac{3K}{\rho}\nabla \cdot \boldsymbol{\epsilon}^e_{vol}. \tag{7.49}$$

An increase in density decreases the speed by which elastic waves travel through the material, but the elastic strain remains the same.

For the purposes of demonstration, two sets of numerical simulations where run. In the first set of numerical experiments two simulations were run. In the first simulation the density for amorphous silicon is used, $2.33 \times 10^3\, kg/m^3$. The bulk and shear moduli are $67\, GPa$ and $31\, GPa$, respectively. The linear isotropic stain hardening constitutive model is used with a yield stress of $1.4\, GPa$ and a plastic modulus of $1.15\, GPa$. The initial diffusivity is artificially increased to $D_0 = 10^{-10} m^2/s$. In the second simulation the density is increased to $2.33 \times 10^8\, kg/m^3$ with all other parameters remaining the same. Fig. 7.3 shows a side-by-side comparison of the two simulations at $6.0 \times 10^{-9}\, s$, $26.0 \times 10^{-9}\, s$, and $48.0 \times 10^{-9}\, s$. From Fig. 7.3 it can be seen that even with a five order increase in density, qualitatively the variations between the simulations are small.

In the second set of simulations two simulations were run. In the first, an initial diffusivity is set to $D_0 = 10^{-17} m^2/s$, which falls within the range of published values [137]. The density is artificially raised to $2.33 \times 10^{18}\, kg/m^3$. The bulk and shear moduli are $67\, GPa$ and $31\, GPa$, respectively. The linear isotropic stain hardening constitutive model is used with a yield stress of $1.4\, GPa$ and a plastic modulus of $1.15\, GPa$. The second test was run using an initial density of $2.33 \times 10^{24}\, kg/m^3$. Fig. 7.4 shows a side-by-side comparison of the two simulations at three different time steps - $4.0\, s$, $16.0\, s$, and $28.0\, s$. From Fig. 7.4

it can be seen qualitatively that there is very little variation between the two simulations despite the five orders of magnitude difference in density.

7.3.3 Comparison of Constitutive Models

A proper constitutive model needs to be selected in order to achieve the desired behavior that matches that of a silicon anode during lithiation. Simulations were run using three constitutive models, linear elastic, perfectly plastic, and linear isotropic strain hardening. The bulk and shear moduli for all three models are $67\,GPa$ and $31\,GPa$, respectively. For the perfectly plastic and linear isotropic strain models the yield stress is $1.4\,GPa$ and for the linear isotropic strain hardening models the plastic modulus is $1.15\,GPa$. The volume expansion coefficient used for this set of simulations and all subsequent simulations is $\eta = .56$. The flux boundary conditions and diffusivity parameters are the same for all three simulations.

Fig. 7.5 shows images of the pillars at mid-saturation and full saturation. From the image it can be seen that at full saturation there is a noticeable difference between the linear elastic and the two plasticity models. The largest noticeable difference is along the base of the pillars where the largest concentration of plastic deformation occurs. In images (a-c) of Fig. 7.5 it can be seen that plastic deformation is already occurring along the base of the pillars producing different deformation patterns compared against the linear elastic pillar.

While there are noticeable differences between the linear elastic model and the two plasticity models, there are also qualitative differences between the two plasticity models.

7.3.4 Flux Boundary Conditions

Both the perfectly plastic and the linear strain hardening constitutive models are dependent upon the prior history of the material. Because of this history dependent behavior the choice of flux boundary conditions affects the shape of the pillars. To demonstrate the affect that the choice of the flux magnitude has on deformation behavior, two simulations were run. The diffusion parameters for both simulations are $\alpha = 6$, $\beta = 0.5$, and $p_{max} = 3\,GPa$. The linear isotropic strain hardening constitutive model is used with the bulk and shear modulus values of $67\,GPa$ and $31\,GPa$, respectively. The yield stress is $1.4\,GPa$ and the plasticity modulus is $1.15\,GPa$. The magnitude of the flux boundary condition are 10^{-9}

and 10^{-8}. Fig. 7.6 shows two sequence of images where the top row shows results from the simulation run at the lower flux value and the lower row shows the sequence of images for the simulation run at the higher flux value.

Both sets of simulations illustrate the sharp transition between high normalized concentrations and low normalized concentrations as the lithium diffuses through the anode. The noticeable difference between the simulations is in the normalized concentration levels that are behind the sharp transition in normalized concentration. As would be expected the simulation with the larger influx of normalized concentration will have larger normalized concentration values behind the high/low transition. From the experimental results presented by Wang et al., the initial concentration values behind the phase transition have a normalized concentration of ~ 0.67. For the lower flux boundary condition the normalized concentration values fall in the range of 0.5 to 0.6 while the high flux boundary condition have normalized concentration values greater than 0.7.

As can be seen from Fig. 7.6, differences in flux boundary condition values not only affect the level of concentration values behind the sharp phase transition, they also changes patterns in deformation. In the pillar with the larger flux boundary condition values, the bulging near the base is more pronounced in comparison to the pillar with the small flux values. Fig. 7.7 shows the state of the nanopillar as it reaches its fully lithiated state where the differences in flux boundary condition values produces noticeable differences in the deformation of the nanopillar.

7.3.5 Diffusivity

Using density functional theory calculations a range of diffusivity values have been determined for diffusion of lithiation in amorphous silicon of $10^{-18}\mathrm{m}^2\mathrm{s}^{-1}$ to $10^{-14}\mathrm{m}^2\mathrm{s}^{-1}$ [137]. The diffusivity model presented in this paper produces diffusivity values that fall within the proposed range. The issue that has been discovered is that while the calculated values for diffusivity may fall within the computed range of values for a chosen set of tuning parameters, the deformation behavior can vary based on what tuning parameters are chosen.

Previous works [20, 189, 190, 191] have chosen models that compute diffusivity solely as a function of concentration. However, it is clear that the mobility of lithium ions within amorphous silicon has to be a function of both lithium concentrations and pressure. From

numerical experiments it can be seen that the inclusion of pressure in the diffusivity calculation does affect the deformation of the pillar during lithiation.

Two simulations were run to show the difference that can occur when pressure is accounted for in the diffusivity calculation. The first simulation used the following parameters, $\alpha = 6.0$ and $\beta = 0.0$, and the second simulation used $\alpha = 6.0$ and $\beta = 1.0$. By setting β equal to zero, pressure is neglected from the calculation. Both simulations used the linear isotropic strain hardening constitutive model with bulk and shear modulus values of $67\,GPa$ and $31\,GPa$, respectively. The yield stress is $1.4\,GPa$ and the plasticity modulus is $1.15\,GPa$. The magnitude of the flux boundary condition is 10^{-9}. Fig. 7.8 shows a comparison of the two simulation. Fig. 7.8a and Fig. 7.8b show the results of the simulation during the lithiation process, from these figures it can be seen that by changing the pressure tuning parameter the difference in deformations, especially near the base of the pillar, are noticeable. Fig. 7.8c and Fig. 7.8d show the results of the simulations as the pillars reach full saturation. In the fully lithiated state the differences between the two simulations are less noticeable. Both sets of parameters produce the sharp phase transition between high and low normalized concentration levels as lithium diffuses through the pillar. The difference being in the deformation behavior during the lithiation process.

7.3.6 Use of Material Substrate

Up to this point all simulations have been made with the assumption that the base of the pillar is fixed. Experimental results show that this is not the case. Wang et al. [20] have shown through experiments that the base of the electrode can expand in the radial direction by $\sim 20\%$. It is unknown if the deformation comes as a result of the sliding between the anode and the substrate material or deformation in the substrate.

Two simulations were run to test the effects that a deformable material substrate will have on pillar deformation. For the purposes of this paper the pillar is assumed to be affixed to the substrate and the substrate material is allowed to deform. The bulk and shear moduli of the substrate material are $180.4\,GPa$ and $76\,GPa$, respectively. The density of the material is the same as that of the pillar. The linear isotropic strain hardening plasticity model is used with a yield stress of $.7\,GPa$ and a plasticity modulus of $1.15\,GPa$. Fig. 7.9 shows a comparison between the pillar attached to a nondeformable substrate and a pillar

attached to a deformable substrate.

From Fig. 7.9 it can be seen that there are noticeable differences between the two simulations. In the nondeformable substrate image it can be seen that large deformations occur in the region where pillar attaches to the base, by way of comparison in the deformable substrate image it can be seen that the deformation due to the pillar swelling is shared between both the pillar and the substrate. The deformable substrate reduces the accute deformations that occur in the pillar as highlighted by circled regions of Fig. 7.9a and Fig. 7.9b.

The use of a deformable substrate also affects the shape of the pillar as it reaches full lithiation. Fig. 7.9c and Fig. 7.9d show the pillars in the lithiated state. A visual comparison of the images shows that the shapes of the two pillars are different.

7.3.7 The Full Lithiation and Delithiation Cycle of the Silicon Anode

Using what was learned from the previous sections, a full lithiation/delithiation cycle was performed on the large pillar. The linear isotropic strain hardening constitutive model was used for the pillar with the bulk and shear moduli being $67.0\,GPa$ and $31.0\,GPa$, respectively, the yield stress was $1.4\,GPa$ and the plasticity modulus was $1.15\,GPa$. The diffusivity parameters used during the lithiation phase were $D_0 = 10^{-17}\,m^2/s$, $\alpha = 6.0$, $\beta = 0.5$ and $p_{max} = 3\,GPa$. During the delithiation phase diffusivity was fixed at the final diffusivity value during the lithation phase. The linear isotropic strain hardening constitutive model was used for the substrate material with the bulk and shear moduli being $180.4\,GPa$ and $76.0\,GPa$, respectively, the yield stress was $0.7\,GPa$ and the plasticity modulus was $1.15\,GPa$. There was no diffusion of lithium into the substrate material. Fig. 7.10 shows the results from the simulation of the silicon anode undergoing the full litiation/delithiation cycle.

As was discussed in the introduction to this paper and by Wang et. al. [20], one key observable trait of the lithiation of silicon is the sharp lithiation front that occurs during the lithiation process. As can be seen from Fig. 7.10 the sharp lithiation front is present during the lithiation phase of the simulation. Another key observable trait found in both the simulation and in experiments [20] is the occurrence of the much more shallow concentration gradients in the region behind the region of the sharp lithation front.

For the delithiation phase of the simulation, the diffusivity value was fixed at last value obtained during the lithiation phase. With the use of the fixed larger diffusivity value the rate of diffusion throughout the entire pillar appears to be more uniform. A comparison of the first and last images of the sequence found in Fig. 7.10 shows the amount of plastic deformation that occurs in both the nanopillar and the substrate during one full lithiation/delithaition cycle.

The simulation was run on nine cores of a single compute node. The cpu for the compute node is an Intel Xeon E5-2667 with six physical cores extended to 12 cores with hyper-threading enabled. The cpu clock speed is 2.9GHz. The total number of time steps needed to complete the simulation was 27472 and the estimated wall-clock time was 48.57 hours.

7.4 Conclusion

Silicon has the ability to absorb up to 3.75 lithium ions for each silicon atom. The ability to absorb large amounts of lithium makes silicon an excellent candidate material to increase the efficiency of electrochemical storage devices. The drawback for silicon is the accompanying large volume change that comes as a result of absorbing large amounts of lithium. The ability to simulate accurately the large physical deformations that occur during the lithation/delithation of silicon will to help to better understand how it can best be used as an anode material.

In this chapter it has been demonstrated that the full lithiation and delitiation of a silicon anode can be simulated using the material point method. A description of the method for used to model diffusion within MPM was given and validation tests were performed. This paper presented a coupled chemical-mechanical model that couples stress to the diffusion process and concentration to the constitutive model for stress. The coupled chemical-mechanical model was implemented in the Uintah Computational Framework's MPM component. Multiple numerical simulations were run to explore the different possible choices that could be made pertaining to constitutive model selection, boundary conditions, and parameter selection. Using what was learned from the different numerical simulations, a full lithiation/delitiation cycle simulation was performed.

The implementation of the coupled chemical-mechanical model into the Uintah Computational Framework allows for Uintah MPM to be used as a tool to for future work in

examining different anode geometries, material substrates, and flux rates.

In future work methods for modeling fracture need to be introduced into the Uintah framework in order to better simulate the full physical effects of lithiation and delithation on a silicon anode. In this current work the effects of an applied voltage have been approximated using the flux boundary condition. Future work needs to be done to implement the full electrochemistry in to the numerical models.

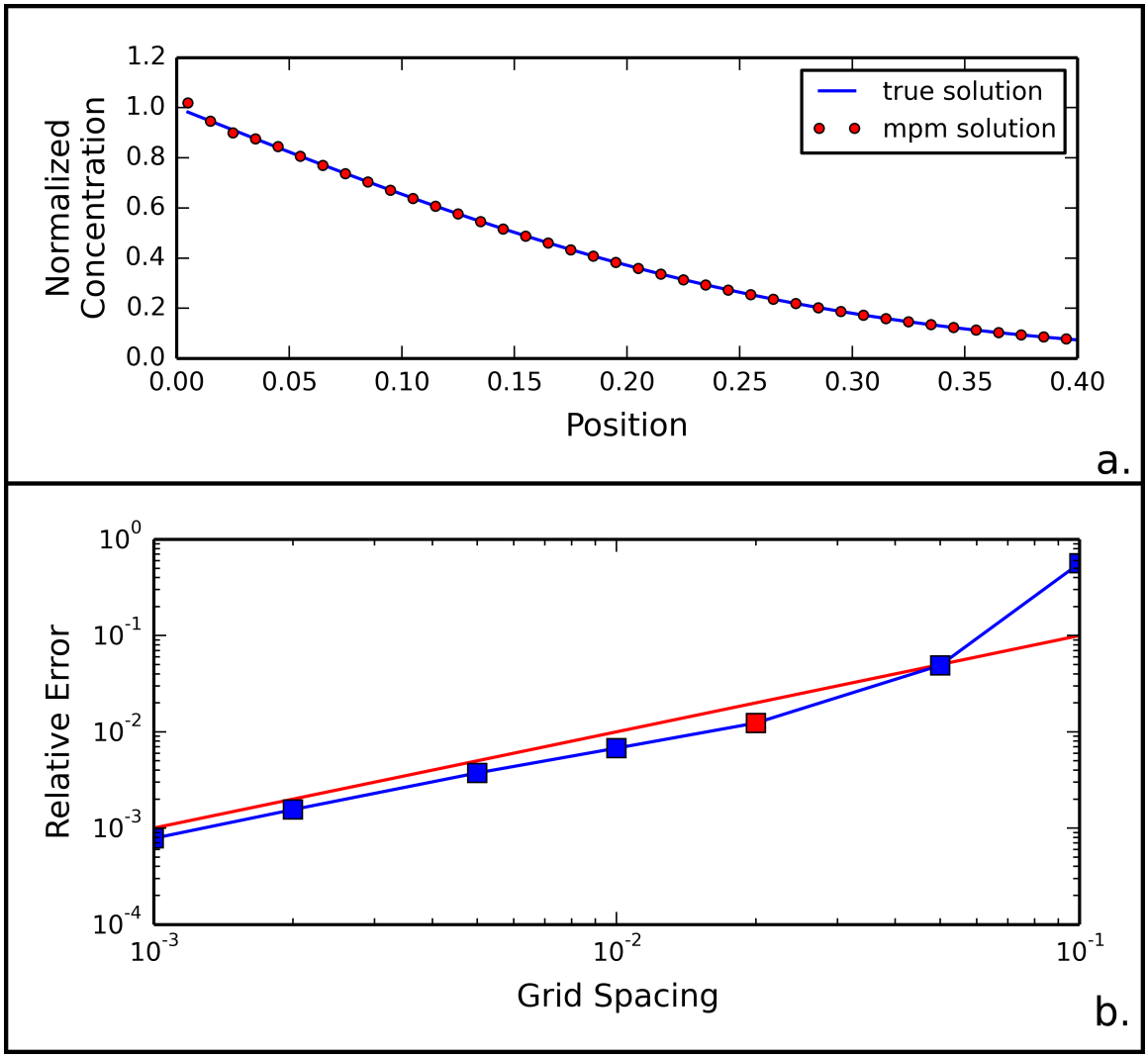

Fig. 7.1. Snapshot of the concentration profile and relative error: (a) shows the computed results from the MPM diffusion method compared against the true solution for grid size 0.02 and at time .25, and (b) is the error convergence plot for the MPM diffusion method. The red marker correlates with the results shown in (a) and the red line is for reference purposes and is a slope of 1.0.

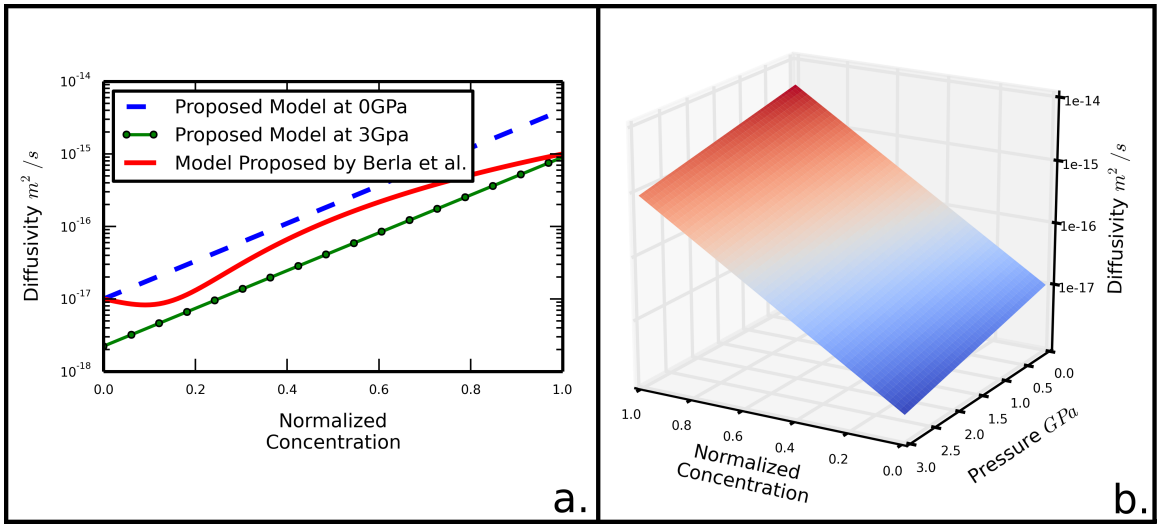

Fig. 7.2. Plots of diffusivity as a function of concentration and pressure: (a) comparison of the diffusivity function evaluated at 0 GPa and 3 GPa compared against the diffusivity function proposed by Berla et al. (b) The surface plot of the diffusivity function over the range of [0,1] for concentration and [0,3] GPa for pressure.

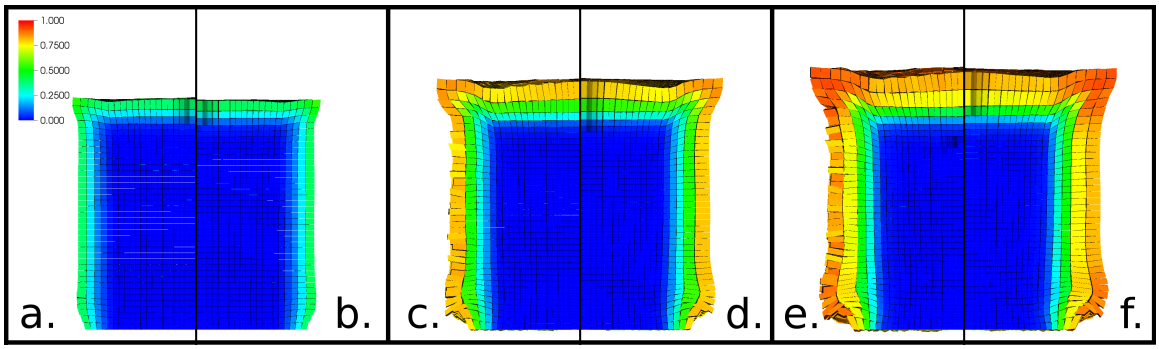

Fig. 7.3. Comparison of the use of different densities (a, c, e) density of $2.33 \times 10^3\, kg/m^3$ and (b, c, f) density of $2.33 \times 10^8\, kg/m^3$.

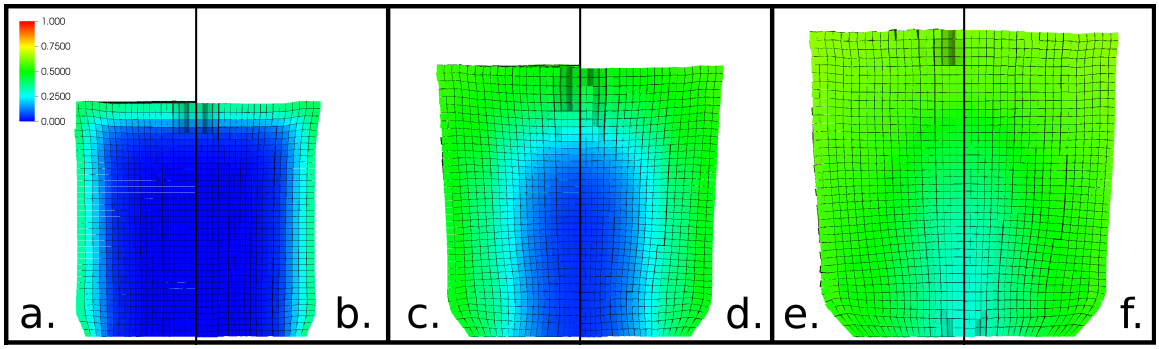

Fig. 7.4. Comparison of the use of different densities (a, c, e) density of $2.33 \times 10^{18}\, kg/m^3$ and (b, c, f) density of $2.33 \times 10^{24}\, kg/m^3$.

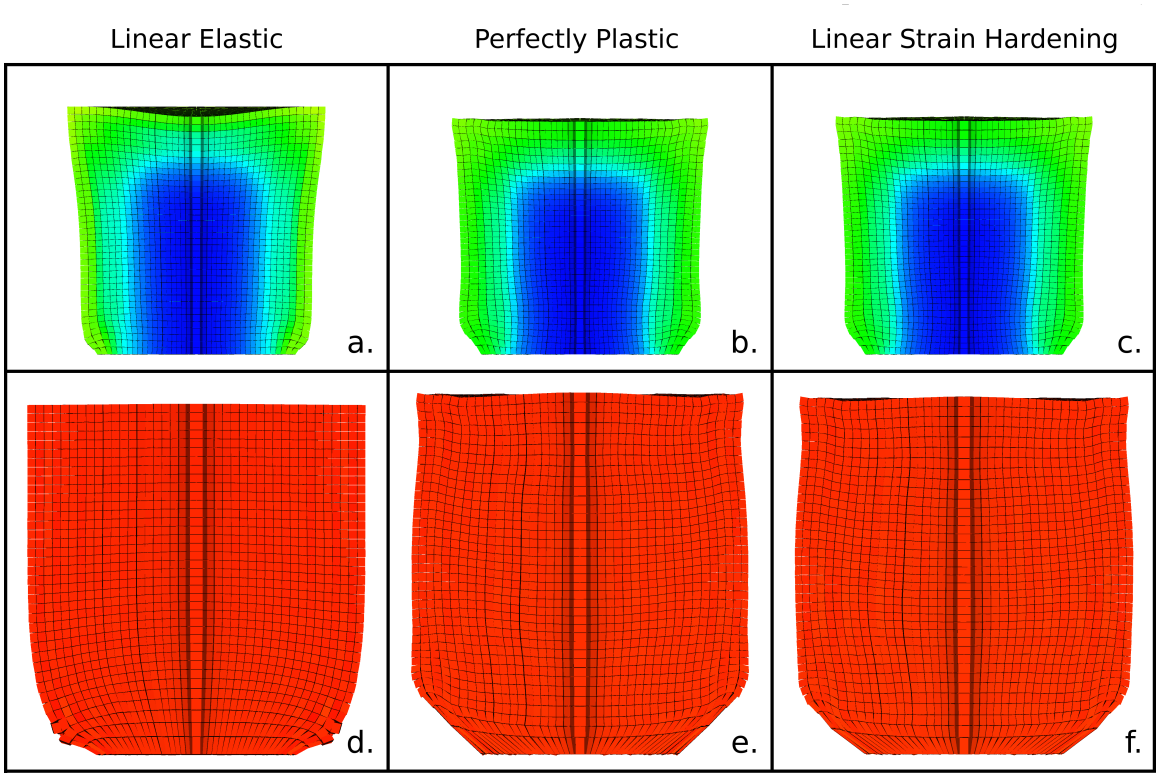

Fig. 7.5. A comparison of different constitutive models (a-c) are images of pillars mid-saturation, (d-f) are images of pillars at full-saturation. Images (a, d) are linear elastic, (b, e) are perfect plasticity and (c, f) linear strain Hardening.

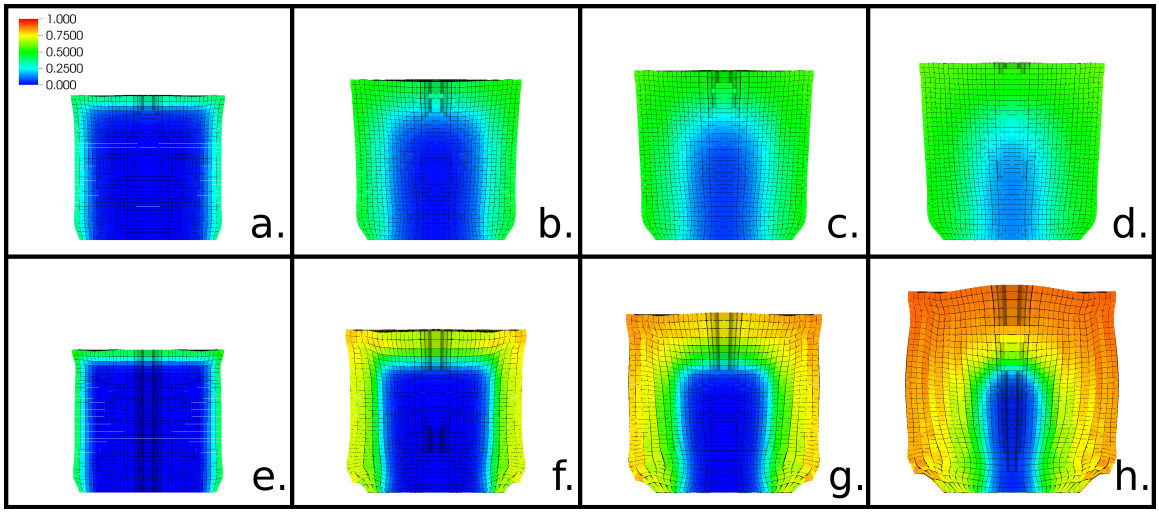

Fig. 7.6. Results of the use of different flux boundary condition values (a-d) is a series of images showing the lithiation process using a 10^{-9} flux boundary condition and the linear isotropic strain hardening model. (e-h) is a series of images during the lithiation process using a 10^{-8} flux boundary condition and the linear isotropic strain hardening model.

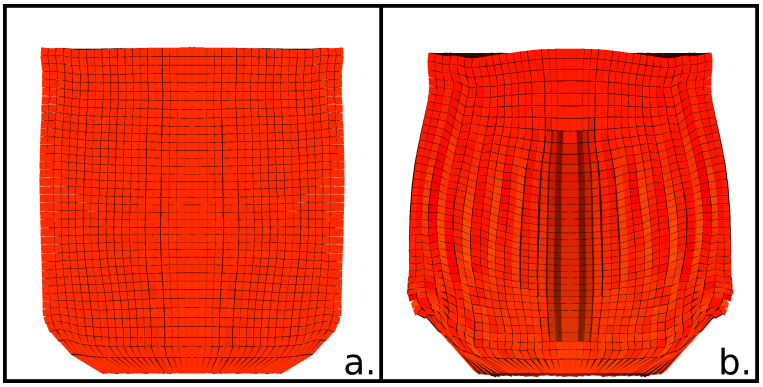

Fig. 7.7. The end results based on the boundary conditions (a) fully lithiated state of nanopillar using 10^{-9} flux boundary condition and the linear isotropic strain hardening model. (b) fully lithiated state of nanopillar using 10^{-8} flux boundary condition and the linear isotropic strain hardening model.

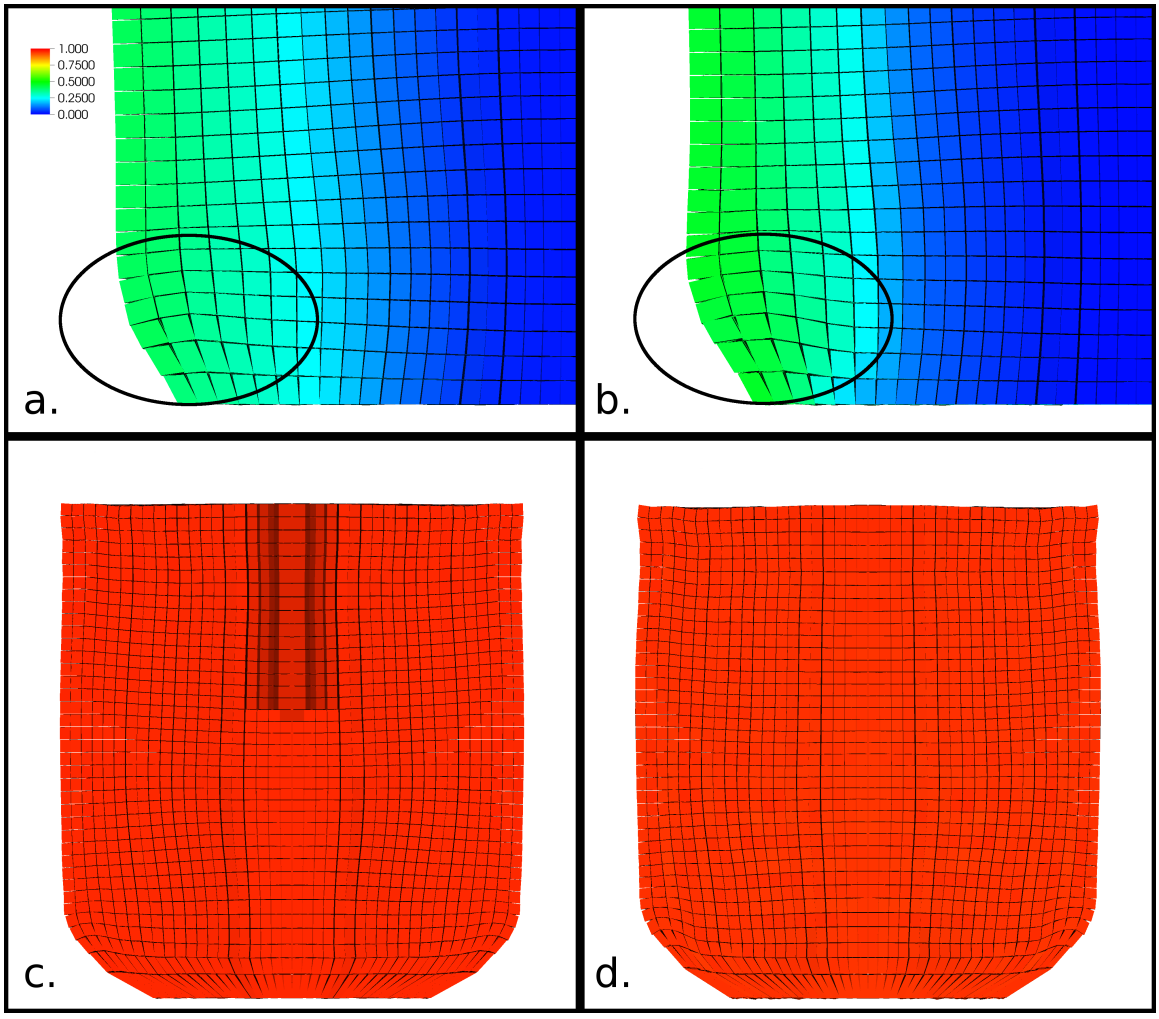

Fig. 7.8. Comparison of different diffusion parameters, (a) a pillar using diffusivity parameters $\alpha = 6.0$ and $\beta = 0.0$ during lithiation process. (b) a pillar using diffusivity parameters $\alpha = 6.0$ and $\beta = 1.0$ during lithiation process. (c) a pillar using diffusivity parameters $\alpha = 6.0$ and $\beta = 0.0$ at full saturation. Image (d) pillar using diffusivity parameters $\alpha = 6.0$ and $\beta = 1.0$ at full saturation.

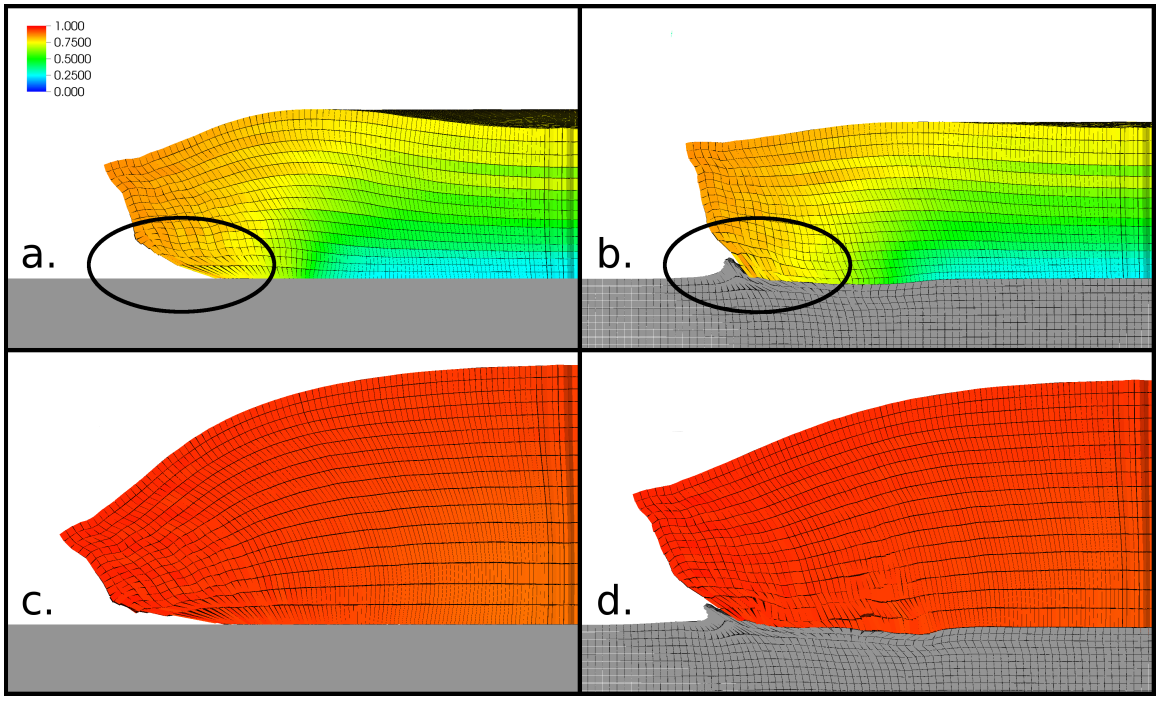

Fig. 7.9. Comparison of different substrate materials (a, c) pillar attached to a non-deformable material substrate. (b, d) pillar attached to a deformable material substrate.

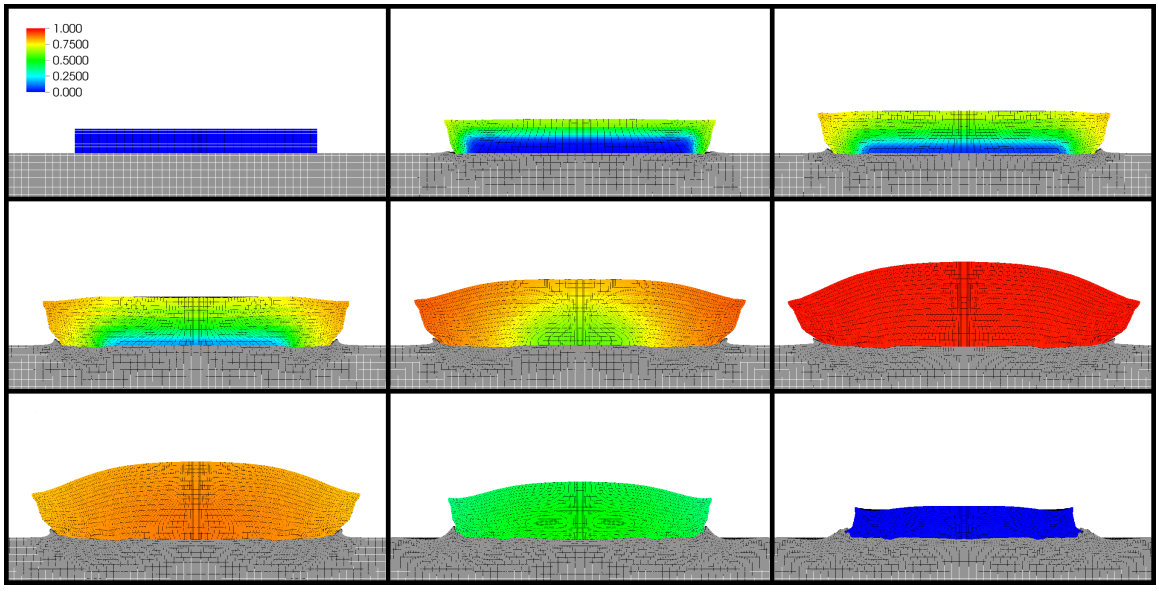

Fig. 7.10. A sequence of images showing the full lithiation and delithiation of a silicon nanopillar.

CHAPTER 8

MODELING A FULL CELL

In Chapter 7 a fully coupled chemical diffusion and deformation model was presented with the focus being on the anode. In this chapter the focus will be on modeling the full cell. As was the case in the previous chapter, certain assumptions will be made. The most notable will be the absence of physical deformation and the transport of chemical species by advection.

The outline of the chapter will proceed as follows: 1) a discussion of existing work will be given; 2) a discussion of the model along with the governing equations will be presented; 3) a discussion of the numerical methods used to solve the governing equations; 4) results will be presented; and 5) concluding thoughts.

8.1 Existing Work

One of the most popular methods to model electrochemical cells originally developed by Doyle, Fuller, and Newman is the Pseudo 2D (P2D) or Newman model [208, 209, 42]. This model simplifies the geometry of the electrochemical cell by treating the electrodes as spherical particles and the electrolyte is reduced to a one dimensional domain. The transport of positive ionic species within the solid electrode is governed by Fickian diffusion and within electrolyte it is governed by concentrated solution theory. The interaction between the electrode and electrolyte is governed by Butler-Volmer kinetics. To further reduce the computational complexity of the P2D model, Zhang et al. made a further simplifying assumption by neglecting variations in the concentrations of the electrolytes but retained in the model for the physics of spherical electrode particles and kinetic equation at the interface of the electrodes. This model is known as the single particle model (SPM) [210]. For a more complete description of the P2D and SPM models, the reader is referred to the review article by Jokar et al. [211].

Rossi et al. [212] performed numerical simulations to investigate thin film membrane

electrochemical cells. Continuum scale models are used and focus is given to exploring the microscopic region at the boundary and electrode/electrolyte interfaces. The boundary and interface conditions are described by Stern layer theory, and the reactions at the interfaces are described by the Frumkin-Butler-Volmer equations. The model of the electrochemical cell is defined by the coupled system of equations described by the Poisson-Nernst-Planck equations, which models the time evolutions of the field equations for both electrostatics and chemical species. To perform the numerical studies, a one dimension finite element formulation on an adaptively refined mesh as described by Wallmersperger et al. [213] is used. The time integration is done using an implict-Euler scheme coupled with a Newton-Raphson iterative method to solve the nonlinear equations that are found in the ionic species flux equation.

Martinez-Rosas et al. [214] propose a model based on the work of Doyle et al. [215], but with modified boundary conditions and the use a method of lines (MOL) approach to the numerical solution. Mirzadeh and Gibou [216] develop a conservative semi-implicit algorithm to solve the Poisson-Nernst-Planck equations. The spatial discretization is done using a hybrid finite-difference / finite volume method. A quad-tree-based adaptively refined Cartesian grid is used to resolve irregular boundaries. The adaptively refined grid also allows for the necessary refinement that is needed to resolve the exponential varying quantities that are involved in the electric double layer. The boundary of the domain is represented as the zero level set of a signed distance function. Through numerical experiments conservation of species is shown and accuracy is second order.

Smith and Bazant [217] propose a multiphase porous electrode theory framework for modeling electrodes that is based on nonequilibrium thermodynamics. The theory is implemented in a python based software package that uses a python based DAE library which wraps popular open source numerical software libraries such as SUNDIALS and ADOL-C.

Analysis of the geometric structure of anodes and cathodes is another key research area in full cell modeling. Miranda et al. [218] looked at how geometric properties of an interdigitized anode and cathode affected the performance of an electrochemical cell. They used the electrochemical model developed by Doyle et al. [215] and FEM to perform the analysis of 2d structures. Priimagi et al. [219] looked at geometric properties of 3d pillars for anode and cathode. They used porous electrode and Newman's concentration theory for

the cell model [220, 208] and FEM to perform the numerical solutions.

8.2 Model Description and Governing Equations

This section will cover the equations that govern the transport of ionic species through the full cell and the transport of electrons through the electrodes. First is a general description of the Poisson-Nernst-Planck, which is then followed by specific descriptions of the equations that are used for both positive and negatively charged chemical species for each material in the cell. Last will be a discussion of the electrode/electrolyte flux model.

8.2.1 Poisson-Nernst-Planck Equation

The Poisson-Nernst-Planck equation describes the transport of charged chemical species acting under an electrostatic field. The Poisson equation that describes the electrostatic field is written as [60]

$$\nabla^2\phi = -\frac{\rho}{\epsilon_0}, \tag{8.1}$$

where ϕ is the electrostatic potential, ρ is the charge density, and ϵ_0 is the relative permittivity. The Nernst-Plack equation describes the evolution of a charged chemical species acting under the electrostatic field [41]. The mass flux of the charged species is

$$\mathbf{j} = -D(\nabla c + c\frac{ze}{k_BT}\nabla\phi) + c\mathbf{v} \tag{8.2}$$

where c is the concentration of the changed chemical species, D is the diffusivity, z is the valence, e is the unit charge, k_B is the Boltzmann's constant, T is the temperature, ϕ is the electrostatic potential, and $\mathbf{v}$ is the velocity of the host material. The mass flux of the charged species is then combined with the continuity equation to describe the time rate of change of the charged chemical species, i.e.,

$$\frac{\partial c}{\partial t} = -\nabla\cdot\mathbf{j} \tag{8.3}$$

$$= -\nabla\cdot\left[-D(\nabla c + c\frac{ze}{k_BT}\nabla\phi) + c\mathbf{v}\right]. \tag{8.4}$$

A derivation of (8.2), from a thermodynamic perspective, is presented in Chapter 2, but in this section an examination of the physical nature of the problem is discussed. At its core the Nernst-Planck equation is an advection-diffusion equation. This can be seen when (8.4) is rewritten as follows:

$$\frac{\partial c}{\partial t} = \nabla\cdot(D\nabla c) - \nabla\cdot(c\mathbf{v}^*). \tag{8.5}$$

where the term $\mathbf{v}^*$ shall be called the total velocity. Two physical processes are at play to produce the total velocity. The first is the movement of the charged chemical species acting under an applied electrostatic field and the second is the resultant change in position that comes about from movement of the host material. The total velocity is then the sum of the two velocities, i.e.,

$$\mathbf{v}^* = \mathbf{v}_{es} + \mathbf{v}_h. \tag{8.6}$$

The electrostatically induced change in position is defined as

$$\mathbf{v}_{es} = -D\frac{ze}{k_B T}\nabla\phi. \tag{8.7}$$

The velocity based on the host material, $\mathbf{v}_h$, is material specific. For example, in a fluid, the velocity could be derived from the Navier-Stokes equations or in a solid material, the velocity is derived from the momentum balance (2.94).

8.2.2 Electron Transport in Semiconductive Electrodes

For this model semiconductive material such as silicon and lithium iron phosphate are being explored as the electrode materials. The current density within a semiconductive material is described by the following equation:

$$\mathbf{i} = -eD_e\nabla c_e \ + eD_h\nabla c_h + ec_e\mu_e\nabla\phi - ec_h\mu_h\nabla\phi, \tag{8.8}$$

where c_h is the concentration of holes and c_e is the concentration of electrons. The current density described by Equation (8.8) can be broken into a current based on diffusion and current based on drift,

$$\mathbf{i} = \underbrace{-eD_e\nabla c_e \ + eD_h\nabla c_h,}_{\text{diffusion current}} + \underbrace{ec_e\mu_e\nabla\phi - ec_h\mu_h\nabla\phi}_{\text{drift current}}. \tag{8.9}$$

The current density can also written in terms of the sum of a electron current density and a hole current density,

$$\mathbf{i} = \mathbf{i}_e + \mathbf{i}_h. \tag{8.10}$$

For the purposes of this chapter the hole base current density will be assumed to be zero and thus the current density will consist solely of the electron diffusion and drift currents,

$$\mathbf{i} = \mathbf{i}_e = \underbrace{-eD_e\nabla c_e}_{\text{diffusion current}} + \underbrace{ec_e\mu_e\nabla\phi}_{\text{drift current}}. \tag{8.11}$$

The relationship between the electron current density and flux is defined by the following relationship:

$$\mathbf{i}_e = e\mathbf{j}_e. \tag{8.12}$$

Based on Equation (8.12) the flux of electrons, with its diffusion and advection components, is defined as

$$\mathbf{j}_e = \underbrace{-D_e \nabla c_e}_{\text{diffusion}} + \underbrace{c_e \mu_e \nabla \phi}_{\text{advection}}. \tag{8.13}$$

8.2.3 Electrolyte

Within the electrolyte the governing equation for the electrostatic field is defined by the Poisson equation,

$$\nabla \cdot \epsilon_E \nabla \phi = -\frac{e}{\epsilon_0} \sum_i z_i c_i, \tag{8.14}$$

where ϕ is the electrostatic potential, ϵ_E is the relative permittivity of the electrolyte, e is the unit charge, ϵ_0 is the absolute permittivity, z_i is the valence of chemical species i and c_i is the concentration of species i. For the purposes of this chapter a binary electrolyte will be assumed with the valence of the positive and negative ionic species being 1 and -1, respectively. In terms of the binary electrolyte, the Poisson equation is written as

$$\nabla \cdot \epsilon_E \nabla \phi = -\frac{e}{\epsilon_0}(p - n), \tag{8.15}$$

with p being the concentration of the positive ionic species and n being the concentration of the negative species.

The mass transport of each ionic species within the electrolyte medium is defined by the following set of equations:

$$\begin{aligned} \frac{\partial c_i}{\partial t} &= -\nabla \cdot \mathbf{j}_i, \\ \mathbf{j}_i &= -D_i \left[\nabla c_i + \frac{e z_i c_i}{k_b T} \nabla \phi \right], \end{aligned} \tag{8.16}$$

where $\mathbf{j}_i$ is the flux of chemical species i within the electrolyte, D_i is the diffusivity coefficient for species i within the electrolyte, k_b is the Boltzmann's constant, and T is the temperature. The above set of equations are generalized to describe the mass transport of any given

species with any given valence within the electrolyte. Assuming a binary species within the electrolyte the mass transport of the positive ionic species is written as

$$\begin{aligned} \frac{\partial p}{\partial t} &= -\nabla \cdot \mathbf{j}_p, \\ \mathbf{j}_p &= -D_p \left[\nabla p + \frac{ep}{k_b T} \nabla \phi \right], \end{aligned} \tag{8.17}$$

and for the negative ionic species the mass transport equations are written as,

$$\begin{aligned} \frac{\partial n}{\partial t} &= -\nabla \cdot \mathbf{j}_n, \\ \mathbf{j}_n &= -D_n \left[\nabla n - \frac{en}{k_b T} \nabla \phi \right]. \end{aligned} \tag{8.18}$$

8.2.4 Anode and Cathode

The governing equation for the electrostatic potential field is defined in a similar manner to that of the electrolyte. The negative species are the free electrons and the positive species are the positive ionic species found in the electrolyte. Assuming a valence of 1 for the positive ionic species, the electrostatic potential equation is written as

$$\nabla \cdot \epsilon_{A|C} \nabla \phi = -\frac{e}{\epsilon_0}(p - n) \tag{8.19}$$

where $\epsilon_{A|C}$ is the permittivity of the anode or cathode material, p is the concentration of the positive ionic species, and n is the concentrations of the electrons.

The mass transport of the positive species within the anode or cathode is defined by the following set of equations:

$$\begin{aligned} \frac{\partial p}{\partial t} &= -\nabla \cdot \mathbf{j}_p, \\ \mathbf{j}_p &= -D_p \left[\nabla p + \frac{ep}{k_b T} \nabla \phi \right]. \end{aligned} \tag{8.20}$$

This set of equations is of the same form as those used in the electrolyte as defined by Equation (8.17) with the exception being that the diffusion coefficient is with respect to the anode or cathode material.

Compared to the transport equations for the positive species the transport of the electrons is defined in terms of the electronic conductivity of the anode or cathode material. The mass transport equations are

$$\begin{aligned} \frac{\partial n}{\partial t} &= -\nabla \cdot \mathbf{j}_n, \\ \mathbf{j}_n &= \sigma_{A|C} \nabla \phi, \end{aligned} \tag{8.21}$$

where $\sigma_{A|C}$ is the conductivity of the anode or cathode material.

8.2.5 Electrode/Electrolyte Interface

The kinetics of an ionic chemical species crossing the electrode/electrolyte interface is defined by the Butler-Volmer equation or in cases of a large overpotential the Tafel equation [41]. The Tafel equation for the interfacial current density as a function of the overpotential is defined as

$$\mathbf{i} = a^* \exp\left(\frac{\Delta\phi}{b^*}\right). \tag{8.22}$$

where $\Delta\phi$ is the overpotential defined as $\Delta\phi = \phi - \phi_0$. An approximation to the Tafel equation can be derived as is shown in Fig. 8.1. In this example it can be seen that if the linear approximation is used, the current density will be zero until the overpotential exceeds a value of one. An argument can be made that for this particular case the approximation is valid because the current density computed by the Tafel equation is near zero within the overpotential region from zero to one.

For the full-cell model in this chapter a course-grained electrode/electrolyte interfacial flux model is proposed that follows a similar argument to the one presented above. In this model an overpotential threshold is set such that as long as the overpotential is within the range of zero to the set threshold, the interfacial flux is zero. Once the overpotenial threshold is exceeded the interfacial flux is computed based on concentrations and electrostic potential gradients. The equation for the interface flux is as defined as follows:

$$\mathbf{j} = \begin{cases} 0 & \text{if } |\Delta\phi| < \gamma_t \\ -\alpha\frac{D_A+D_E}{2}\left[\nabla p + \frac{ep}{k_bT}\nabla\phi\right] & \text{if } |\Delta\phi| >= \gamma_t \end{cases}, \tag{8.23}$$

where $D_{A|C}$ and D_E are the diffusion coefficients of the ionic species in the anode or cathode and electrolyte, respectively, α is a user-defined reduction parameter that can be used to increase or decrease the magnitude of the flux across the interface, and γ_t is a user-defined overpotential threshold.

8.3 Numerical Approach to Modeling the Full Electrochemical Cell

A cell-centered finite volume discretization based on a Cartesian mesh will be used to solve the governing set of equation within the electrolyte, anode, and cathode. The problem domain, Ω, can be broken up in to subdomains, Ω_j, defined by the elements of the Cartesian mesh. The union of the element subdomains comprises the entirety of the problem domain,

$$\Omega = \bigcup_j \Omega_j. \tag{8.24}$$

The intersection of the subdomains is the empty set,

$$\bigcap_j \Omega_j = \emptyset. \tag{8.25}$$

For notational purposes the subscript letter j represents the global index of the j^{th} element.

8.3.1 Volume Integral of the Poisson Equation

Whether the area of focus is the anode, cathode, or electrolyte the equation that resolves the electrostatic potential field is defined by the Poisson equation,

$$\nabla \cdot \epsilon_x \nabla \phi = -\alpha \sum_i z_i c_i. \tag{8.26}$$

Taking the volume integral of both sides of (8.26) produces the following:

$$\int_\Omega (\nabla \cdot \epsilon_x \nabla \phi)\, dV = -\alpha \int_\Omega \left(\sum_i z_i c_i \right) dV, \tag{8.27}$$

and written in terms of the union of subdomains defined by the elements produces

$$\sum_j \int_{\Omega_j} (\nabla \cdot \epsilon_x \nabla \phi)\, dV = -\alpha \sum_j \int_{\Omega_j} \left(\sum_i z_i c_i \right) dV. \tag{8.28}$$

Using the divergence theorem the volume integral for each subdomain as found on the left hand side of (8.28) is written as a surface integral,

$$\int_{\Omega_j} (\nabla \cdot \epsilon_x \nabla \phi)\, dV = \int_{\partial \Omega_j} (\epsilon_x \nabla \phi \cdot \mathbf{n})\, dA, \tag{8.29}$$

where $\partial \Omega_j$ is the surface of element j. Using the observation made in (8.29), (8.28) can be rewritten as

$$\sum_j \int_{\partial \Omega_j} (\epsilon_x \nabla \phi \cdot \mathbf{n})\, dA = -\alpha \sum_j \int_{\Omega_j} \left(\sum_i z_i c_i \right) dV. \tag{8.30}$$

8.3.2 Volume Integral of the Mass Transport Equation

As was the case for resolving the electrostatic potential field the same is true for the mass transport of chemical species. The mass transport equation for each chemical species in the electrolyte, anode, and cathode can be written in the generalized form,

$$\frac{\partial c_i}{\partial t} = -\nabla \cdot \mathbf{j}_i. \tag{8.31}$$

Taking the volume integral of both sides of (8.31) produces

$$\int_{\Omega} \frac{\partial c_i}{\partial t} dV = -\int_{\Omega} \nabla \cdot \mathbf{j}_i \, dV, \tag{8.32}$$

and written in terms the union of the subdomains,

$$\sum_j \int_{\Omega_j} \frac{\partial c_i}{\partial t} dV = -\sum_j \int_{\Omega_j} \nabla \cdot \mathbf{j}_i \, dV. \tag{8.33}$$

As was the case with the Poisson equation, the divergence theorem is used to transform the volume integral on the right-hand side of Equation (8.33) to a surface integral,

$$\int_{\Omega_j} \nabla \cdot \mathbf{j}_i dV = \int_{\partial\Omega_j} \mathbf{j}_i \cdot \mathbf{n} \, dA. \tag{8.34}$$

8.3.3 Approximation of Field Equations

The function $p(\mathbf{x}, t)$ will be used to define the concentration of the positive species in the electrolyte, anode, and cathode. The function $n(\mathbf{x}, t)$ will be used to define the concentration of the negative species in the electrolyte and the concentration of free electrons within the anode and cathode. The function $\phi(\mathbf{x}, t)$ will be used to define the electrostatic potential field. An approximation of the functions $p(\mathbf{x}, t)$, $n(\mathbf{x}, t)$, and $\phi(\mathbf{x}, t)$ will be done using the respective values at the cell-centers and linear basis functions,

$$p(\mathbf{x}, t) \approx \sum_j p_j(t)\psi_j(\mathbf{x}), \tag{8.35}$$

$$n(\mathbf{x}, t) \approx \sum_j n_j(t)\psi_j(\mathbf{x}), \tag{8.36}$$

$$\phi(\mathbf{x}, t) \approx \sum_j \phi_j(t)\psi_j(\mathbf{x}). \tag{8.37}$$

The approximation of the gradients of the field equations are derived as follows:

$$\nabla p(\mathbf{x}, t) \approx \sum_j p_j(t)\nabla\psi_j(\mathbf{x}), \tag{8.38}$$

$$\nabla n(\mathbf{x}, t) \approx \sum_j n_j(t)\nabla\psi_j(\mathbf{x}), \tag{8.39}$$

$$\nabla \phi(\mathbf{x}, t) \approx \sum_j \phi_j(t)\nabla\psi_j(\mathbf{x}). \tag{8.40}$$

8.3.4 The One-Dimensional Problem

In order to simplify the discussion of the spatial discretization using the finite volume a one-dimensional problem will be used. All elements within the problem domain will be of

uniform size, dx. Because the problem has been reduced to one dimension the two faces of an element can be referred to as the left and right faces, and the notation for a flux across an interface will use the following notation:

$$\mathbf{j}_{p,j,l} \tag{8.41}$$

where the subscripts indicate that this is the flux of the positive across the left face of the j^{th} element. Similar notation is used to describe a gradient across an element face,

$$\nabla\phi_{j,l} \tag{8.42}$$

where this would be that gradient of the electrostatic potential across the left face of the j^{th} element.

8.3.5 Finite Volume Formulation of the Electrolyte Governing Equations

In one dimension the surface integral as described in (8.29) is written as

$$\int_{\partial\Omega_j} (\epsilon_E \nabla\phi \cdot \mathbf{n})\, dA \quad \rightarrow \quad \epsilon_E \nabla\phi_{j+1/2} - \epsilon_E \nabla\phi_{j-1/2}, \tag{8.43}$$

where the subscript $j \pm 1/2$ represents the evaluation of the gradient of ϕ at the left and right sides of element j. Using the notation described above the gradient evaluations at the left and right sides of the element j are

$$\nabla\phi_{j-1/2} = \nabla\phi_{j,l}, \tag{8.44}$$

$$\nabla\phi_{j+1/2} = \nabla\phi_{j,r}. \tag{8.45}$$

Using linear basis functions the gradients of the electrostatic potential for the left and right sides of element j are

$$\begin{aligned} \nabla\phi_{j,l} &\approx \frac{\phi_j - \phi_{j-1}}{dx}, \\ \nabla\phi_{j,r} &\approx \frac{\phi_{j+1} - \phi_j}{dx}. \end{aligned} \tag{8.46}$$

Using the gradient approximation described by (8.46) the approximation to (8.43) is

$$\begin{aligned} \epsilon_E \nabla\phi_{j+1/2} - \epsilon_E \nabla\phi_{j-1/2} &\approx \epsilon_E \frac{(\phi_{i+1} - \phi_i)}{dx} - \epsilon_E \frac{(\phi_i - \phi_{i-1})}{dx}, \\ &= \epsilon_E \frac{(\phi_{i+1} - 2\phi_i + \phi_{i-1})}{dx}. \end{aligned} \tag{8.47}$$

The volume integral on the right hand side of the Poisson equation is approximated using a midpoint integration method,

$$-\frac{e}{\epsilon_0} \int_{\Omega_j} (p - n)\, dV \approx -\frac{e}{\epsilon_0} (p_j - n_j) dx. \tag{8.48}$$

By combining (8.47) and (8.48) the finite volume approximation of the Poisson equation is

$$\frac{\phi_{j-1} - 2\phi_j + \phi_{j+1}}{dx^2} = -\frac{e}{\epsilon_E \epsilon_0}(p_j - n_j). \tag{8.49}$$

As was described in an earlier section, the Nernst-Planck flux equation can be written in terms of a diffusion component and an advection component. In finite volume descretizations of the advection equations care needs to be taken when determining the form of the flux calculations. When sharp gradients are present oscillations can arise if care is not taken when choosing the numerical form of the flux equations. With this in mind the numerical form of the fluxes for the positive chemical species within the electrolyte region of the chemical cell is defined by the following set of equations:

$$\begin{aligned} \mathbf{j}_{p,j,L} &= -D_p \frac{(p_{j-1} - p_j)}{dx} - p_{l,L} \frac{D_p e}{k_b T} \frac{(\phi_{j-1} - \phi_j)}{dx}, \\ \mathbf{j}_{p,j,R} &= -D_p \frac{(p_{j+1} - p_j)}{dx} - p_{l,R} \frac{D_p e}{k_b T} \frac{(\phi_{j+1} - \phi_j)}{dx}, \end{aligned} \tag{8.50}$$

where the values of $p_{l,L}$ and $p_{l,R}$ are the computed values of the left and right concentration values. A description of how the values are calculated will be given in the section on limiters.

The numerical solution for the flux values of the negative species within the electrolyte is defined in a like manner,

$$\begin{aligned} \mathbf{j}_{n,j,l} &= -D_n \frac{(n_{j-1} - n_j)}{dx} + \frac{D_n e}{k_b T} \frac{(n_{j-1} + n_j)}{2} \frac{(\phi_{j-1} - \phi_j)}{dx}, \\ \mathbf{j}_{n,j,r} &= -D_n \frac{(n_{j+1} - n_j)}{dx} + \frac{D_n e}{k_b T} \frac{(n_{j+1} + n_j)}{2} \frac{(\phi_{j+1} - \phi_j)}{dx}. \end{aligned} \tag{8.51}$$

The rate of change of the average concentration for element j is then determined by the following equation:

$$\frac{\partial p_j}{\partial t} = -\frac{1}{dx}(\mathbf{j}_{p,j,l} + \mathbf{j}_{p,j,r}). \tag{8.52}$$

In similar fashion the rate of change of the negative ionic species is determined by the following equation:

$$\frac{\partial n_j}{\partial t} = -\frac{1}{dx}(\mathbf{j}_{n,j,l} + \mathbf{j}_{n,j,r}). \tag{8.53}$$

A further discussion of the time integration methods will be presented below.

8.3.6 Anode/Cathode

The numerical approximation to the Poisson problem that determines the electrostatic potential field within the anode and cathode is formulated in the same way that is done for the electrolyte. The numerical approximation is written as follows:

$$\frac{\phi_{j-1} - 2\phi_j + \phi_{j+1}}{dx^2} = -\frac{e}{\epsilon_{A|C}\epsilon_0}(p_j - n_j) \tag{8.54}$$

where $\epsilon_{A|C}$ is the relative permittivity of the respective anode or cathode material. The numerical approximation of the flux of the positive species within the anode or cathode material is determined in like manner to that of the positive species in the electrolyte and is written as follows:

$$\begin{aligned} \mathbf{j}_{p,j,l} &= -D_p\frac{(p_{j-1} - p_j)}{dx} - p_{l,L}\frac{D_p e}{k_b T}\frac{(\phi_{j-1} - \phi_j)}{dx}, \\ \mathbf{j}_{p,j,r} &= -D_p\frac{(p_{j+1} - p_j)}{dx} - p_{l,R}\frac{D_p e}{k_b T}\frac{(\phi_{j+1} - \phi_j)}{dx}. \end{aligned} \tag{8.55}$$

The numerical approximations to the flux of negative species in the anode and cathode is formulated in a similar fashion to the numerical approximation of the positive with changes made in the appropriate constants,

$$\begin{aligned} \mathbf{j}_{n,j,l} &= \sigma_A\frac{(\phi_{j-1} - \phi_j)}{dx}, \\ \mathbf{j}_{n,j,r} &= \sigma_A\frac{(\phi_{j+1} - \phi_j)}{dx}. \end{aligned} \tag{8.56}$$

The rate of change of the cell averaged positive and electron species concentrations is determined by the following set of equations:

$$\begin{aligned} \frac{\partial p_j}{\partial t} &= -\frac{1}{dx}(\mathbf{j}_{p,j,l} + \mathbf{j}_{p,j,r}), \\ \frac{\partial n_j}{\partial t} &= -\frac{1}{dx}(\mathbf{j}_{n,j,l} + \mathbf{j}_{n,j,r}). \end{aligned} \tag{8.57}$$

8.3.7 Electrode/Electrolyte Interface

A description of the numerical implementation of the electrode/electrolyte interface will be presented in terms of the anode/electrolyte interface but the same approach applies to the cathode interface. For this example at the anode/electrolyte interface the anode is to the left of the interface and the electrolyte is to the right. The element on the anode side of the interface is element j and the element on the electrolyte side of the interface is element $j + 1$.

The discretization of the Poisson equation for element j is written as follows:

$$\left[\epsilon_A \frac{(\phi_{j-1} - \phi_j)}{dx} + \frac{(\epsilon_A + \epsilon_E)}{2} \frac{(\phi_{j+1} - \phi_j)}{dx}\right] \frac{1}{dx} = -\frac{e}{k_b T}(p_j - n_j). \tag{8.58}$$

The permittivity at the interface is the average of the two material permittivities. While a higher order approximation would produce more accurate results, the averaging of the two produces sufficient results for the given simulation.

The discretization of the Poisson equation for element $j+1$ is written as follows:

$$\left[\frac{(\epsilon_A + \epsilon_E)}{2} \frac{(\phi_j - \phi_{j+1})}{dx} + \epsilon_C \frac{(\phi_{j+2} - \phi_{j+1})}{dx}\right] \frac{1}{dx} = -\frac{e}{k_b T}(p_{j+1} - n_{j+1}). \tag{8.59}$$

8.3.7.1 Free Electron and Negative Ionic Species Transfer

In the electrochemical full cell model neither the electrons in the electrodes nor the negative ionic species cross the electrode/electrolyte interface, as such the numerical implementation needs to respect this requirement. The numerical implementation for the jth element is written as follows:

$$\begin{aligned} \mathbf{j}_{n,j,l} &= \sigma_A \frac{(\phi_{j-1} - \phi_j)}{dx}, \\ \mathbf{j}_{n,j,r} &= 0, \\ n'_j &= -\frac{1}{dx}(\mathbf{j}_{n,j,l} + \mathbf{j}_{n,j,r}). \end{aligned} \tag{8.60}$$

The discretization of element $j+1$ is written as follows:

$$\begin{aligned} \mathbf{j}_{n,j+1,l} &= 0, \\ \mathbf{j}_{n,j+1,r} &= -D_n \frac{(n_{j+2} - n_{j+1})}{dx} - \frac{D_p e}{k_b T} \frac{(n_{j+2} + n_{j+1})}{2} \frac{(\phi_{j+2} - \phi_{j+1})}{dx}, \\ n'_{j+1} &= -\frac{1}{dx}(\mathbf{J}_{n,j+1,l} + \mathbf{J}_{n,j+1,r}). \end{aligned} \tag{8.61}$$

8.3.7.2 Positive Ionic Species Transfer

In the case of the positive ionic species, electrons are allowed to transfer across the interface. The drop in electrostatic potential is simply determined by taking the difference of the electrostatic potentials of the two adjacent elements at the interface,

$$\Delta\phi = \phi_j - \phi_{j+1}. \tag{8.62}$$

The discretization for element j is written as follows:

$$\begin{aligned}
\mathbf{j}_{p,j,l} &= -D_A \left[\frac{(p_{j-1} - p_j)}{dx} - \frac{e}{k_b T} \frac{(p_{j-1} + p_j)}{2} \frac{(\phi_{j-1} - \phi_j)}{dx} \right], \\
\mathbf{j}_{p,j,r} &= \begin{cases} 0 & \text{if } |\Delta\phi| < \gamma_t \\ -\alpha \frac{D_A + D_E}{2} \left[\frac{(p_{j+1} - p_j)}{dx} + \frac{e}{k_b T} \frac{(p_{j+1} + p_j)}{2} \frac{(\phi_{j+1} - \phi_j)}{dx} \right] & \text{if } |\Delta\phi| >= \gamma_t \end{cases}, \\
p'_j &= -\frac{1}{dx} (\mathbf{j}_{p,j,l} + \mathbf{j}_{p,j,r}).
\end{aligned} \tag{8.63}$$

The discretization of element $j+1$ is written as follows:

$$\begin{aligned}
\mathbf{j}_{p,j+1,l} &= \begin{cases} 0 & \text{if } |\Delta\phi| < \gamma_t \\ -\alpha \frac{D_A + D_E}{2} \left[\frac{(p_j - p_{j+1})}{dx} + \frac{e}{k_b T} \frac{(p_j + p_{j+1})}{2} \frac{(\phi_j - \phi_{j+1})}{dx} \right] & \text{if } |\Delta\phi| >= \gamma_t \end{cases}, \\
\mathbf{j}_{p,j+1,r} &= -D_E \left[\frac{(p_{j+2} - p_{j+1})}{dx} - \frac{e}{k_b T} \frac{(p_{j+2} + p_{j+1})}{2} \frac{(\phi_{j+2} - \phi_{j+1})}{dx} \right], \\
p'_j &= -\frac{1}{dx} (\mathbf{j}_{p,j+1,l} + \mathbf{j}_{p,j+1,r}).
\end{aligned} \tag{8.64}$$

8.4 Flux Limiters

Because of the overpotential threshold that is used in the interface flux calculation, the flux value goes from zero to computed value within one time step. If not addressed correctly, sudden change will introduce oscillations into computed results. The solution to this problem is the use of a flux limiter. For this particular case the van Leer limiter [221] was chosen and is implemented as follows:

$$\psi(r_i) = \frac{r_i + |r_i|}{1 + |r_i|} \tag{8.65}$$

$$r_i = \frac{(p_{i+1} - p_i)}{(p_i - p_{i-1})} \tag{8.66}$$

$$\begin{aligned}
p_L &= p_i + \frac{(p_i - p_{i-1})}{2} \psi(r_i), \\
p_R &= p_{i+1} - \frac{(p_{i+2} - p_{i+1})}{2} \psi\left(\frac{1}{r_{i+1}} \right).
\end{aligned} \tag{8.67}$$

In numerical experiments an upwind scheme was also shown to be favorable, but the use of limiters was chosen for robustness in preventing instabilities.

8.5 Time Integration Methods

8.5.1 DAE Index Classification of the Governing Equations

8.5.1.1 Binary Electrolyte

The DAE that describes the electrostatic potential field and the transport of positive and negative ionic chemical species within the electrolyte can be formed by combining the finite volume spatial discretizations defined by (8.49), (8.50), (8.51), (8.52), and (8.53),

$$p_i' = D\frac{p_{i-1} - 2p_i + p_{i+1}}{dx^2} + \frac{\alpha}{2dx^2}[(p_{i-1} + p_i)(\phi_{i-1} - \phi_i) + (p_{i+1} + p_i)(\phi_{i+1} - \phi_i)], \quad (8.68a)$$

$$n_i' = D\frac{n_{i-1} - 2n_i + n_{i+1}}{dx^2} - \frac{\alpha}{2dx^2}[(n_{i-1} + n_i)(\phi_{i-1} - \phi_i) + (n_{i+1} + n_i)(\phi_{i+1} - \phi_i)], \quad (8.68b)$$

$$0 = \frac{\phi_{i-1} - 2\phi_i + \phi_{i+1}}{dx^2} + \beta(p_i - n_i). \quad (8.68c)$$

The system of equations defined by (8.68) is a semi-explicit DAE of index 1. To demonstrate this fact it will help to write the system of equations in its matrix vector form,

$$\mathbf{p}' = \mathbf{M}\mathbf{p} + \frac{\alpha}{2dx^2}\left[(\mathbf{A_1}\mathbf{p}) \circ (\mathbf{B_1}\boldsymbol{\phi}) + (\mathbf{A_2}\mathbf{p}) \circ (\mathbf{B_2}\boldsymbol{\phi})\right], \quad (8.69a)$$

$$\mathbf{n}' = \mathbf{M}\mathbf{n} - \frac{\alpha}{2dx^2}\left[(\mathbf{A_1}\mathbf{n}) \circ (\mathbf{B_1}\boldsymbol{\phi}) + (\mathbf{A_2}\mathbf{n}) \circ (\mathbf{B_2}\boldsymbol{\phi})\right], \quad (8.69b)$$

$$\mathbf{0} = \mathbf{M}\boldsymbol{\phi} + \beta(\mathbf{p} - \mathbf{n}), \quad (8.69c)$$

where bold uppercase letters are matricies and bold lower case letters are vectors. The Hadamard product is defined by $\circ$. Differentiating Equation (8.69c) with respect to t produces the following:

$$\mathbf{0} = \mathbf{M}\boldsymbol{\phi}' + \beta(\mathbf{p}' - \mathbf{n}'). \quad (8.70)$$

Given that the matrix $\mathbf{M}$ is nonsingular, Equation (8.70) can be rewritten as

$$\boldsymbol{\phi}' = \beta\mathbf{M}^{-1}(\mathbf{n}' - \mathbf{p}'). \quad (8.71)$$

Replacing (8.69c) by (8.71) in the system of equations found in (8.69) produces the following ODE:

$$\mathbf{p}' = \mathbf{M}\mathbf{p} + \frac{\alpha}{2dx^2}\left[(\mathbf{A_1}\mathbf{p}) \circ (\mathbf{B_1}\boldsymbol{\phi}) + (\mathbf{A_2}\mathbf{p}) \circ (\mathbf{B_2}\boldsymbol{\phi})\right], \quad (8.72a)$$

$$\mathbf{n}' = \mathbf{M}\mathbf{n} - \frac{\alpha}{2dx^2}\left[(\mathbf{A_1}\mathbf{n}) \circ (\mathbf{B_1}\boldsymbol{\phi}) + (\mathbf{A_2}\mathbf{n}) \circ (\mathbf{B_2}\boldsymbol{\phi})\right], \quad (8.72b)$$

$$\boldsymbol{\phi}' = \beta\mathbf{M}^{-1}(\mathbf{n}' - \mathbf{p}'), \quad (8.72c)$$

thus showing that the system of equations defined by (8.69) is a DAE of index 1.

8.5.1.2 Anode/Cathode

The DAE that describes the electrostatic potential field and the transport of positive and negative ionic chemical species within the anode can be formed by combining the finite volume spatial discretizations defined by Equations (8.54), (8.55), (8.56), (8.57), and (8.57),

$$p_i' = D\frac{p_{i-1} - 2p_i + p_{i+1}}{dx^2} + \frac{\alpha}{2dx^2}[(p_{i-1} + p_i)(\phi_{i-1} - \phi_i) + (p_{i+1} + p_i)(\phi_{i+1} - \phi_i)], \quad (8.73a)$$

$$n_i' = -\sigma_A\frac{\phi_{i-1} - 2\phi_i + \phi_{i+1}}{dx^2} \quad (8.73b)$$

$$0 = \frac{\phi_{i-1} - 2\phi_i + \phi_{i+1}}{dx^2} + \beta(p_i - n_i). \quad (8.73c)$$

The system of equations defined by (8.73) is a semi-explicit DAE of index 1. To demonstrate this fact it will help to write the system of equations in its matrix vector form,

$$\mathbf{p}' = \mathbf{M}\mathbf{p} + \frac{\alpha}{2dx^2}\left[(\mathbf{A_1}\mathbf{p}) \circ (\mathbf{B_1}\boldsymbol{\phi}) + (\mathbf{A_2}\mathbf{p}) \circ (\mathbf{B_2}\boldsymbol{\phi})\right], \quad (8.74a)$$

$$\mathbf{n}' = \mathbf{M}\mathbf{n} - \frac{\alpha}{2dx^2}\left[(\mathbf{A_1}\mathbf{n}) \circ (\mathbf{B_1}\boldsymbol{\phi}) + (\mathbf{A_2}\mathbf{n}) \circ (\mathbf{B_2}\boldsymbol{\phi})\right], \quad (8.74b)$$

$$\mathbf{0} = \mathbf{M}\boldsymbol{\phi} + \beta(\mathbf{p} - \mathbf{n}), \quad (8.74c)$$

where bold uppercase letters are matricies and bold lower case letters are vectors. The Hadamard product is defined by $\circ$. Differentiating Equation (8.74c) with respect to t produces the following:

$$\mathbf{0} = \mathbf{M}\boldsymbol{\phi}' + \beta(\mathbf{p}' - \mathbf{n}'). \quad (8.75)$$

Given that the matrix $\mathbf{M}$ is nonsingular, Equation (8.75) can be rewritten as

$$\boldsymbol{\phi}' = \beta\mathbf{M}^{-1}(\mathbf{n}' - \mathbf{p}'). \quad (8.76)$$

Replacing (8.74c) by (8.76) in the system of Equations (8.74) produces the following ODE:

$$\mathbf{p}' = \mathbf{M}\mathbf{p} + \frac{\alpha}{2dx^2}\left[(\mathbf{A_1}\mathbf{p}) \circ (\mathbf{B_1}\boldsymbol{\phi}) + (\mathbf{A_2}\mathbf{p}) \circ (\mathbf{B_2}\boldsymbol{\phi})\right], \quad (8.77a)$$

$$\mathbf{n}' = \mathbf{M}\mathbf{n} - \frac{\alpha}{2dx^2}\left[(\mathbf{A_1}\mathbf{n}) \circ (\mathbf{B_1}\boldsymbol{\phi}) + (\mathbf{A_2}\mathbf{n}) \circ (\mathbf{B_2}\boldsymbol{\phi})\right], \quad (8.77b)$$

$$\boldsymbol{\phi}' = \beta\mathbf{M}^{-1}(\mathbf{n}' - \mathbf{p}'), \quad (8.77c)$$

thus showing that the system of equations defined by (8.73) is a DAE of index 1.

8.6 Numerical Tests

Each of the numerical simulations that will be run, to show different aspects of the numerics described above, will consist of an anode, electrolyte, and cathode. The entire

domain of the problem is $100\,nm$. The widths of the two electrodes are $20\,nm$ each with the remaining $60\,nm$ being devoted to the electrolyte. The left and right boundaries of the problem will be called boundary a and boundary b, respectively. Fig. 8.2 shows a simple schematic of the 1d problem. The electrode on the left (orange) will be the anode and the electrode on the right (red) will be the cathode.

For all numerical experiments the following set of common parameters are used: The relative permittivity of the anode, cathode, and electrolyte is 10 and the absolute permittivity is $8.854 \times 10^{-21}\,C/(V \cdot nm)$. All experiments assume an isothermal condition and a temperature of $300\,K$. The Boltzmann's constant and unit charge are $1.38 \times 10^{-23}\,J/K$ and $1.602 \times 10^{-19}\,C$, respectively.

8.6.1 Boundary Conditions

In this work explorations are done to examine the mass transport of lithium phenomena based on charging via an imposed current. Based on this criterion the boundary conditions for the anode and cathode are set to the following Neumann boundary condition values:

$$J_{n,anode} = 3\,mA/cm^2, \tag{8.78}$$

$$J_{n,cathode} = 3\,mA/cm^2, \tag{8.79}$$

with a positive flux going from left to right in the domain. With the flux at the anode and cathode being the same in both magnitude and direction, charge across the entire problem domain is conserved. For the positive lithium ions the boundaries are pure insulators and thus the Neumann boundary conditions are

$$J_{p,anode} = 0, \tag{8.80}$$

$$J_{p,cathode} = 0. \tag{8.81}$$

The boundary conditions for the electrostatic potential are a combination of Dirichlet boundary condition at the anode and a Neumann boundary condition on the cathode side. In the case of the electrostatic potential field we are dealing with a relative potential, which means that a given potential only has significance, if it is given in terms of a given reference. In this case we assign the following two the anode and cathode boundary conditions:

$$\phi_{anode} = 0, \tag{8.82}$$

$$J_{\phi,cathode} = 0. \tag{8.83}$$

A Dirichlet condition on one of the boundaries is necessary in order for the problem to be well-posed, and the Neumann condition will allow for the electrostatic potential to evolve according to the change in the charge field driven by the Nerst-Planck equations.

8.6.2 Resolution Test

The initial concentration in the anode is zero for both the free electrons and the positive ionic species. The concentration for both the positive and negative ionic species is in the electrolyte is $6.0\,\#/nm^3$. In the cathode the initial concentration for both the free electrons and positive ionic species is $30.0\,\#/nm^3$. The diffusivity values used for the positive ions in the anode and cathode are $10^{-5}\,cm^2/s$ and $10^{-3}\,cm^2/s$, respectively. The conductivity of the electrons in the anode and cathode are $3.0 \times 10^{-18} S/cm$ and $10^{-17} S/cm$, respectively. In the electrolyte the negative ions remain fixed, i.e., a diffusivity of 0, and the diffusivity of the positive ions is $1.7 \times 10^{-4} cm^2/s$.

A series of simulations run in order to show the convergence of the numerical approach as the grid is refined. In this study each grid has a uniform spacing between cell-centers. The number of cells used for the simulations are 10^2, 10^3, 10^4, and 10^5. Each simulation uses the same time step size of 10^{-4} and the simulation is run for 10^5 iterations. Fig. 8.3 shows a comparison of the four refinement levels at the 10^5 timestep. The charge plot is the net total charge in each cell, i.e.,

$$\text{charge} = (p_i - n_i)\Delta x_i. \tag{8.84}$$

Fig. 8.4 shows one area of the the potential and charge plots from Fig. 8.3. In this plot it can easily be seen the relationship between the electrostatic potential and the net charge as the grid is refined.

An alternate look at the convergence of the numerical method can be seen through the calculation of an error norm for each resolution. The error is computed by treating the solution based on the 10^5 resolution grid as the true solution. The L2 error norm is computed in the following manner:

$$\text{error} = \sqrt{\frac{\sum_i \left(f_i - f(x_i)\right)^2}{N}} \tag{8.85}$$

where N is the total number of cells. Fig. 8.5 shows the calculated error norm of the electrostatic potential for each of the three grid refinements.

8.6.3 Effect of Flux Limiters

This set of numerical tests are presented to show how the use of the flux limiter removes the oscillations that can occur during a numerical simulation. The initial concentration of free electrons and ions are in the anode. In this test case there are three locations at which oscillations will form. The first is the anode/electrolyte interface. These oscillations begin as the positive ions in the electrolyte begin to move toward the anode and build up a higher concentration of the positive ions in the stern layer. The second location where oscillations begin to develop is at the electrolyte/cathode interface. These oscillations are initiated as the electrostatic potential difference across the electrolyte/cathode interface exceed the given threshold and positive ions from the cathode are then allowed to enter into the electrolyte. The third location where oscillations begin to build up is at the cathode/electrolyte interface, with the oscillations this time being on the cathode side. These oscillations develop as the positive ionic species within the cathode begin to migrate toward the cathode/electrolyte interface as a result of electrostatic potential gradients across the cathode. An example of the three oscillations that develop can be seen in Fig. 8.6.

Fig. 8.7 shows a comparison between the nonlimiter version and limiter version of the simulations at the anode/electrolyte interface. From this figure it can be seen the the limiter removes the oscillations that develop as positive ions migrate from the bulk electrolyte toward the anode/electrolyte interface.

Fig. 8.8 shows the effectiveness of the limiter in mitigating the oscillations that form as the positive ionic species transfer from the cathode into the electrolyte. As the positive ionic species enter the electrolyte and begin to migrate toward the anode, the oscillations, which come as a result of the advection portion of the flux equation, begin to develop. The top sequence of images show the results of using an averaged value of the positive ionic concentrations between the two cells as compared to the limiter version. In the limiter version the oscillations no longer exist.

Fig. 8.9 shows an example of the oscillations that develop at the cathode/electrolyte interface. As compared to the other areas of the simulations these oscillations continue to persist throughout the simulation. The persistance of these oscillations comes as a result of the electrostatic gradients that remain through the duration of the charging phase of the simulation. In the other two areas the electrostatic gradient tends to flatten out in the

beginning portions of the simulation, and thus the transport of the positive species within the anode and electrolyte is driven by the diffusion portion of the flux equation. The diffusion process begins to even out the oscillations as the simulation proceeds. In the cathode this physical process does not occur in the same manner so the oscillations will continue to grow at the interface during the simulation. The limiter based version is remove the oscillations. In Fig. 8.9 the top sequence of images shows the nonlimiter version as compared to the limiter-based version. Comparing the two shows the effects of the limiter in removing the nonphysical oscillations.

8.6.4 Full-Charge Simulation

In this section a simulation is run that will show the full-charge cycle. The initial concentration in the anode is zero for both the free electrons and the positive ionic species. The concentration for both the positive and negative ionic species is in the electrolyte is $6.0\,\#/nm^3$. In the cathode the initial concentration for both the free electrons and positive ionic species is $30.0\,\#/nm^3$. The diffusivity values used for the positive ions in the anode and cathode are $10^{-5}\,cm^2/s$ and $10^{-3}\,cm^2/s$, respectively. The conductivity of the electrons in the anode and cathode is $3.0\times10^{-18}S/cm$ and $10^{-17}S/cm$, respectively. In the electrolyte the negative ions remain fixed, i.e., a diffusivity of 0, and the diffusivity of the positive ions is $1.7\times10^{-4}cm^2/s$. Fig. 8.10 shows the results from the simulation at times $0.0s$, $1.0s$, and $2.0s$.

8.7 Conclusion

In this chapter we developed a full one dimensional electrochemical cell model using the Poisson-Nernst-Planck equations and a novel electode/electrolyte interface flux. The electrode/electrolyte interface flux model uses an overpotential threshold that limits the flow of ions across the interface until a potential differential is met. This buildup of charge leads to sharp gradients that result in a large flux of ions across the interface. From a numerical perspective this sudden flux of ions across the interface can lead to numerical instabilities in the model. In traditional fluid mechanics flux limiters provide a valuable tool to handle numerical instabilites of the type seen at the electrode/electrolyte interface. As was described in Equation (8.13) the transport of ions comes as a result of both diffusion

and advection. The the use of a flux limiter in the model stops the numerical instabilities that would otherwise occur.

The work that is presented in the chapter is a preliminary model. Ouermi [222] has updated the model so that the effects of the electrode at the interface are modeled implicitly and has incorporated Butler-Volmer kinetics [41] as boundary conditions. Any future next steps in the model development will need to incorporate the work of Smith and Bazant [217] into the updates made by Ouermi.

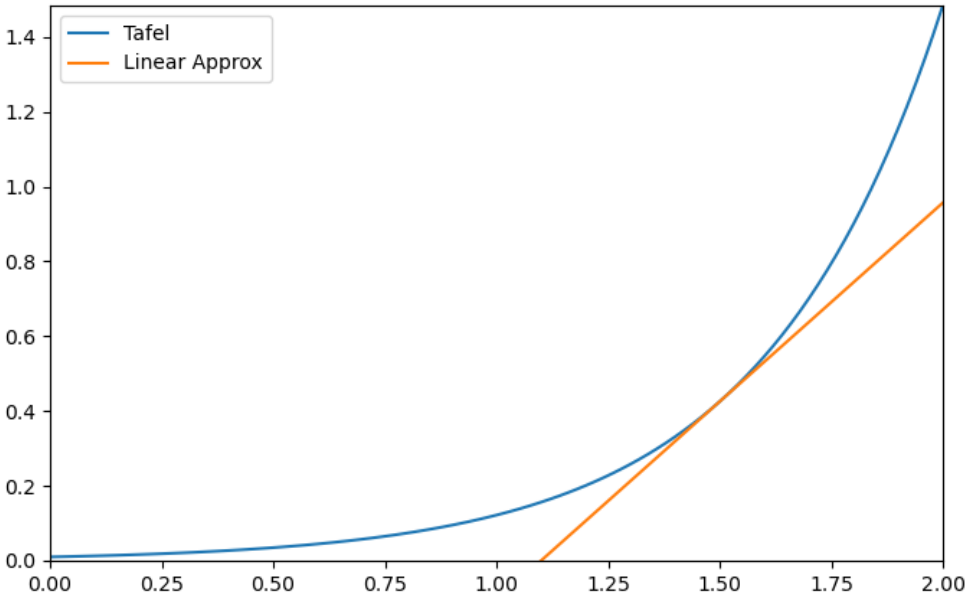

Fig. 8.1. Tafel approximation.

Fig. 8.2. Schematic of one dimensional problem.

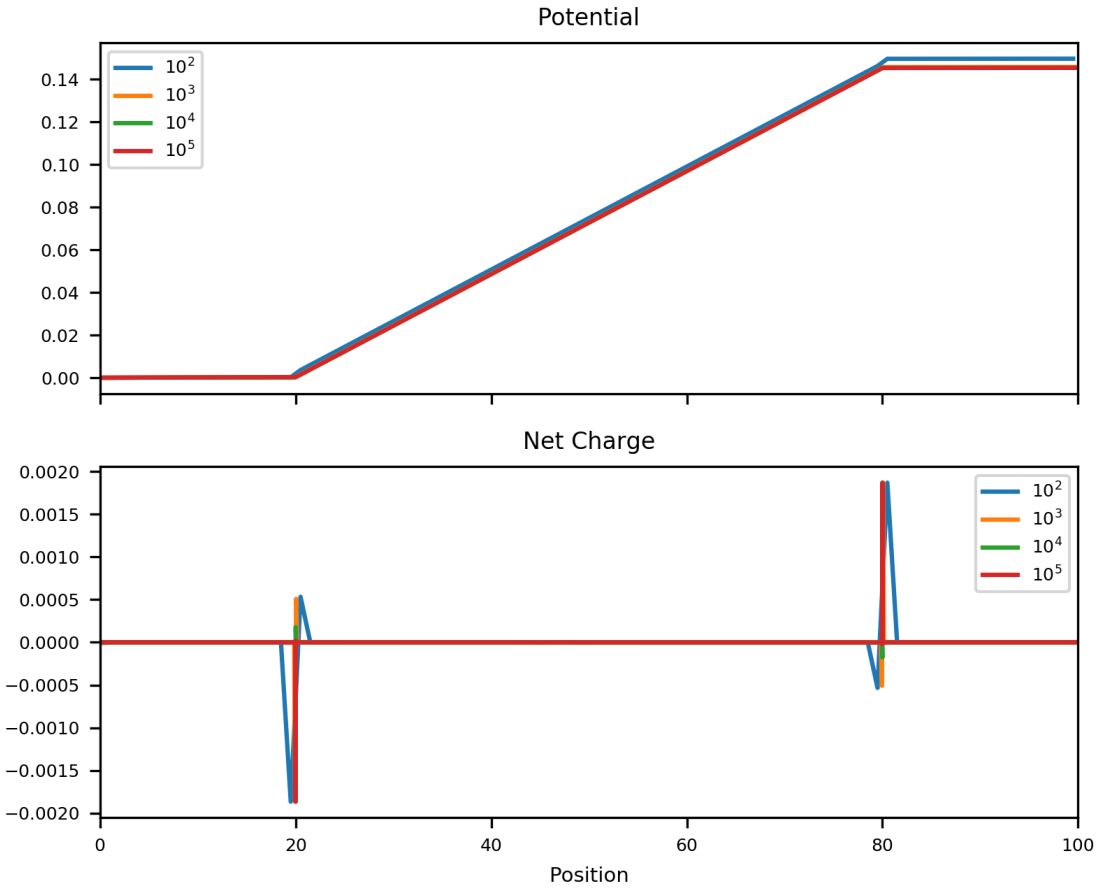

Fig. 8.3. Results from resolution test, (a) electrostatic potential, (b) net charge.

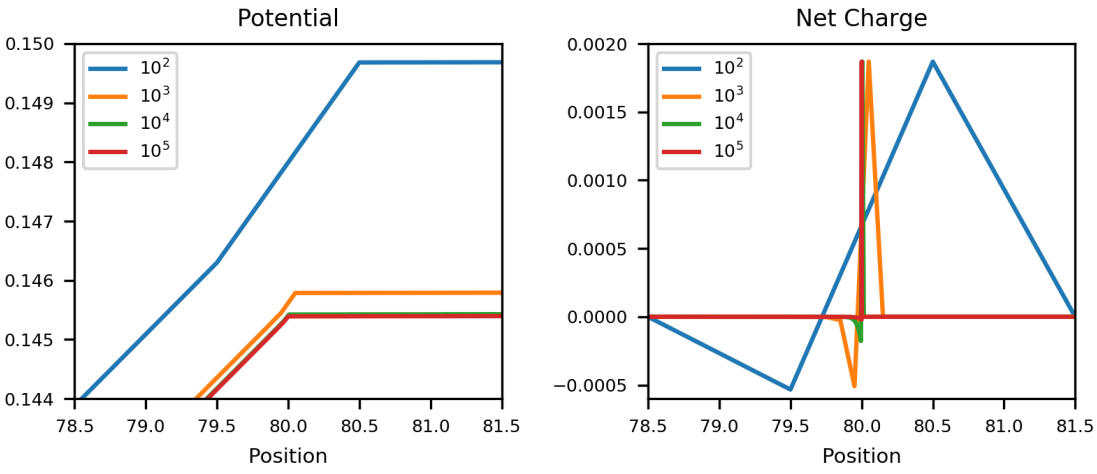

Fig. 8.4. Results from resolution test near interface, (a) electrostatic potential, (b) net charge.

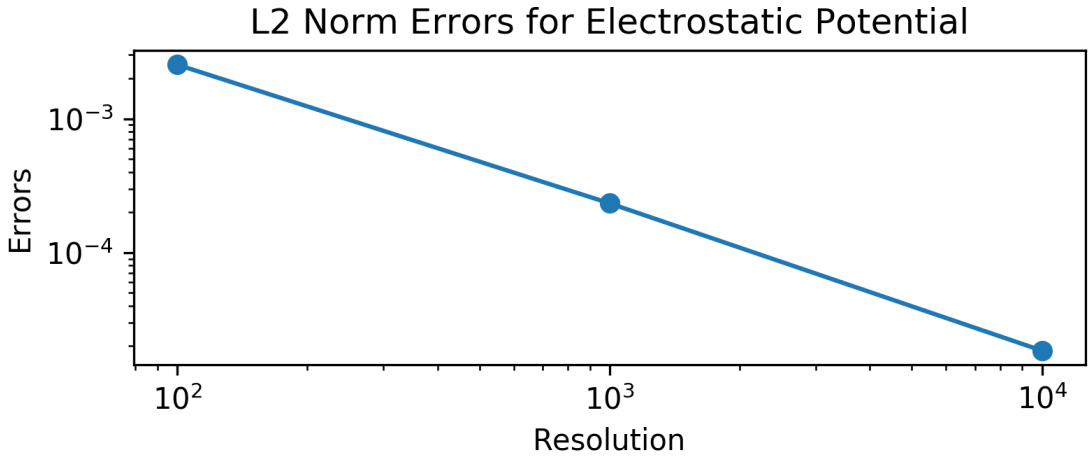

Fig. 8.5. Error results from resolution test.

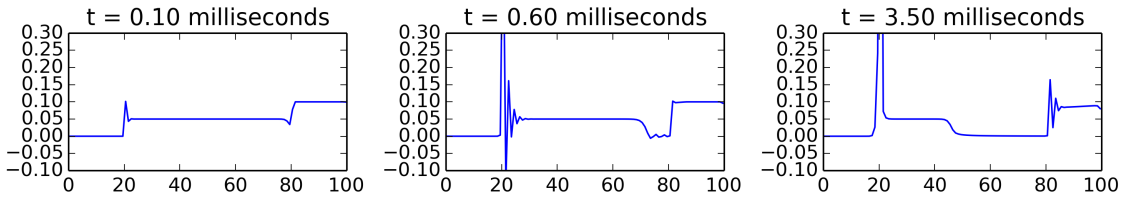

Fig. 8.6. Results of simulation without flux limiter, (a) initial oscillations at the anode/electrolyte interface, (b) oscillations initiated by transfer of positive ions at the cathode/electrolyte interface, (c) oscillations at cathode/electrolyte interface.

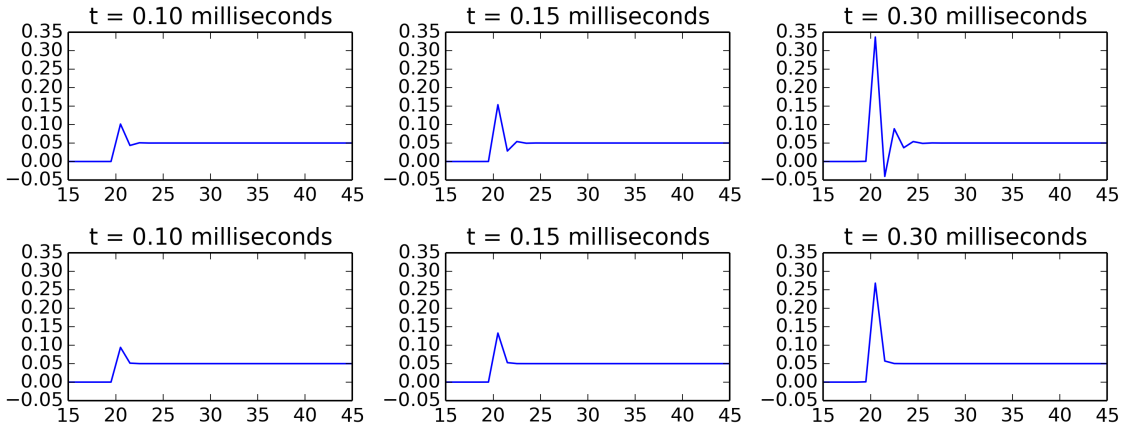

Fig. 8.7. Comparison between application of non-limiter and limiter based simulations at the anode/electrolyte interface.

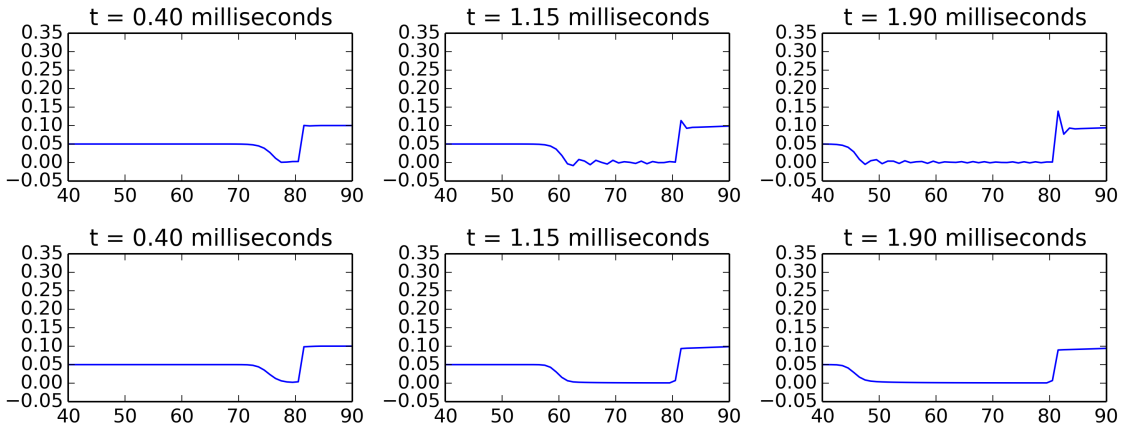

Fig. 8.8. Comparison between nonlimiter and limiter version in the electrolyte.

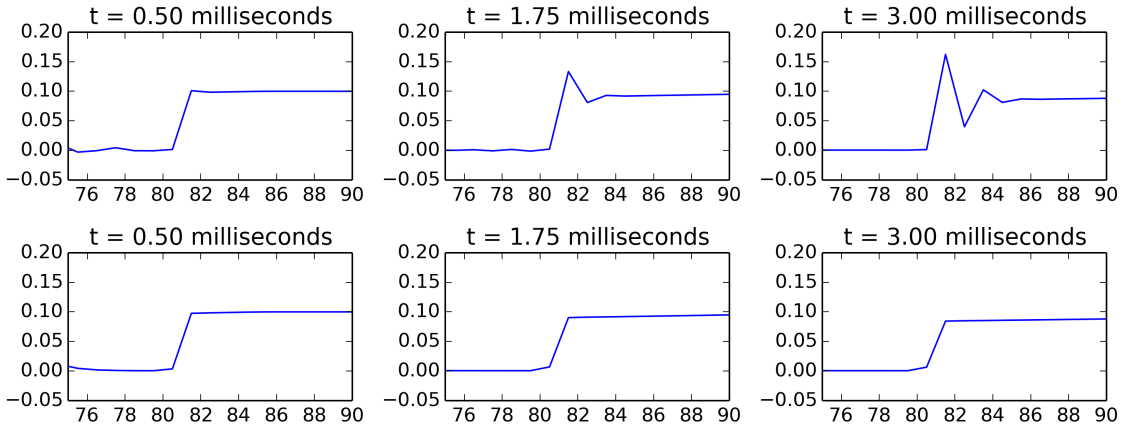

Fig. 8.9. Comparison between nonlimiter and limiter version at the cathode/electrolyte interface.

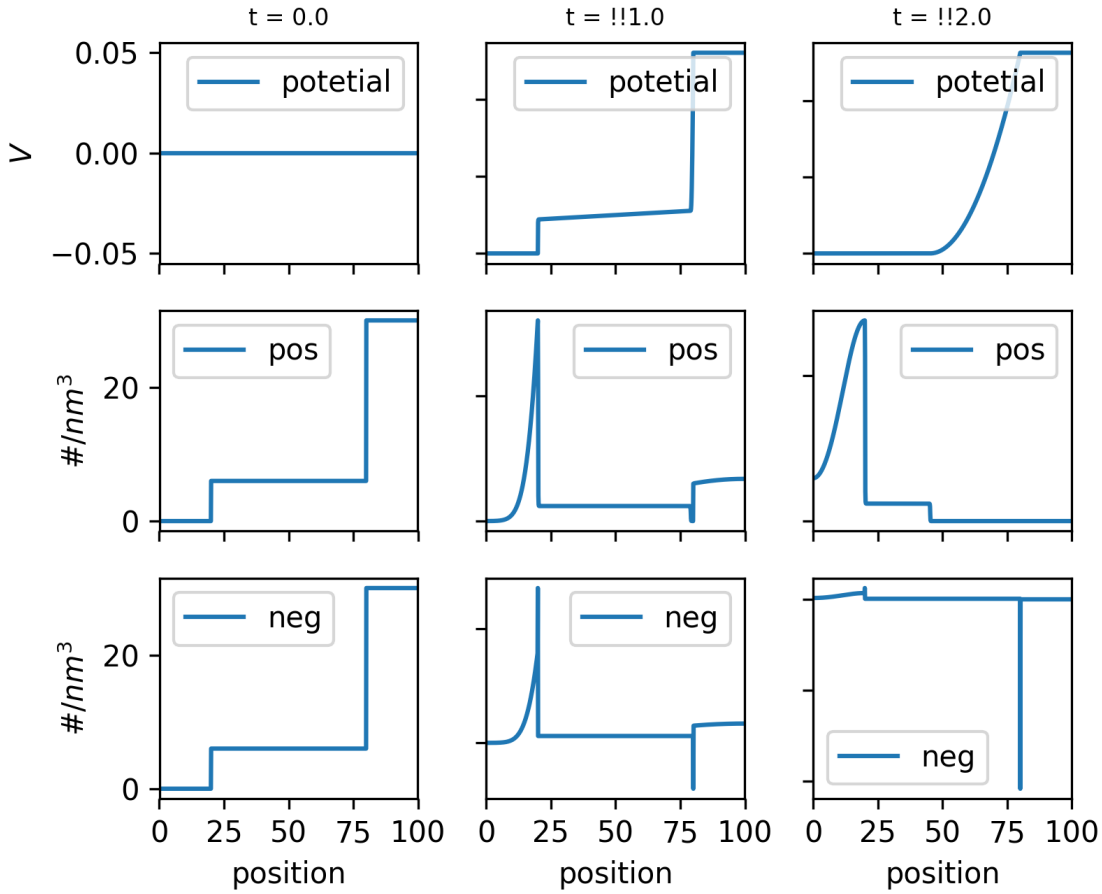

Fig. 8.10. Results from a full charge.

CHAPTER 9

CONCLUSIONS AND FUTURE WORK

Batteries range in size and form factor, and they can be found in the smallest portable devices to large EV vehicles. One does not need to look far to find a device that uses a battery of some type, and thus batteries play an important role in civilian society. Batteries also play an important role in our national defense. As was noted in the introduction, the current warfighter uses a variety of mission critical electronic devices that all depend on some type of battery. From these observations it can easily be seen that there is need for batteries that are lighter, more efficient, and resilient.

Computational tools can aid researcher and engineers in the development of novel battery designs that meet the demands of both modern warfighters and citizens. In this dissertation MPM is proposed as one of those tools. MPM has now been in use for more than 25 years [19, 18]. Since its introduction it has been the focus of continued research and has been applied to solving problems in fields ranging from geotechnical engineering to robotics. MPM has proven adept a modeling a range problems. Its strengths come from the hybrid nature of the method, which uses both particle and a mesh. By using a particle to model the domain, issues with mesh distortion are avoided, and by using a background mesh self-contact is resolved easily. The hybrid nature of MPM also has its own set of issues as a result of information being passed between particles and mesh.

The topics of research covered in this dissertation cover a variety of areas ranging from numerical methods, to computer science, to electrochemistry. The diversity in topics is a result of the cross-cutting nature of the program that funded this research. That said, the core of the research presented in this dissertation is directed toward one of the following key goals:

1. Find new insights into the stability and accuracy of MPM.
2. Develop methods to improve the stability and accuracy of MPM.

3. Seek insights and find methods for improving MPM's capabilites to model multiphysics problems, specifically applied to electrochemistry.
4. Develop a hybrid MPM/FVM approach to modeling a full electrochemical cell.

Not all of the goals have been achieved, but progress was made toward each of them. A summary of the contributions from this dissertation along with directions in future work are provided in the sections below.

9.1 Nullspace Filters and Corrected Derivatives for MPM

Chapter 4 discussed the issues that arise from the mismatch in dimensionality between particles and the background mesh. Generally speaking, the number of particles is greater than the number of nodes in MPM. This mismatch results in a nullspace in the mapping from nodes to particles. Because of this, noise can be introduced in the calculation as particle values are updated based on nodal values. One approach that has been presented is the use of an global SVD filter to remove the nullspace noise [124, 125], but solving of the SVD, especially at scale, is computationally expensive. Chapter 4 presented an alternative to the SVD approach by approximating the nullspace filter by remapping particle values to the nodes and then back to the particles. This approach was shown in the demonstrated numerical experiments to reduce numerical nullspace noise.

Another technique that was proposed in Chapter 4 was the use of corrective derivatives when mapping particle values to the nodes. The use of corrective derivatives was originally developed for particle methods [13, 114, 115, 127] and has been adapted to MPM. In initial numerical experiments it has been shown that the use of corrected derivatives improves the accuracy of MPM.

9.2 Time Integration Methods for Multiscale Problems Using MPM

Chapter 5 presented the problems that can arise when modeling multiphysics problems that have different physical processes that evolving at different timescales. For an explicit approach the stability criteria for one of the physical process becomes the limiting factor for the other. When solving problems of this type, one solution is to use either a fully implicit or semi-implicit time integration scheme. Doing so allows for much larger time

steps without causing numerical instabilities. One possible drawback with this approach is that even though the solution may be stable for larger step sizes the accuracy is reduced. The question then is, what is the reduction in accuracy and is it acceptable?

In Chapter 5 an initial survey of different explicit time integrations schemes for MPM was presented for context, followed by a description of the implicit MPM approach proposed by Guilkey and Weiss [146]. A disscussion on the formation of the Jacobian, used in the solving of implicit equations, using both an analytic and numerical approach is presented.

The final portion of the chapter presented a analyis of the different local errors in the implicit scheme. The analysis resulted in local error estimates for each of the nodal and particle value updates. The local error estimates can be used to determine if a timestep size is sufficiently accurate as compared to using a stability-only approach for timestep selection.

9.3 Diffusion Modeling in MPM

Stability constraints were examined for REFMPM using Von Neumann analyis. Generally, MPM is not an ideal candidate for an analysis of this type because of the movement of the particle in relation to the background mesh, but given that REFMPM is done in the reference configuration it lends itself to Von Neumann, given the proper assumptions.

Diffusion in the reference configuration using REFMPM was also presented along with an analysis of the methods stability contraints. It was shown that stability for diffusion using REFMPM is not simply function of conductivity/diffusivity and grid spacing but also the particle deformation gradients.

A discussion was also presented on the nullspace noise that occurs in diffusion calculations in REFMPM along with approaches to mitigate the introduced noise.

9.4 Using the Material Point Method to Model Chemical/Mechanical Coupling in the Deformation of a Silicon Anode

Chapter 7 discusses the use of MPM to model a silicon electrode undergoing multiple charge/discharge cycles. Silicon as an electrode material has very favorable characteristics because of its ability to absorb a large number lithium ions relative to quantity of host material at full charge. At full litiation a silicon electrode can increase its volume by $\sim$ 280% [20]. This change in volume leads to large deformations, which are both plastic and

elastic, for which MPM is well suited.

A description of the fully coupled chemical-mechanical model was presented, along with a description of the details regarding the method used for modeling diffusion in MPM, the flux boundary conditions, and validation tests for MPM diffusion. Multiple simulations were run to explore different constitutive models, boundary conditions, and parameters. The results from the runs were then presented.

Based on what was learned from the different experimental runs, a full simulation was run that modeled the coupled behavior of chemical diffusion and mechanical deformation. The results of the simulation demonstrated what a silicon electrode undergoes when it experiences multiple charge/discharge cycles and the resultant plastic deformations that occurs from the large changes in volume. This chapter also demonstrates the value that MPM can bring to the electrochemical community as tool for modeling different electrochemistry problems.

9.5 Modeling a Full Cell

Chapter 8 presented a full one dimensional electrochemical cell model based on the Poisson-Nernst-Plank equation, which used FVM to solve the model equations. A novel electrode/electrolyte interface model was also introduced. The basic design of the model was to prevent the flow of ions across the interface until an overpotential threshold was met. Once the threshold was met ions were allowed to flow across the interface. The sudden flow of ions across the interface resulted in numerical instabilities. The approach to resolve the instabilities was to use a flux limiter.

A full description of the one dimensional FVM implementation is provided along with analysis of the time integration method used in the model. Results from the model simulation runs are also provided along with a comparison between the unstable and the stable results that came as result of using the flux limiter. The model presented in Chapter 8 is a first-order approximation of the physics involved of a model of this type. A more complete model will need to incorporate Butler-Volmer kinetics at the electrode/electrolyte interface.

9.6 Future Work

9.6.1 Scaling the Use of Nullspace Filter and Corrected Derivatives

Chapter 4 presented the use of local nullspace filters and corrected derivatives as an improvement to MPM in terms of stability and accuracy. All the work that was presented was done on a one dimensional grid using linear basis functions. For future work the nullspace filtering techniques and the use of corrective derivatives need to be extend to multidimensional problems, using different basis functions, and run at scale. Another area of future research would be to examine if these techniques can be used to mitigate instabilities that arise from volumetric locking.

9.6.2 Adaptive Time Integration for Implicit MPM

Chapter 5 presented local error estimates that can be used to determining the accuracy of a given time step size. Future work that is of interest is to explore the use of local error estimates in an adaptive time integration scheme for implicit MPM. To the best of our knowledge adaptive time integrations methods have been proposed for explicit MPM [162, 163], but they are based on stability constraints and not accuracy.

9.6.3 Modeling a Full Electrochemical Cell Using a Hybrid MPM/FVM Approach

MPM was used to model the full cycle charge/discharge cycle of a silicon electrode. As was noted in Section 2.4.4 electrostatics is an important factor in modeling the diffusion ions within the electrode. As was demonstrated in Chapter 7 the diffusion portion of the model accounted only for chemical diffusion as a driving mechanism for the movement of ions through the electrode material. A topic of future research is to incorporate the affects of electrostatis on the movement of ions through the electrode material. To this end an implicit FVM solver has already been implemented in Uintah for solving the electrostatic equation.

Another area of future research would be to implement fracture in to the model and code. Research has shown that after multiple charge/discharge cycles a silicon electrode will begin to fracture [223], which reduces the life of the electrode.

9.6.4 Modeling a Full Cell

As was noted previously, the work presented in Chapter 8 was preliminary in nature. While the model allowed for the testing and demonstration of the validity of the numerial approaches used, the model itself was a first order approximation of the actual physics. A direction of future work would be to incorporate Butler-Volmer kinetics into model. Ouermi [222] has already begun research in this direction.

9.7 Summary

To summaries, research presented in this disseration demonstrates the work put forward to gaining deeper insights into MPM with regards to stability and accuracy, seeking methods for improving the accuracy and stability of MPM, improving MPM's capability for modeling multiphysics problems as applied to the modeling electrochemisty problems, and developing a FVM/MPM approach for modeling a full electrochemical cell. As is noted above there is much work to do in achieving these goals.

REFERENCES

[1] R. Montoya and S. Ewing, "What percentage of electric cars in the U.S.?" *Edmunds*, Jan. 12, 2024. [Online]. Available: https://www.edmunds.com/electric-car/articles/percentage-of-electric-cars-in-us.html

[2] M. Cerullo, "Number of Americans using solar power expected to more than triple by 2030," *CBS News*, Apr. 22, 2021. [Online]. Available: https://www.cbsnews.com/news/solar-power-american-triple-2030/

[3] Solar Energy Industries Association, "U.S. solar market insights," 2023. [Online]. Available: https://www.seia.org/us-solar-market-insight

[4] ——, "Solar industry research data," 2023. [Online]. Available: https://www.seia.org/solar-industry-research-data

[5] D. Hambling, "The overloaded soldier: Why U.S. infantry now carry more weight than ever," *Popular Mechanics*, Dec. 26, 2018. [Online]. Available: https://www.popularmechanics.com/military/research/a25644619/soldier-weight/

[6] J. King, "The overweight infantryman," *Modern War Institute*, Jan. 10, 2017. [Online]. Available: https://mwi.usma.edu/the-overweight-infantryman/

[7] The Economist, "A plug for the battery," *The Economist*, Jan. 16, 2016. [Online]. Available: https://www.economist.com/leaders/2016/01/16/a-plug-for-the-battery

[8] M. T. M. Koper, A. Z. Weber, K. Chan, and J. Cheng, "Introduction: Computational electrochemistry," *Chem. Rev.*, vol. 122, no. 12, pp. 10579–10580, Jun. 2022.

[9] S. Maiti, M. T. Curnan, K. Maiti, S. Choung, and J. W. Han, "Accelerating Li-based battery design by computationally engineering materials," *Chem*, vol. 9, no. 12, pp. 3415–3460, Dec. 2023.

[10] K. Smith *et al.*, "Computational design of batteries from materials to systems," National Renewable Energy Lab, Golden, CO, USA, Tech Rep. No. NREL/PR-5400-68770, Jun. 2017.

[11] C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer Simulation.* Boca Raton, FL, USA: CRC Press, 1991.

[12] R. A. Gingold and J. J. Monaghan, "Smoothed particle hydrodynamics: Theory and application to non-spherical stars," *Monthly Not. R. Astron. Soc.*, vol. 181, no. 3, pp. 375–389, Dec. 1977.

[13] W. K. Liu, S. Jun, S. Li, J. Adee, and T. Belytschko, "Reproducing kernel particle methods for structural dynamics," *Int. J. Numer. Methods Eng.*, vol. 38, no. 10, pp. 1655–1679, May 1995.

[14] F. H. Harlow, "The particle-in-cell computing method for fluid dynamics," *Methods Comput. Phys.*, vol. 3, pp. 319–343, 1964.

[15] J. U. Brackbill and H. M. Ruppel, "FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions," *J. Comput. Phys.*, vol. 65, no. 2, pp. 314–343, Aug. 1986.

[16] D. Sulsky, Z. Chen, and H. L. Schreyer, "A particle method for history-dependent materials," *Comput. Methods Appl. Mech. Eng.*, vol. 118, no. 1, pp. 179–196, Sep. 1994.

[17] D. Sulsky, S.-J. Zhou, and H. L. Schreyer, "Application of a particle-in-cell method to solid mechanics," *Comput. Phys. Commun.*, vol. 87, no. 1, pp. 236–252, May 1995.

[18] W. T. Solowski *et al.*, "Material point method: Overview and challenges ahead," in *Advances in Applied Mechanics*, S. P. A. Bordas and D. S. Balint, Eds. Cambridge, MA, USA: Elsevier, 2021, vol. 54, pp. 113–204.

[19] A. de Vaucorbeil, V. P. Nguyen, S. Sinaie, and J. Y. Wu, "Material point method after 25 years: Theory, implementation, and applications," in *Advances in Applied Mechanics*, S. P. A. Bordas and D. S. Balint, Eds. Cambridge, MA, USA: Elsevier, 2020, vol. 53, pp. 185–398.

[20] J. W. Wang *et al.*, "Two-phase electrochemical lithiation in amorphous silicon," *Nano Lett.*, vol. 13, no. 2, pp. 709–715, Jan. 2013.

[21] M. Steffen, R. M. Kirby, and M. Berzins, "Analysis and reduction of quadrature errors in the material point method (MPM)," *Int. J. Numer. Methods Eng.*, vol. 76, no. 6, pp. 922–948, Apr. 2008.

[22] P. de Koster, R. Tielen, E. Wobbes, and M. Möller, "Extension of B-spline material point method for unstructured triangular grids using Powell–Sabin splines," *Comput. Part. Mech.*, vol. 8, p. 273–288, Mar. 2021.

[23] W. Chen and M. Li, "Coarsened C1-continuous shape functions using B-spline patches for simulating nonlinear heterogeneous structures," *Comput. Methods Appl. Mech. Eng.*, vol. 421, Art. no. 116808, Mar. 2024.

[24] A. Sadeghirad, "B-spline convected particle domain interpolation method," *Eng. Anal. Bound. Elem.*, vol. 160, pp. 106–133, Mar. 2024.

[25] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs, "Isogeometric analysis: CAD, finite elements, nurbs, exact geometry and mesh refinement," *Comput. Methods Appl. Mech. Eng.*, vol. 194, no. 39, pp. 4135–4195, Oct. 2005.

[26] G. Moutsanidis, C. C. Long, and Y. Bazilevs, "IGA-MPM: The isogeometric material point method," *Comput. Methods Appl. Mech. Eng.*, vol. 372, Art. no. 113346, Dec. 2020.

[27] W. M. Coombs, T. J. Charlton, M. Cortis, and C. E. Augarde, "Overcoming volumetric locking in material point methods," *Comput. Methods Appl. Mech. Eng.*, vol. 333, pp. 1–21, May 2018.

[28] L. Wang *et al.*, "An efficient and locking-free material point method for three-dimensional analysis with simplex elements," *Int. J. Numer. Methods Eng.*, vol. 122, no. 15, pp. 3876–3899, Apr. 2021.

[29] M. Xie, P. Navas, and S. López-Querol, "An implicit locking-free B-spline material point method for large strain geotechnical modelling," *Int. J. Numer. Anal. Methods Geomech.*, vol. 47, no. 15, pp. 2741–2761, Jul. 2023.

[30] A. Di Perna, S. Cuomo, and M. Martinelli, "Modelling of landslide-structure interaction (LSI) through material point method (MPM)," in *Progress in Landslide Research and Technology*, I. Alcántara-Ayala *et al.*, Eds. Cham, Switzerland: Springer Nature, 2023, vol. 2, no. 1, pp. 159–178.

[31] K.-L. Zhao, L. chao Qiu, and Y. Liu, "Two-phase MPM modeling of dry granular fronts and watery tails formed in debris flows," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 128, Art. no. 107624, Jan. 2024.

[32] A. Troncone, L. Pugliese, A. Parise, and E. Conte, "Back-analysis of the post-failure stage of a landslide in sensitive clays," in *Geotechnical Engineering in the Digital and Technological Innovation Era*, A. Ferrari, M. Rosone, M. Ziccarelli, and G. Gottardi, Eds. Cham, Switzerland: Springer Nature, 2023, pp. 561–568.

[33] J. Fern, A. Rohe, K. Soga, and E. Alonso, *The Material Point Method for Geotechnical Engineering: A Practical Guide*, 1st ed. Boca Raton, FL, USA: CRC Press, 2019.

[34] M. E. Yildizdag, "Numerical modeling and simulation of material extrusion-based 3-D printing processes with a material point method framework," *Continuum Mech. Thermodyn.*, Dec. 2023, advance online publication.

[35] Y. Lian, J. Chen, M.-J. Li, and R. Gao, "A multi-physics material point method for thermo-fluid-solid coupling problems in metal additive manufacturing processes," *Comput. Methods Appl. Mech. Eng.*, vol. 416, Art. no. 116297, Nov. 2023.

[36] G. Bianchi, L. Lanzetti, D. Mariana, and S. Cinquemani, "Bioinspired design and experimental validation of an aquatic snake robot," *Biomimetics*, vol. 9, Art. no. 87, no. 2, Feb. 2024.

[37] J. Davy, P. Lloyd, J. H. Chandler, and P. Valdastri, "A framework for simulation of magnetic soft robots using the material point method," *IEEE Robot. Automat. Lett.*, vol. 8, no. 6, pp. 3470–3477, Apr. 2023.

[38] F. Cochevelou, D. Bonner, and M.-P. Schmidt, "Differentiable soft-robot generation," in *Proc. Genetic Evol. Comput. Conf.* New York, NY, USA: Association for Computing Machinery, Lisbon, Portugal, Jul. 15–19, 2023, p. 129–137.

[39] X. Zhang, Z. Chen, and Y. Liu, "Introduction," in *The Material Point Method: A Continuum-Based Particle Method for Extreme Loading Cases.* Oxford, UK: Academic Press, 2017, pp. 1–9.

[40] S. B. Phu Nguyen, Alban de Vaucorbeil, *The Material Point Method: Theory, Implementations and Applications*, 1st ed. Cham, Switzerland: Springer, 2023.

[41] A. J. Bard and L. R. Faulkner, *Electrochemical Methods: Fundamentals and Applications*, 2nd ed. New York, NY, USA: Wiley, 2000.

[42] J. Newman and K. E. Thomas-Alyea, *Electrochemical Systems*, 3rd ed. New York, NY, USA: John Wiley & Sons, 2004.

[43] V. S. Bagotsky, *Fundamentals of Electrochemistry*. New York, NY, USA: Wiley, 2005.

[44] R. O'Hayre, S.-W. Cha, W. Colella, and F. B. Prinz, *Fuel Cell Fundamentals*, 3rd ed. New York, NY, USA: John Wiley & Sons, 2016.

[45] P. Selinis and F. Farmakis, "A review on the anode and cathode materials for lithium-ion batteries with improved subzero temperature performance," *J. Electrochem. Soc.*, vol. 169, no. 1, Art. no. 010526, Jan. 2022.

[46] M. H. Hossain, M. A. Chowdhury, N. Hossain, M. A. Islam, and M. H. Mobarak, "Advances of lithium-ion batteries anode materials—A review," *Chem. Eng. J. Adv.*, vol. 16, Art. no. 100569, Nov. 2023.

[47] N. Nitta, F. Wu, J. T. Lee, and G. Yushin, "Li-ion battery materials: Present and future," *Mater. Today*, vol. 18, no. 5, pp. 252–264, Jun. 2015.

[48] P. U. Nzereogu, A. D. Omah, F. I. Ezema, E. I. Iwuoha, and A. C. Nwanya, "Anode materials for lithium-ion batteries: A review," *Appl. Surface Sci. Adv.*, vol. 9, Art. no. 100233, Jun. 2022.

[49] J. W. Gibbs, "On the equilibrium of heterogeneous substances," *Am. J. Sci.*, vol. 3S, 16, no. 96, pp. 441–458, 1878.

[50] S. J. Gill, "The chemical potential," *J. Chem. Educ.*, vol. 39, no. 10, Art. no. 506, Oct. 1962.

[51] J. C. M. Li, R. A. Oriani, and L. S. Darken, "The thermodynamics of stressed solids," *Z. Phys. Chem.*, vol. 49, pp. 271–290, Aug. 1966.

[52] F. C. Larché and J. W. Cahn, "A linear theory of thermochemical equilibrium of solids under stress," *Acta Metall.*, vol. 21, no. 8, pp. 1051–1063, Aug. 1973.

[53] ——, "A nonlinear theory of thermochemical equilibrium of solids under stress," *Acta Metall.*, vol. 26, no. 1, pp. 53–60, Jan. 1978.

[54] ——, "The effect of self-stress on diffusion in solids," *Acta Metall.*, vol. 30, no. 10, pp. 1835–1845, Oct. 1982.

[55] ——, "Overview no. 41 the interactions of composition and stress in crystalline solids," *Acta Metall.*, vol. 33, no. 3, pp. 331–357, Mar. 1985.

[56] V. A. Sethuraman, V. Srinivasan, A. F. Bower, and P. R. Guduru, "In situ measurements of stress-potential coupling in lithiated silicon," *J. Electrochem. Soc.*, vol. 157, no. 11, pp. A1253–A1261, Sep. 2010.

[57] Y.-F. Gao and M. Zhou, "Strong dependency of lithium diffusion on mechanical constraints in high-capacity Li-ion battery electrodes," *Acta Mech. Sin.*, vol. 28, no. 4, pp. 1068–1077, Aug. 2012.

[58] G. A. Holzapfel, *Nonlinear Solid Mechanics: A Continuum Approach for Engineers*. New York, NY, USA: Wiley, 2000.

[59] A. Einstein, "Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen," *Ann. Phys.*, vol. 322, no. 8, pp. 549–560, May 1905.

[60] D. J. Griffiths, *Introduction to Electrodynamics*, 5th ed. Cambridge, UK: Cambridge University Press, 2023.

[61] L. Anand, "A Cahn–Hilliard-type theory for species diffusion coupled with large elastic–plastic deformations," *J. Mech. Phys. Solids*, vol. 60, no. 12, pp. 1983–2002, Dec. 2012.

[62] G. Bucci, S. P. V. Nadimpalli, V. A. Sethuraman, A. F. Bower, and P. R. Guduru, "Measurement and modeling of the mechanical and electrochemical response of amorphous Si thin film electrodes during cyclic lithiation," *J. Mech. Phys. Solids*, vol. 62, pp. 276–294, Jan. 2014.

[63] Z. Cui, F. Gao, and J. Qu, "A finite deformation stress-dependent chemical potential and its applications to lithium ion batteries," *J. Mech. Phys. Solids*, vol. 60, no. 7, pp. 1280–1295, Jul. 2012.

[64] A. D. Drozdov, "Viscoplastic response of electrode particles in Li-ion batteries driven by insertion of lithium," *Int. J. Solids Struct.*, vol. 51, no. 3, pp. 690–705, Feb. 2014.

[65] A. F. Bower, E. Chason, P. R. Guduru, and B. W. Sheldon, "A continuum model of deformation, transport and irreversible changes in atomic structure in amorphous lithium–silicon electrodes," *Acta Mater.*, vol. 98, pp. 229–241, Oct. 2015.

[66] A. F. Bower, P. R. Guduru, and V. A. Sethuraman, "A finite strain model of stress, diffusion, plastic flow, and electrochemical reactions in a lithium-ion half-cell," *J. Mech. Phys. Solids*, vol. 59, no. 4, pp. 804–828, Apr. 2011.

[67] S. Prussin, "Generation and distribution of dislocations by solute diffusion," *J. Appl. Phys.*, vol. 32, no. 10, pp. 1876–1881, Dec. 1961.

[68] S. Timoshenko and J. N. Goodier, *Theory of Elasticity*. New York, NY, USA: McGraw-Hill, 1969.

[69] G. B. Stephenson, "Deformation during interdiffusion," *Acta Metall.*, vol. 36, no. 10, pp. 2663–2683, Oct. 1988.

[70] J. L. Chaboche, "A review of some plasticity and viscoplasticity constitutive theories," *Int. J. Plast.*, vol. 24, no. 10, pp. 1642–1693, Oct. 2008.

[71] J. Christensen and J. Newman, "Stress generation and fracture in lithium insertion materials," *J. Solid State Electrochem.*, vol. 10, no. 5, p. 293–319, Mar. 2006.

[72] E. H. Lee, "Some comments on elastic-plastic analysis," *Int. J. Solids Struct.*, vol. 17, no. 9, pp. 859–872, Aug. 1981.

[73] V. A. Lubarda, "Constitutive theories based on the multiplicative decomposition of deformation gradient: Thermoelasticity, elastoplasticity, and biomechanics," *Appl. Mech. Rev.*, vol. 57, no. 2, pp. 95–108, Mar. 2004.

[74] M. W. Evans and F. H. Harlow, "A machine calculation method for hydrodynamics problems," Los Alamos Scientific Laboratory, Los Alamos, NM, USA, Tech. Rep., Nov. 1955.

[75] F. H. Harlow, "Fluid dynamics in group T-3 Los Alamos National Laboratory: (LA-UR-03-3852)," *J. Comput. Phys.*, vol. 195, no. 2, pp. 414–433, Apr. 2004.

[76] M. W. Evans, F. H. Harlow, and D. E. H. Jr., "The particle-in-cell method for two-dimensional hydrodynamic problems," Los Alamos Scientific Laboratory, Los Alamos, NM, USA, Tech. Rep. No. LAMS-2082, Dec. 1956.

[77] M. W. Evans and F. H. Harlow, "The particle-in-cell method for hydrodynamic calculations," Los Alamos Scientific Laboratory, Los Alamos, NM, USA, Tech. Rep. No. LA-2139, Nov. 1957.

[78] O. Buneman, "Dissipation of currents in ionized media," *Phys. Rev.*, vol. 115, pp. 503–517, Aug. 1959.

[79] J. Dawson, "One-dimensional plasma model," *Phys. Fluids*, vol. 5, no. 4, pp. 445–459, Apr. 1962.

[80] F. H. Harlow, "The particle-in-cell method for numerical solution of problems in fluid dynamics," Los Alamos Scientific Laboratory, Los Alamos, NM, USA, Tech. Rep. No. LADC-5288, Dec. 1962.

[81] B. Wang, "A particle-in-cell method with adaptive phase-space remapping for kinetic plasmas," Ph.D. dissertation, Dept. Comput. Sci., Duke Univ., Durham, NC, USA, 2011.

[82] R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles*. Boca Raton, FL, USA: CRC Press, 1988.

[83] J. U. Brackbill, D. B. Kothe, and H. M. Ruppel, "FLIP: A low-dissipation, particle-in-cell method for fluid flow," *Comput. Phys. Commun.*, vol. 48, no. 1, pp. 25–38, Jan. 1988.

[84] Y. Zhu and R. Bridson, "Animating sand as a fluid," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 965–972, Jul. 2005.

[85] Z. Chen and R. M. Brannon, "An evaluation of the material point method," Sandia Laboratory, Albuquerque, NM, USA, Tech. Rep. No. SAND2002-0482, Feb. 2002.

[86] A. Sadeghirad, R. M. Brannon, and J. Burghardt, "A convected particle domain interpolation technique to extend applicability of the material point method for problems involving massive deformations," *Int. J. Numer. Methods Eng.*, vol. 86, no. 12, pp. 1435–1456, Jun. 2011.

[87] S. G. Bardenhagen and E. Kober, "The generalized interpolation material point method," *Comput. Model. Eng. Sci.*, vol. 5, no. 6, pp. 477–495, 2004.

[88] A. Sadeghirad, R. M. Brannon, and J. E. Guilkey, "Second-order convected particle domain interpolation (CPDI2) with enrichment for weak discontinuities at material interfaces," *Int. J. Numer. Methods Eng.*, vol. 95, no. 11, pp. 928–952, May 2013.

[89] M. Steffen, P. C. Wallstedt, J. E. Guilkey, R. M. Kirby, and M. Berzins, "Examination and analysis of implementation choices within the Material Point Method (MPM)," *Comput. Model. Eng. Sci.*, vol. 31, no. 2, pp. 107–127, 2008.

[90] A. Stomakhin, C. Schroeder, C. Jiang, L. Chai, J. Teran, and A. Selle, "Augmented MPM for phase-change and varied materials," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 138:1–138:11, Jul. 2014.

[91] Y. Gan, Z. Sun, Z. Chen, X. Zhang, and Y. Liu, "Enhancement of the material point method using B-spline basis functions," *Int. J. Numer. Methods Eng.*, vol. 113, no. 3, pp. 411–431, Jan. 2018.

[92] P. C. Wallstedt and J. E. Guilkey, "A weighted least squares particle-in-cell method for solid mechanics," *Int. J. Numer. Methods Eng.*, vol. 85, no. 13, pp. 1687–1704, Apr. 2011.

[93] E. Edwards and R. Bridson, "A high-order accurate particle-in-cell method," *Int. J. Numer. Methods Eng.*, vol. 90, no. 9, pp. 1073–1088, Jun. 2012.

[94] Y. Hu, Y. Fang, Z. Ge, Z. Qu, Y. Zhu, A. Pradhana, and C. Jiang, "A moving least squares material point method with displacement discontinuity and two-way rigid body coupling," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–14, Jul. 2018.

[95] Q. A. Tran, W. Solowski, M. Berzins, and J. Guilkey, "A convected particle least square interpolation material point method," *Int. J. Numer. Methods Eng.*, vol. 121, no. 6, pp. 1068–1100, Oct. 2019.

[96] J. U. Song and H. G. Kim, "An improved material point method using moving least square shape functions," *Comput. Part. Mech.*, vol. 8, p. 751–766, Nov. 2021.

[97] P. C. Wallstedt and J. E. Guilkey, "Improved velocity projection for the material point method," *Comput. Model. Eng. Sci.*, vol. 19, no. 3, pp. 223–232, 2007.

[98] C. Jiang, C. Schroeder, A. Selle, J. Teran, and A. Stomakhin, "The affine particle-in-cell method," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 1–10, Jul. 2015.

[99] C. Jiang, C. Schroeder, and J. Teran, "An angular momentum conserving affine-particle-in-cell method," *J. Comput. Phys.*, vol. 338, pp. 137–164, Jun. 2017.

[100] D. Z. Zhang, X. Ma, and P. T. Giguere, "Material point method enhanced by modified gradient of shape function," *J. Comput. Phys.*, vol. 230, no. 16, pp. 6379–6398, Jul. 2011.

[101] A. Stomakhin, C. Schroeder, L. Chai, J. Teran, and A. Selle, "A material point method for snow simulation," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 102:1–102:10, Jul. 2013.

[102] A. B. Langdon, "Effects of the spatial grid in simulation plasmas," *J. Comput. Phys.*, vol. 6, no. 2, pp. 247–267, Oct. 1970.

[103] J. U. Brackbill, "The ringing instability in particle-in-cell calculations of low-speed flow," *J. Comput. Phys.*, vol. 75, no. 2, pp. 469–492, Apr. 1988.

[104] L. Chen, A. B. Langdon, and C. K. Birdsall, "Reduction of the grid effects in simulation plasmas," *J. Comput. Phys.*, vol. 14, no. 2, pp. 200–222, Feb. 1974.

[105] J. W. Brackbill and G. Lapenta, "A method to suppress the finite-grid instability in plasma simulations," *J. Comput. Phys.*, vol. 114, no. 1, pp. 77–84, Sep. 1994.

[106] M. Ortiz, "A note on energy conservation and stability of nonlinear time-stepping algorithms," *Comput. Struct.*, vol. 24, no. 1, pp. 167–168, Dec. 1986.

[107] G. Lapenta, "Exactly energy conserving semi-implicit particle in cell formulation," *J. Comput. Phys.*, vol. 334, pp. 349–366, Apr. 2017.

[108] J. U. Brackbill, "On energy and momentum conservation in particle-in-cell plasma simulation," *J. Comput. Phys.*, vol. 317, pp. 405–427, Jul. 2016.

[109] S. G. Bardenhagen, "Energy conservation error in the material point method for solid mechanics," *J. Comput. Phys.*, vol. 180, no. 1, pp. 383–403, Jan. 2002.

[110] E. Love and D. L. Sulsky, "An energy-consistent material-point method for dynamic finite deformation plasticity," *Int. J. Numer. Methods Eng.*, vol. 65, no. 10, pp. 1608–1638, Mar. 2006.

[111] ——, "An unconditionally stable, energy–momentum consistent implementation of the material-point method," *Comput. Methods Appl. Mech. Eng.*, vol. 195, no. 33, pp. 3903–3925, Jul. 2006.

[112] C. M. Mast, P. Mackenzie-Helnwein, P. Arduino, G. R. Miller, and W. Shin, "Mitigating kinematic locking in the material point method," *J. Comput. Phys.*, vol. 231, no. 16, pp. 5351–5373, Jun. 2012.

[113] P. C. Wallstedt and J. E. Guilkey, "An evaluation of explicit time integration schemes for use with the generalized interpolation material point method," *J. Comput. Phys.*, vol. 227, no. 22, pp. 9628–9642, Nov. 2008.

[114] G. R. Johnson and S. R. Beissel, "Normalized smoothing functions for SPH impact computations," *Int. J. Numer. Methods Eng.*, vol. 39, no. 16, pp. 2725–2741, Aug. 1996.

[115] P. W. Randles and L. D. Libersky, "Smoothed particle hydrodynamics: Some recent improvements and applications," *Comput. Methods Appl. Mech. Eng.*, vol. 139, no. 1, pp. 375–408, Dec. 1996.

[116] J. J. Monaghan, "SPH without a tensile instability," *J. Comput. Phys.*, vol. 159, no. 2, pp. 290–311, Apr. 2000.

[117] T. Belytschko, Y. Guo, W. Kam Liu, and S. Ping Xiao, "A unified stability analysis of meshless particle methods," *Int. J. Numer. Methods Eng.*, vol. 48, no. 9, pp. 1359–1400, Jun. 2000.

[118] T. Belytschko, Y. Krongauz, J. Dolbow, and C. Gerlach, "On the completeness of meshfree particle methods," *Int. J. Numer. Methods Eng.*, vol. 43, no. 5, pp. 785–819, Dec. 1998.

[119] T. Belytschko and S. Xiao, "Stability analysis of particle methods with corrected derivatives," *Comput. Math. Appl.*, vol. 43, no. 3, pp. 329–350, Feb. 2002.

[120] S. Bai and C. Schroeder, "Stability analysis of explicit MPM," *Comput. Graph. Forum*, vol. 41, no. 8, pp. 19–30, Mar. 2022.

[121] R. Ni and X. Zhang, "A precise critical time step formula for the explicit material point method," *Int. J. Numer. Methods Eng.*, vol. 121, no. 22, pp. 4989–5016, Nov. 2020.

[122] M. Berzins, "Nonlinear stability and time step selection for the MPM method," *Comput. Part. Mech.*, vol. 5, no. 4, pp. 455–466, Oct. 2018.

[123] R. Spigler and M. Vianello, "Convergence analysis of the semi-implicit Euler method for abstract evolution equations," *Numer. Funct. Anal. Optim.*, vol. 16, no. 5–6, pp. 785–803, May 1995.

[124] C. Gritton, "Ringing instability in particle methods," Master's thesis, Dept. Comput., University of Utah, Salt Lake City, UT, USA, 2014.

[125] C. Gritton, M. Berzins, and R. M. Kirby, "Improving accuracy in particle methods using null spaces and filters," in *Proc. 4th Int. Conf. Part. Based Methods Fund. Appl.* Barcelona, Spain: International Center for Numerical Methods in Engineering, Sep. 28–30, 2015, Barcelona, Spain, pp. 202–213.

[126] Y. Krongauz and T. Belytschko, "Consistent pseudo-derivatives in meshless methods," *Comput. Methods Appl. Mech. Eng.*, vol. 146, no. 3–4, pp. 371–386, Jul. 1997.

[127] G. A. Dilts, "Moving-least-squares-particle hydrodynamics—I. Consistency and stability," *Int. J. Numer. Methods Eng.*, vol. 44, no. 8, pp. 1115–1155, Feb. 1999.

[128] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed. Baltimore, MD, USA: Johns Hopkins University Press, 1996.

[129] L. N. Trefethen and D. Bau, III, *Numerical Linear Algebra.* Philadelpha, PA, USA: SIAM, 1997.

[130] P.-G. Martinsson and J. A. Tropp, "Randomized numerical linear algebra: Foundations and algorithms," *Acta Numer.*, vol. 29, p. 403–572, Nov. 2020.

[131] N. Halko, P. G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Rev.*, vol. 53, no. 2, pp. 217–288, May 2011.

[132] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control.* Cambridge, UK: Cambridge University Press, 2019.

[133] C. Gritton and M. Berzins, "Improving accuracy in the MPM method using a null space filter," *Comput. Part. Mech.*, pp. 131–142, Jan. 2017.

[134] C. C. Hammerquist and J. A. Nairn, "A new method for material point method particle updates that reduces noise and enhances stability," *Comput. Methods Appl. Mech. Eng.*, vol. 318, pp. 724–738, May 2017.

[135] J. A. Nairn and C. C. Hammerquist, "Material point method simulations using an approximate full mass matrix inverse," *Comput. Methods Appl. Mech. Eng.*, vol. 377, Art. no. 113667, Apr. 2021.

[136] Q.-A. Tran and W. Solowski, "Temporal and null-space filter for the material point method," *Int. J. Numer. Methods Eng.*, vol. 120, no. 3, pp. 328–360, Jun. 2019.

[137] G. A. Tritsaris, K. Zhao, O. U. Okeke, and E. Kaxiras, "Diffusion of lithium in bulk amorphous silicon: A theoretical study," *J. Phys. Chem. C*, vol. 116, no. 42, pp. 22212–22216, Sep. 2012.

[138] R. Ruffo, S. S. Hong, C. K. Chan, R. A. Huggins, and Y. Cui, "Impedance analysis of silicon nanowire lithium ion battery anodes," *J. Phys. Chem. C*, vol. 113, no. 26, pp. 11390–11398, Jun. 2009.

[139] J. Xie, N. Imanishi, T. Zhang, A. Hirano, Y. Takeda, and O. Yamamoto, "Li-ion diffusion in amorphous Si films prepared by RF magnetron sputtering: A comparison of using liquid and polymer electrolytes," *Mater. Chem. Phys.*, vol. 120, no. 2, pp. 421–425, Apr. 2010.

[140] L. E. Kinsler, , A. R. Frey, A. B. Coppens, and J. V. Sanders, *Fundamentals of Acoustics*. New York, NY, USA: Wiley, 2000.

[141] R. L. Burden and J. D. Faires, *Numerical Analysis*, 9th ed. Boston, MA, USA: Cengage Learning, 2009.

[142] M. Berzins, "Time integration errors and energy conservation properties of the stormer verlet method applied to MPM," in *Proc. Int. Conf. Part. Based Methods Fund. Appl.* Barcelona, Spain: International Center for Numerical Methods in Engineering, Oct. 28–30, 2019, Barcelona, Spain, pp. 555–566.

[143] T. F. Gast, C. Schroeder, A. Stomakhin, C. Jiang, and J. M. Teran, "Optimization integrator for large time steps," *IEEE Trans. Vis. Comput. Graph.*, vol. 21, no. 10, pp. 1103–1115, Oct. 2015.

[144] M. Berzins, "Energy conservation and accuracy of some MPM formulations," *Comput. Part. Mech.*, vol. 9, p. 1205–1217, Feb. 2022.

[145] S. J. Cummins and J. U. Brackbill, "An implicit particle-in-cell method for granular materials," *J. Comput. Phys.*, vol. 180, no. 2, pp. 506–548, Aug. 2002.

[146] J. E. Guilkey and J. A. Weiss, "Implicit time integration for the material point method: Quantitative and algorithmic comparisons with the finite element method," *Int. J. Numer. Methods Eng.*, vol. 57, no. 9, pp. 1323–1338, May 2003.

[147] D. Sulsky and A. Kaul, "Implicit dynamics in the material-point method," *Comput. Methods Appl. Mech. Eng.*, vol. 193, no. 12, pp. 1137–1170, Mar. 2004.

[148] B. Wang, P. J. Vardon, M. A. Hicks, and Z. Chen, "Development of an implicit material point method for geotechnical applications," *Comput. Geotechn.*, vol. 71, pp. 159–167, Jan. 2016.

[149] M. Berzins, "Symplectic time integration methods for the material point method, experiments, analysis and order reduction," in *14th World Congr. Comput. Mechan.* Lisbon, Portugal: European Community on Computational Methods in Applied Sciences, Jan. 11–15, 2021, virtual, pp. 1–12.

[150] D. Sulsky, H. Schreyer, K. Peterson, R. Kwok, and M. Coon, "Using the material-point method to model sea ice dynamics," *J. Geophys. Res. Oceans*, vol. 112, Art. no. C2, Feb. 2007.

[151] N. M. Newmark, "A method of computation for structural dynamics," *J. Eng. Mech. Division*, vol. 85, no. 3, pp. 67–94, Jul. 1959.

[152] A. C. Hindmarsh *et al.*, "SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers," *ACM Trans. Math. Softw.*, vol. 31, no. 3, pp. 363–396, Sep. 2005.

[153] C. Kelley, *Solving Nonlinear Equations with Newton's Method*. Philadelphia, PA, USA: SIAM, 2003.

[154] D. A. Knoll and D. E. Keyes, "Jacobian-free Newton–Krylov methods: A survey of approaches and applications," *J. Comput. Phys.*, vol. 193, no. 2, pp. 357–397, Jan. 2004.

[155] P. Brown and Y. Saad, "Hybrid Krylov methods for nonlinear systems of equations," *SIAM J. Sci. Statist. Comput.*, vol. 11, no. 3, pp. 450–481, Jul. 1990.

[156] P. Bogacki and L. F. Shampine, "A 3(2) pair of Runge - Kutta formulas," *Appl. Math. Lett.*, vol. 2, no. 4, pp. 321–325, 1989.

[157] L. F. Shampine, "Error estimation and control of ODEs," *J. Sci. Comput.*, vol. 25, pp. 3–16, Nov. 2005.

[158] G. M. Hulbert and I. Jang, "Automatic time step control algorithms for structural dynamics," *Comput. Methods Appl. Mech. Eng.*, vol. 126, no. 1–2, pp. 155–178, Sep. 1995.

[159] C.-K. Choi and H.-J. Chung, "Error estimates and adaptive time stepping for various direct time integration methods," *Comput. Struct.*, vol. 60, no. 6, pp. 923–944, Jul. 1996.

[160] I. Romero and L. M. Lacoma, "Analysis and improved methods for the error estimation of numerical solutions in solid and multibody dynamics," in *Multibody Dynamics: Computational Methods and Applications*, J. C. García Orden, J. M. Goicolea, and J. Cuadrado, Eds. Dordrecht, Netherlands: Springer, 2007, pp. 69–89.

[161] D. F. Rossi, W. G. Ferreira, W. J. Mansur, and A. F. G. Calenzani, "A review of automatic time-stepping strategies on numerical time integration for structural dynamics analysis," *Eng. Struct.*, vol. 80, pp. 118–136, Dec. 2014.

[162] Y. Fang, Y. Hu, S.-M. Hu, and C. Jiang, "A temporally adaptive material point method with regional time stepping," *Comput. Graph. Forum*, vol. 37, no. 8, pp. 195–204, Sep. 2018.

[163] X. Wang *et al.*, "Hierarchical optimization time integration for CFL-rate MPM stepping," *ACM Trans. Graph.*, vol. 39, Apr. 2020.

[164] J. A. Nairn and J. E. Guilkey, "Axisymmetric form of the generalized interpolation material point method," *Int. J. Numer. Methods Eng.*, vol. 101, no. 2, pp. 127–147, Jan. 2015.

[165] T. Fagan, V. Lemiale, J. Nairn, Y. Ahuja, R. Ibrahim, and Y. Estrin, "Detailed thermal and material flow analyses of friction stir forming using a three-dimensional particle based model," *J. Mater. Process. Technol.*, vol. 231, pp. 422–430, May 2016.

[166] X. Y. Gu, C. Y. Dong, T. Cheng, Y. Zhang, and Y. Bai, "The transient heat conduction MPM and GIMP applied to isotropic materials," *Eng. Anal. Bound. Elem.*, vol. 66, pp. 155–167, May 2016.

[167] J. A. Nairn, "Modeling heat flow across material interfaces and cracks using the material point method," *Comput. Part. Mech.*, vol. 6, no. 1, pp. 133–144, Jan. 2019.

[168] X. Lei, S. He, A. Abed, X. Chen, Z. Yang, and Y. Wu, "A generalized interpolation material point method for modelling coupled thermo-hydro-mechanical problems," *Comput. Methods Appl. Mech. Eng.*, vol. 386, Art. no. 114080, Dec. 2021.

[169] J. Yu, J. Zhao, W. Liang, and S. Zhao, "A semi-implicit material point method for coupled thermo-hydro-mechanical simulation of saturated porous media in large deformation," *Comput. Methods Appl. Mech. Eng.*, vol. 418, Art. no. 116462, Jan. 2024.

[170] D. V. Widder, *The Heat Equation*, 1st ed. Amsterdam, Netherlands: Elsevier, 1975.

[171] J. Tao, Y. Zheng, Z. Chen, and H. Zhang, "Generalized interpolation material point method for coupled thermo-mechanical processes," *Int. J. Mech. Mater. Des.*, vol. 12, p. 577–595, Feb. 2016.

[172] M. Steffen, R. M. Kirby, and M. Berzins, "Decoupling and balancing of space and time errors in the material point method (MPM)," *Int. J. Numer. Methods Eng.*, vol. 82, no. 10, pp. 1207–1243, Jun. 2010.

[173] A. de Vaucorbeil, V. P. Nguyen, and C. R. Hutchinson, "A total-Lagrangian material point method for solid mechanics problems involving large deformations," *Comput. Methods Appl. Mech. Eng.*, vol. 360, Art. no. 112783, 2020.

[174] J.-S. Chen, C. Pan, C.-T. Wu, and W. K. Liu, "Reproducing kernel particle methods for large deformation analysis of non-linear structures," *Comput. Methods Appl. Mech. Eng.*, vol. 139, no. 1, pp. 195–227, Dec. 1996.

[175] J. C. Rade Vignjevic, Juan R. Reveles, "SPH in a total lagrangian formalism," *Comput. Model. Eng. Sci.*, vol. 14, no. 3, pp. 181–198, 2006.

[176] J. Oliver and E. Onate, "A total Lagrangian finite element formulation for the geometrically nonlinear analysis of shells," in *Flexible Shells: Theory and Applications*, E. L. Axelrad and F. A. Emmerling, Eds. Berlin, Germany: Springer, 1984, pp. 241–256.

[177] G. Avancini and R. A. K. Sanches, "A total Lagrangian position-based finite element formulation for free-surface incompressible flows," *Finite Elem. Anal. Des.*, vol. 169, Art. no. 103348, Feb. 2020.

[178] K. P. Ruggirello and S. C. Schumacher, "A dynamic adaptation technique for the material point method," Sandia National Laboratory, Albuquerque, NM, USA, Tech. Rep. No. SAND2013-4361C, May 2013.

[179] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations*. Philadelphia, PA, USA: SIAM, 2007.

[180] J. A. Nairn, "Coupling transport equations to mechanics in the material point method using an approximate full capacity matrix inverse," *Comput. Methods Appl. Mech. Eng.*, vol. 420, Art. no. 116757, Feb. 2024.

[181] J. D. Logan, *Applied Partial Differential Equations*, 2nd ed. Cham, Switzerland: Springer, 2013.

[182] M. N. Obrovac and L. Christensen, "Structural changes in silicon anodes during lithium insertion/extraction," *Electrochem. Solid State Lett.*, vol. 7, no. 5, Art. no. A93, Mar. 2004.

[183] C. K. Chan *et al.*, "High-performance lithium battery anodes using silicon nanowires," *Nat. Nanotechnol.*, vol. 3, no. 1, pp. 31–35, Jan. 2008.

[184] J. M. Tarascon and M. Armand, "Issues and challenges facing rechargeable lithium batteries," *Nature*, vol. 414, no. 6861, pp. 359–367, Nov. 2001.

[185] S. Ohara, J. Suzuki, K. Sekine, and T. Takamura, "A thin film silicon anode for Li-ion batteries having a very large specific capacity and long cycle life," *J. Power Sources*, vol. 136, no. 2, pp. 303–306, Oct. 2004.

[186] E. Mills, J. Cannarella, Q. Zhang, S. Bhadra, C. B. Arnold, and S. Y. Chou, "Silicon nanopillar anodes for lithium-ion batteries using nanoimprint lithography with flexible molds," *J. Vac. Sci. Technol. B Nanotechnol. Microelectron.*, vol. 32, no. 6, Art. no. 06FG10, Nov. 2014.

[187] L. Baggetto, D. Danilov, and P. H. L. Notten, "Honeycomb-structured silicon: Remarkable morphological changes induced by electrochemical (de)lithiation," *Adv. Mater.*, vol. 23, no. 13, pp. 1563–1566, Feb. 2011.

[188] C. H. Wu, "The role of Eshelby stress in composition-generated and stress-assisted diffusion," *J. Mech. Phys. Solids*, vol. 49, no. 8, pp. 1771–1794, Aug. 2001.

[189] X. H. Liu, L. Zhong, S. Huang, S. X. Mao, T. Zhu, and J. Y. Huang, "Size-dependent fracture of silicon nanoparticles during lithiation," *ACS Nano*, vol. 6, no. 2, pp. 1522–1531, Jan. 2012.

[190] M. Wang, X. Xiao, and X. Huang, "Study of lithium diffusivity in amorphous silicon via finite element analysis," *J. Power Sources*, vol. 307, pp. 77–85, Mar. 2016.

[191] L. A. Berla, S. W. Lee, I. Ryu, Y. Cui, and W. D. Nix, "Robustness of amorphous silicon during the initial lithiation/delithiation cycle," *J. Power Sources*, vol. 258, pp. 253–259, Jul. 2014.

[192] Y. An and H. Jiang, "A finite element simulation on transient large deformation and mass diffusion in electrodes for lithium ion batteries," *Model. Simul. Mater. Sci. Eng.*, vol. 21, no. 7, Art. no. 074007, Oct. 2013.

[193] X. H. Liu *et al.*, "Anisotropic swelling and fracture of silicon nanowires during lithiation," *Nano Lett.*, vol. 11, no. 8, pp. 3312–3318, Jun. 2011.

[194] A. F. Bower and P. R. Guduru, "A simple finite element model of diffusion, finite deformation, plasticity and fracture in lithium ion insertion electrode materials," *Model. Simul. Mater. Sci. Eng.*, vol. 20, no. 4, Art. no. 045004, Apr. 2012.

[195] G. Irving, J. Teran, and R. Fedkiw, "Invertible finite elements for robust simulation of large deformation," in *Proc. 2004 ACM SIGGRAPH/Eurographics Symp. Comput. Animat.* Goslar, Germany: Eurographics Association, Grenoble, France, Aug. 27–29, 2004, pp. 131–140.

[196] M. Anahid and A. R. Khoei, "New development in extended finite element modeling of large elasto-plastic deformations," *Int. J. Numer. Methods Eng.*, vol. 75, no. 10, pp. 1133–1171, Jan. 2008.

[197] C. Peng, W. Wu, H.-s. Yu, and C. Wang, "A SPH approach for large deformation analysis with hypoplastic constitutive model," *Acta Geotechn.*, vol. 10, no. 6, pp. 703–717, Jun. 2015.

[198] J.-S. Chen, C. Pan, and C.-T. Wu, "Large deformation analysis of rubber based on a reproducing kernel particle method," *Comput. Mech.*, vol. 19, no. 3, pp. 211–227, Feb. 1997.

[199] S. G. Bardenhagen, A. D. Brydon, and J. E. Guilkey, "Insight into the physics of foam densification via numerical simulation," *J. Mech. Phys. Solids*, vol. 53, no. 3, pp. 597–617, Nov. 2005.

[200] Center for the Simulation of Accidental Fires and Explosions (C-SAFE), and Scientific Computing and Imaging Institute (SCI), University of Utah, "Uintah computational framework." [Online]. Available: http://uintah.utah.edu/

[201] J. Guilkey *et al.*, "Uintah user guide," Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT, USA, Tech. Rep. No. UUSCI-2009-007, Sep. 2009. [Online]. Available: http://uintah-build.sci.utah.edu/trac/wiki/Documentation/

[202] Y. F. Gao and M. Zhou, "Strong stress-enhanced diffusion in amorphous lithium alloy nanowire electrodes," *J. Appl. Phys.*, vol. 109, no. 1, Art. no. 014310, Jan. 2011.

[203] L. Brassart and Z. Suo, "Reactive flow in large-deformation electrodes of lithium-ion batteries," *Int. J. Appl. Mech.*, vol. 4, no. 3, Art. no. 1250023, Sep. 2012.

[204] R. M. Brannon, private communication, Feb. 2017.

[205] J. C. Simo and T. J. R. Hughes, *Computational Inelasticity*. New York, NY, USA: Springer-Verlag, 1998.

[206] H. Yang, F. Fan, W. Liang, X. Guo, T. Zhu, and S. Zhang, "A chemo-mechanical model of lithiation in silicon," *J. Mech. Phys. Solids*, vol. 70, pp. 349–361, Oct. 2014.

[207] S. Zhang, "Chemomechanical modeling of lithiation-induced failure in high-volume-change electrode materials for lithium ion batteries," *NPJ Comput. Mater.*, vol. 3, Art. no. 7, 2017.

[208] M. Doyle, T. F. Fuller, and J. Newman, "Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell," *J. Electrochem. Soc.*, vol. 140, no. 6, Art. no. 1526, pp. 1526–1533, May 1993.

[209] M. Doyle and J. Newman, "The use of mathematical modeling in the design of lithium/polymer battery systems," *Electrochim. Acta*, vol. 40, no. 13, pp. 2191–2196, Oct. 1995.

[210] D. Zhang, B. N. Popov, and R. E. White, "Modeling lithium intercalation of a single spinel particle under potentiodynamic control," *J. Electrochem. Soc.*, vol. 147, no. 3, pp. 831–838, Mar. 2000.

[211] A. Jokar, B. Rajabloo, M. Désilets, and M. Lacroix, "Review of simplified pseudo-two-dimensional models of lithium-ion batteries," *J. Power Sources*, vol. 327, pp. 44–55, Sep. 2016.

[212] M. Rossi, T. Wallmersperger, S. Neukamm, and K. Padberg-Gehle, "Modeling and simulation of electrochemical cells under applied voltage," *Electrochim. Acta*, vol. 258, pp. 241–254, Dec. 2017.

[213] T. Wallmersperger, D. J. Leo, and C. S. Kothera, "Transport modeling in ionomeric polymer transducers and its relationship to electromechanical coupling," *J. Appl. Phys.*, vol. 101, no. 2, Art. no. 042912, Jan. 2007.

[214] E. Martínez-Rosas, R. Vasquez-Medrano, and A. Flores-Tlacuahuac, "Modeling and simulation of lithium-ion batteries," *Comput. Chem. Eng.*, vol. 35, no. 9, pp. 1937–1948, Sep. 2011.

[215] M. Doyle, J. Newman, A. S. Gozdz, C. N. Schmutz, and J. Tarascon, "Comparison of modeling predictions with experimental data from plastic lithium ion cells," *J. Electrochem. Soc.*, vol. 143, no. 6, Art. no. 1890, Jun. 1996.

[216] M. Mirzadeh and F. Gibou, "A conservative discretization of the Poisson–Nernst–Planck equations on adaptive Cartesian grids," *J. Comput. Phys.*, vol. 274, pp. 633–653, Oct. 2014.

[217] R. B. Smith and M. Z. Bazant, "Multiphase porous electrode theory," *J. Electrochem. Soc.*, vol. 164, no. 11, pp. E3291–E3310, Jun. 2017.

[218] D. Miranda, C. M. Costa, A. M. Almeida, and S. Lanceros-Méndez, "Computer simulation evaluation of the geometrical parameters affecting the performance of two dimensional interdigitated batteries," *J. Electroanal. Chem.*, vol. 780, pp. 1–11, Nov. 2016.

[219] P. Priimägi, D. Brandell, S. Srivastav, A. Aabloo, H. Kasemägi, and V. Zadin, "Optimizing the design of 3D-pillar microbatteries using finite element modelling," *Electrochim. Acta*, vol. 209, pp. 138–148, Aug. 2016.

[220] T. F. Fuller, M. Doyle, and J. Newman, "Simulation and optimization of the dual lithium ion insertion cell," *J. Electrochem. Soc.*, vol. 141, no. 1, Art. no. 1, Jan. 1994.

[221] B. van Leer, "Towards the ultimate conservative difference scheme. II. Monotonicity and conservation combined in a second-order scheme," *J. Comput. Phys.*, vol. 14, no. 4, pp. 361–370, Mar. 1974.

[222] T. A. J. Ouermi, "Full cell comparative study," Army Research Laboratory, MSME Alliance, Durham, NC, USA, Quarterly Report, Oct. 2021.

[223] F. Shi, Z. Song, P. N. Ross, G. A. Somorjai, R. O. Ritchie, and K. Komvopoulos, "Failure mechanisms of single-crystal silicon electrodes in lithium-ion batteries," *Nat. Commun.*, vol. 7, Jun. 2016.