University of California
Los Angeles

Part I: Reconstruction of Missing Data in Social Networks Based on Temporal Patterns of Interactions
Part II: Constitutive Modeling in Solid Mechanics for Graphics Applications

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Mathematics

by

Alexey Dmitrievich Stomakhin

2013

Abstract of the Dissertation

Part I: Reconstruction of Missing Data in Social Networks Based on Temporal Patterns of Interactions Part II: Constitutive Modeling in Solid Mechanics for Graphics Applications

by

Alexey Dmitrievich Stomakhin

Doctor of Philosophy in Mathematics

University of California, Los Angeles, 2013

Professor Joseph Teran, Co-chair

Professor Andrea Bertozzi, Co-chair

In Part I, the author presents a mathematical framework based on a self-exciting point process aimed at analyzing temporal patterns in the series of interaction events between agents in a social network. We develop a reconstruction model formulated as a constraint optimization problem that allows one to predict the unknown participants in a portion of those events. The results are used to predict the perpetrators of the unsolved crimes in the Los Angeles gang network.

Part II discusses the work undertaken by the author in deformable solid body simulation. We first focus on purely elastic solids and develop a method for extending an arbitrary isotropic hyperelastic energy density function to inverted configurations. This energy based extension is designed to improve robustness of elasticity simulations with extremely large deformations typical in graphics applications and demonstrates significant improvements over similar stress based techniques presented in [40, 86]. Moreover, it yields continuous stress and unambiguous stress derivatives in all inverted configurations. We also introduce a novel concept of a hyper-elastic model's primary contour which can be used to predict its robustness and stability. We demonstrate that our invertible energy-density-based approach

outperforms the popular hyperelastic corotated model [13, 56] and show how to use the primary contour methodology to improve the robustness of this model to large deformations.

We further develop a novel snow simulation method utilizing a user-controllable constitutive model defined by an elasto-plastic energy density function integrated with a hybrid Eulerian/Lagrangian Material Point Method (MPM). The method is continuum based and its hybrid nature allows us to use a regular Cartesian grid to automate treatment of self-collision and fracture. It also naturally allows us to derive a grid-based implicit integration scheme that has conditioning independent of the number of Lagrangian particles. We demonstrate the power of our method with a variety of snow phenomena.

The dissertation of Alexey Dmitrievich Stomakhin is approved.

Jeffrey Eldredge

Stanley Osher

Andrea Bertozzi, Committee Co-chair

Joseph Teran, Committee Co-chair

University of California, Los Angeles

2013

To my family and friends

Table of Contents

List of Figures

List of Tables

Acknowledgments

First and foremost I would like to thank my advisors Joseph Teran and Andrea Bertozzi. Needless to say this dissertation would not exist if it were not for their advice, guidance, support and collaboration. I would also like to thank the rest of my committee, Stanley Osher and Jeffrey Eldredge, for the time and interest required to review my work.

Aside from my advisors, I would like to thank all my collaborators as I have never worked with such a large group of extremely bright individuals. The research in Part I was done in collaboration with Martin Short and Andrea Bertozzi. The published article "Reconstruction of Missing Data in Social Networks Based on Temporal Patterns of Interactions", Inverse Problems, 27(11), 115013, 2011 contains much of the research presented in Part I. I am also grateful to George Mohler, Jeffrey Brantingham, George Tita, Eric Lewis, Rachel Hegemann, Laura Smith and Alethea Barbaro for their insight and helpful discussions.

The research in Part II was done in collaboration with Craig Schroeder, Andy Selle, Lawrence Chai, Russell Howes and Joseph Teran, and contains work published in "Energetically Consistent Invertible Elasticity", Eurographics Symposium on Computer Animation (SCA), pp. 25-32, 2012 and "A Material Point Method for Snow Simulation" recently accepted to SIGGRAPH 2013. Others in the group that I have worked with were Jeffrey Hellrung, Jan Hegemann, Yuting Wang, Casey Richardson, Chenfanfu Jiang, Gergely Klár and Jingyi Fang.

I also had a privilege of consulting at Walt Disney Animation Studios. I worked closely with Rasmus Tamstorf, Andy Selle, Lawrence Chai and Aleka McAdams. I am grateful to them for the collaboration which deepened my understanding of special perspective of science in graphics and had a profound impact on my research.

Finally, I would like to thank my parents Dmitry Stomakhin and Tatiana Stomakhina, as well as my sister Ekaterina Stomakhina, for their support and love.

Vita

2008-2009	Undergraduate Intern, Intel Corporation, Moscow, Russia.
2009	B.S. (Applied Mathematics and Physics), Moscow Institute of Physics and Technology.
2010-2011	Teaching Assistant, UCLA.
2010-2013	Research Assistant, UCLA.
2011	M.A. (Mathematics), UCLA.
Summer 2011	Graduate Research Intern, Walt Disney Animation Studios, Burbank, CA.
Summer 2012	Graduate Research Intern, Walt Disney Animation Studios, Burbank, CA.
Spring 2013	Software Engineer, Walt Disney Animation Studios, Burbank, CA.

Publications

A. Stomakhin, M. Short, and A. Bertozzi, *Reconstruction of Missing Data in Social Networks Based on Temporal Patterns of Interactions*, Inverse Problems, 27(11), 115013, 2011

A. Stomakhin, R. Howes, C. Schroeder, and J. Teran, *Energetically Consistent Invertible Elasticity*, ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA), pp. 25-32, 2012

A. Stomakhin, C. Schroeder, L. Chai, J. Teran, and A. Selle, *A Material Point Method for Snow Simulation*, accepted to SIGGRAPH 2013

Part I

Reconstruction of Missing Data in Social Networks Based on Temporal Patterns of Interactions

CHAPTER 1

Introduction

Prediction of missing information is an important part of data analysis in social sciences [75, 49, 37]. The examples studied in literature, mostly by statisticians, include reconstruction of the unknown connections in a social network [36, 35], analyzing non-ignorable non-responses in a survey sampling [38, 10], and many others. The most common way to deal with missing values is to replace them by some plausible estimates using known or model-based cross-dependencies over the network in question.

However, these methods do not typically consider networks that change with time, when another source of information is given by the temporal patterns arising from the network evolution. Such networks are the primary object of study in this manuscript.

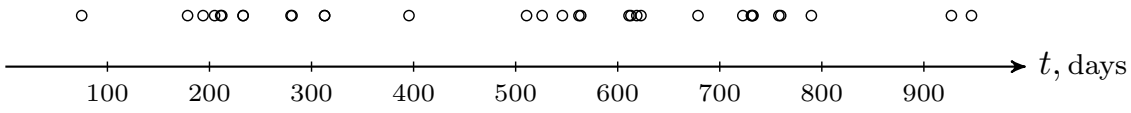

Figure 1.1: Temporal clustering of the interaction events between Clover and East Lake gangs in Los Angeles, during the period 1999-2002.

As our main example, we consider the gang rivalry network in the Los Angeles policing district Hollenbeck [90]. Police data on gang crimes from 1999 to 2002 reveal temporal clustering of gang interaction events, which is demonstrated in Figure 1.1. These temporal patterns can be used to solve the following inverse problem: predict the participants of the gang-related crimes if some of them are not known.

For a given pair of agents, the interaction events can either be independent, following a Poisson process, or temporally dependent, in which case the occurrence of one event can

change the likelihood of subsequent events in the future. Such event dependency for the Los Angeles gang network has been established in [22], where a Hawkes process [32, 33], commonly used in seismology to model earthquakes [68, 100], was compared to inter-gang violent crimes.

The rest of the work is organized as follows. In Chapter 2 we formalize the problem and describe a model of interaction between network agents based on a Hawkes process. In Chapter 3 we propose a way of predicting the unknown participants of interaction events, which we formulate as a constrained optimization problem. We analyze our method and the solution it gives and also present and discuss the prediction results.

CHAPTER 2

Problem Formulation

2.1 Problem Description

We model a social network as a graph with nodes representing the agents and edges, or binary links [35], indicating whether or not the corresponding pair of agents interact. We further look at the series of pairwise interaction events between the agents, characterized by their occurrence times and the pairs involved. We assume that the network structure represented by the graph does not change with time, although each pair of interacting agents can have its own prescribed model of behavior that might involve some time dependence. Suppose all the times of the events are known, but for some of them, data on one or both of the participants are missing. The problem is to reconstruct the missing data about the participants based on the behavioral model.

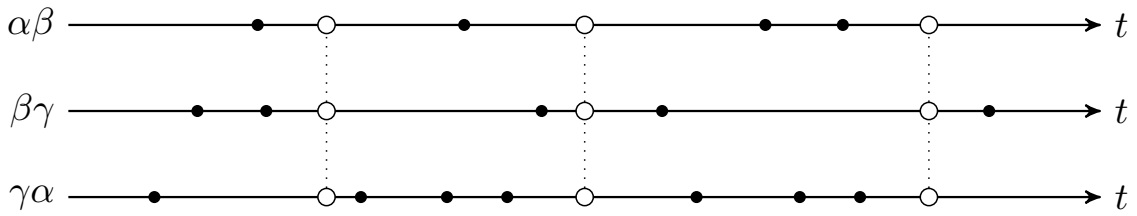

Figure 2.1: Graphical representation of the problem.

Before we proceed, let us discuss a convenient graphical representation of the problem shown in Figure (2.1). Here we deal with a network consisting of three agents α, β, and γ, with all pairs being active. The black points correspond to the events without any missing information. All events are ordered in time and there is a separate timeline for each pair of agents. The incomplete events, which are those with missing data about the participants,

cannot be assigned to any particular timeline and are therefore represented via vertical series of white circles. Our goal is to replace each vertical set of white circles with a black circle on one of the timelines in a way that will give the most plausible picture in accordance with the model.

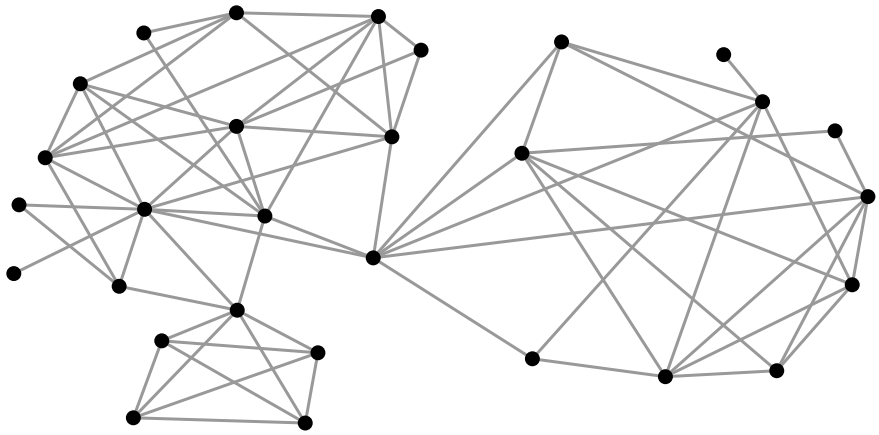

Figure 2.2: Graph of the gangs network in the Los Angeles policing district Hollenbeck [90]. Each of the twenty-nine gangs is represented by a node, and the edges indicate the presence of rivalries between them.

Returning to the network of gangs in Los Angeles: there are twenty-nine agents and the binary links indicate the existing rivalries between them, shown in Figure (2.2). In case of a rivalry, we have a series of crimes corresponding to the interaction events. These are typically murders, shots fired, etc. The data captures the information about which two gangs were involved in a crime; however, for a large fraction of them only victim affiliation is provided. The problem in this case is to estimate the affiliation of the unknown offenders.

2.2 Agent Interaction Model

A Hawkes process [32, 33] is a self-exciting point process commonly used in seismology to model earthquakes [68, 100] and defined by its intensity function

$$\lambda(t) = \mu + \theta \sum_{t_i<t} g(t - t_i). \tag{2.1}$$

The intensity function $\lambda(t)$ is partitioned into the sum of a Poisson background rate μ and a self-exciting component, through which events trigger an increase in the intensity of the process. The elevated rate spreads in time according to the kernel g, with θ being the scaling factor of the effect. In other words, each event generates a sequence of offspring or repeat events, which leads to temporal clustering. This agrees with the evidence that retaliations are commonplace among rival gangs [91, 41]. A similar approach was used to model repeat and near-repeat burglary effects in [79, 60] and temporal dynamics of violence in Iraq in [52], where self-excitation is one of the key qualitative features of the process.

We assume that the interaction events for each pair of agents occur independently according to a Hawkes process. This assumption of independence is based on the tentative conclusion of [80]. That is, the network of Hollenbeck gangs may be in a homogeneous state, meaning that gangs are not tightly coupled to one another. Thus, if gang α is fighting with gang β, and gang γ begins attacking α, then α easily switches away from its rivalry with β to begin fighting γ. This switching is largely a random, independent event in the homogeneous state.

We make no exclusions for inactive pairs since for those we simply have $\mu = 0$, and it is also useful to set $\theta = 0$ to avoid confusion in the following analysis. For the function g, as in [22], we use an exponential distribution, which gives

$$\lambda(t) = \mu + \theta \sum_{t_i<t} \omega e^{-\omega(t-t_i)}. \tag{2.2}$$

Here ω^{-1} sets the time scale over which the overall rate $\lambda(t)$ returns to its baseline level μ after an event occurs [54]. From the behavioral point of view, θ represents the average number of direct offspring for each event and ω^{-1} is the expected waiting time until an offspring. To indicate that each pair of agents has its own interaction parameters, we use index notation and write

$$\lambda_{\alpha\beta}(t) = \mu_{\alpha\beta} + \theta_{\alpha\beta} \sum_{t_i^{\alpha\beta}<t} \omega_{\alpha\beta} e^{-\omega_{\alpha\beta}(t-t_i^{\alpha\beta})}, \tag{2.3}$$

with $\mu_{\alpha\beta}, \theta_{\alpha\beta}, \omega_{\alpha\beta}$ being constants, unique for each pair, and summation over all previous

events between the agents α and β. If no confusion is possible, we will omit the indices $\alpha\beta$ to simplify the notations in the future.

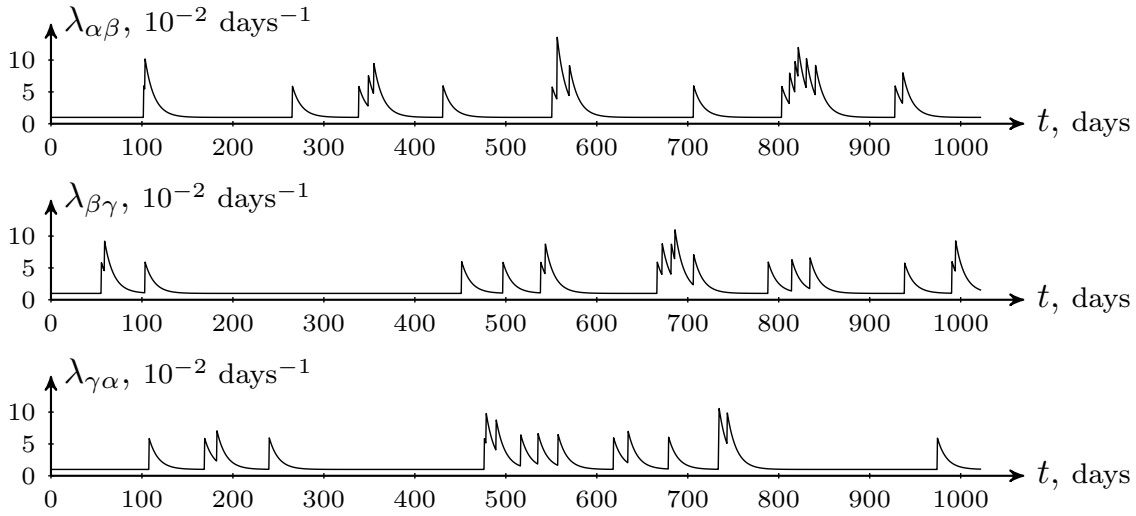

Figure 2.3: Data generated according to a Hawkes process with the same parameters for each pair of agents: $\mu = 10^{-2}$ days^{-1}, $\omega = 10^{-1}$ days^{-1}, $\theta = 0.5$.

In Figure (2.3), we present an example of data generated according to the described model (2.3) for a network consisting of three agents α, β, and γ. Here, the same parameters are used for each pair: $\mu = 10^{-2}$ days^{-1}, $\omega = 10^{-1}$ days^{-1}, $\theta = 0.5$. These have approximately the order of magnitude estimated in [22] for the Los Angeles gang network.

Note here that obtaining the interaction parameters based on given data is a separate problem which is not addressed in the current work. We have some discussion of this in Chapter 4. In what follows, we assume that all the interaction parameters are known and use them to predict missing participants of the incomplete events.

CHAPTER 3

Data Reconstruction

3.1 Reconstruction Method

We will use the following notation:

$$\begin{aligned} N &- \text{total number of events} \\ n &- \text{number of incomplete events} \\ k &- \text{number of agents} \\ K &- \text{total number of pairs } = k(k-1)/2 \end{aligned}$$

To solve the prediction problem in question one could consider the likelihood function, defined on the space of all possible event lists, corresponding to different ways of filling in the missing data, which is to be maximized in order to get the most likely one. Given any *complete* event list, with no missing data, its likelihood is given by (see, for example, [22])

$$\mathcal{L} = \prod_{\alpha\beta} \prod_{t_i^{\alpha\beta}} \lambda_{\alpha\beta}\big(t_i^{\alpha\beta}\big). \tag{3.1}$$

The first product is over all possible *unordered* pairs of agents, and the second one is over all events for a fixed pair.

Note that maximizing (3.1) is a combinatorial type problem since the set of all agent pairs is discrete. One of the possible approaches to this problem would be to use simulated annealing [73] or Monte Carlo Markov Chain [7] techniques to estimate the maximum of the likelihood. These methods though being probabilistic metaheuristics usually require problem-dependent tuning, and can be rather slow. Another technique is to consider an

approximation to the real likelihood function (3.1). Some examples of pseudolikelihood methods for point processes were introduced for instance in [67, 69]. Note however, that no matter what approximation we take it still will be a function defined over a discrete set. Unfortunately then, there seems to be no significantly more optimal way than "full search" for solving this problem exactly in the general case, which is very inefficient since its complexity depends exponentially on n.

The goal is therefore to make the maximization problem computationally less expensive, while maintaining the plausibility of the predictions. To do this, one could define a smooth extension of (3.1) and then look for its maximum, so that some standard continuous optimization method like gradient ascent could be used. This could be achieved by allowing each incomplete event to move continuously between the timelines. However, such an approach is not naturally applicable to (3.1) due to its multiplicative structure.

We therefore propose the following. We design some reasonable approximation to the real likelihood function (3.1), such that its continuous extension is physically meaningful. Let us start with the following simple example. Consider a network consisting of three agents α, β, and γ with all pairs having the same interaction parameters. Suppose only one event is incomplete and there is no information about its participants. Intuitively, because of the self-exciting nature of the process, the event is less likely to belong to the pairs with no nearby interaction, and more likely to belong to those for which it can be considered as a part of a cluster. For instance, in the situation shown in Figure (3.1), agents β and γ are the most likely participants of the incomplete event, as this would place it within a cluster.

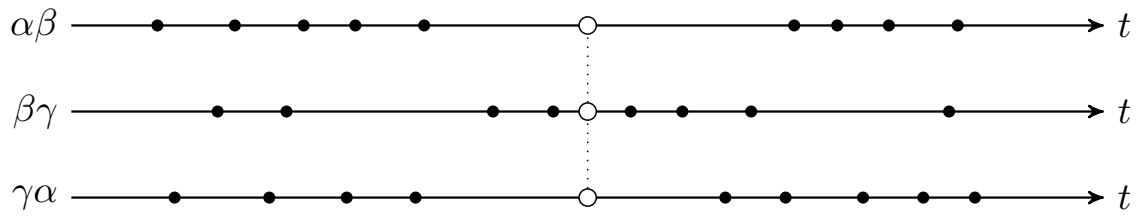

Figure 3.1: Example of reconstruction based on temporal clustering: agents β and γ are the most likely participants of the incomplete event, as this would place it within a cluster.

To give this idea a quantitative formulation, we note that clusters correspond to the periods of time with higher values of the intensity functions, which can be seen in Figure (2.3). Hence, for a missing event it would be reasonable to predict the pair with the highest intensity at the moment of the event. It also makes sense from the probabilistic point of view, because given the fact that an event happened at time t the probability of pair $\alpha\beta$ being involved is proportional to $\lambda_{\alpha\beta}(t)$.

Now we construct our energy functional: an approximation to the likelihood function (3.1) on the space of all possible event lists, corresponding to different ways of filling in the missing data, which is to be maximized in order to get the "most likely" one. Given any *complete* event list, with no missing data, we define its energy as

$$\Lambda = \sum_{\alpha\beta} \sum_{t_i^{\alpha\beta}} \lambda_{\alpha\beta}\big(t_i^{\alpha\beta}\big). \tag{3.2}$$

The first summation is over all possible *unordered* pairs of agents, and the second one is over all events for a fixed pair. Here we basically say that the "chances" of a pair to be involved in an interaction event are equal to its intensity function value at that time. Then we take the sum over all events. Roughly speaking, the metric defined by (3.2) assigns higher values to the event lists with denser clusters, which is precisely what we need to get a reasonable prediction. If no confusion is possible we will replace $t_i^{\alpha\beta}$ with i in the summation index, keeping in mind that each pair of agents has their own timeline and system of indices for the events on it. Substituting (2.3) into (3.2) gives

$$\Lambda = \sum_{\alpha\beta} \sum_{i,j} \delta_{ij}\mu_{\alpha\beta} + \frac{1}{2}(1-\delta_{ij})\theta_{\alpha\beta}\omega_{\alpha\beta}e^{-\omega_{\alpha\beta}|t_i^{\alpha\beta}-t_j^{\alpha\beta}|}. \tag{3.3}$$

Thus, Λ is decoupled into the sum of the energies of the events themselves, determined by the background rates, and the sum of the pairwise interaction energies between the events on the same timeline due to self-excitation. Clustering leads to stronger interaction, increasing the value of Λ. Clearly, functional (3.3) is invariant with respect to time inversion, which means that each event affects its successors and predecessors in the same way.

As alternative to (3.2), one could normalize the intensity functions over all pairs of agents

to make them add up to 1, and define the energy functional as

$$\tilde{\Lambda} = \sum_{\alpha\beta} \sum_{t_i^{\alpha\beta}} \frac{\lambda_{\alpha\beta}\left(t_i^{\alpha\beta}\right)}{\sum_{\alpha'\beta'} \lambda_{\alpha'\beta'}\left(t_i^{\alpha\beta}\right)}, \tag{3.4}$$

an approach that might seem to be more natural from the probabilistic point of view. However it makes the final optimization problem to be solved much more nonlinear and has a drawback discussed in Section 3.2.

Again, maximizing the energy functional (3.2) or (3.4) is a combinatorial type problem since the set of all agent pairs is discrete, and there seems to be no significantly more optimal way than "full search" for solving it in the general case, which is very inefficient. However, unlike the likelihood function (3.1), it admits a physically meaningful smooth extension. This can be obtained by distributing each of the incomplete events over the timelines with weights that "add up", in some sense, to 1. Thus, in Figure (2.1), we would replace the white circles with black ones and add weights to each of those; the complete events naturally receiving weight 1. We can interpret this to mean that each incomplete event occurred *partially* on every timeline with effect (the jump in the intensity function) proportional to the corresponding weight. This new continuous maximization problem not only gives the most likely participants of an event, but also assigns a weight to each pair showing how likely that pair is to be involved.

To avoid misunderstanding, let us specify how we enumerate the events on a timeline, which does matter now due to the normalization coupling of the pairs. The reader can use Figure (3.2) as a reference. We start with incomplete events and assign them numbers from

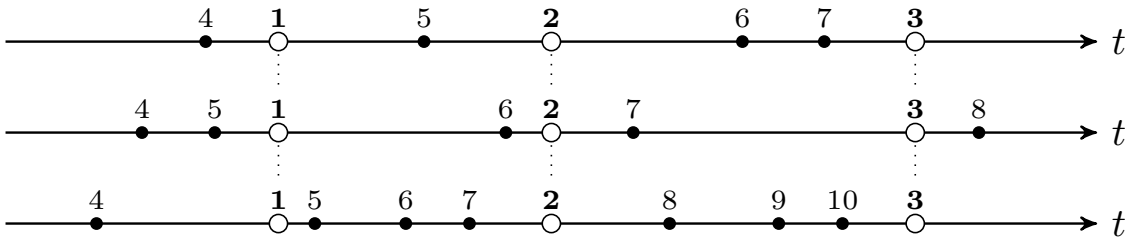

Figure 3.2: Events enumeration example.

1 to n. The order here is not important, as long as it is the same for all timelines. Then,

for each timeline we assign numbers to the complete events starting from $(n+1)$. Thus, there is a separate event indexing system for each timeline, with indices coinciding for the incomplete events.

Using l^2-normalization for the weights, we get the following formulation of the problem

$$\begin{cases} \max \left\{ \sum_{\alpha\beta} \sum_{i,j} \left[\delta_{ij} \mu_{\alpha\beta} m_i^{\alpha\beta} + \right. \right. \\ \qquad \left. \left. + \frac{1}{2}(1-\delta_{ij}) \theta_{\alpha\beta} \omega_{\alpha\beta} e^{-\omega_{\alpha\beta} |t_i^{\alpha\beta} - t_j^{\alpha\beta}|} m_i^{\alpha\beta} m_j^{\alpha\beta} \right] \right\}, \\ \sum_{\alpha\beta} \left(m_i^{\alpha\beta} \right)^2 = 1, \ \forall i = 1, \ldots, n \\ m_i^{\alpha\beta} \geq 0, \ \forall i = 1, \ldots, n, \ \forall \alpha\beta \end{cases} \tag{3.5}$$

where $m_i^{\alpha\beta}$ denotes the weight of the event number i on timeline $\alpha\beta$. As we mentioned before, the complete events have weight 1, so $m_i^{\alpha\beta} \equiv 1$ for $i > n$. The objective function is maximized with respect to $m_i^{\alpha\beta}$ for $i \leq n$, given the normalization and non-negativity constraints.

One could alternatively choose to use l^1-normalization for the weights, which again might seem to be more natural from the probabilistic point of view. The problem in that case is

$$\begin{cases} \max \left\{ \sum_{\alpha\beta} \sum_{i,j} \left[\delta_{ij} \mu_{\alpha\beta} m_i^{\alpha\beta} + \right. \right. \\ \qquad \left. \left. + \frac{1}{2}(1-\delta_{ij}) \theta_{\alpha\beta} \omega_{\alpha\beta} e^{-\omega_{\alpha\beta} |t_i^{\alpha\beta} - t_j^{\alpha\beta}|} m_i^{\alpha\beta} m_j^{\alpha\beta} \right] \right\}, \\ \sum_{\alpha\beta} m_i^{\alpha\beta} = 1, \ \forall i = 1, \ldots, n \\ m_i^{\alpha\beta} \geq 0, \ \forall i = 1, \ldots, n, \ \forall \alpha\beta \end{cases} \tag{3.6}$$

However, this method is unstable with respect to the input data, as we will see in Section 3.2.

Note here that the discrete, combinatorial version of this method can be obtained from

(3.5) or (3.6) by forcing all weights to be integers

$$\begin{cases} \max \Big\{ \sum_{\alpha\beta} \sum_{i,j} \Big[\delta_{ij}\mu_{\alpha\beta} m_i^{\alpha\beta} + \\ \qquad +\frac{1}{2}(1-\delta_{ij})\theta_{\alpha\beta}\omega_{\alpha\beta} e^{-\omega_{\alpha\beta}|t_i^{\alpha\beta} - t_j^{\alpha\beta}|} m_i^{\alpha\beta} m_j^{\alpha\beta} \Big] \Big\} \\ \sum_{\alpha\beta} m_i^{\alpha\beta} = 1, \ \forall i = 1, \ldots, n \\ m_i^{\alpha\beta} \in \{0,1\}, \ \forall i = 1, \ldots, n, \ \forall \alpha\beta \end{cases} . \tag{3.7}$$

3.2 Examples

The purpose of this section is to discuss a few examples that will reveal some useful properties of the problem (3.5).

3.2.1 Example 1: Timescale Detection

Suppose $N = n = 2$, so we have two incomplete events, and suppose we do not have any information at all about the participants. For simplicity we also assume $\mu_{\alpha\beta} \equiv 0$ and $\theta_{\alpha\beta} \equiv 1$. Then the problem to be solved according to (3.5) is

$$\begin{cases} \max \sum_{\alpha\beta} \omega_{\alpha\beta} e^{-\omega_{\alpha\beta}\Delta t} m_1^{\alpha\beta} m_2^{\alpha\beta} \\ \sum_{\alpha\beta} (m_i^{\alpha\beta})^2 = 1, \ \forall i = 1, 2 \\ m_i^{\alpha\beta} \geq 0, \ \forall i = 1, 2, \ \forall \alpha\beta \end{cases} , \tag{3.8}$$

with Δt being the time interval between the events. Note that (3.8) can be written conveniently in vector form as

$$\begin{cases} \max \boldsymbol{m}_1^T \boldsymbol{D} \boldsymbol{m}_2 \\ \|\boldsymbol{m}_1\|_2 = \|\boldsymbol{m}_2\|_2 = 1 \\ m_i^{\alpha\beta} \geq 0, \ \forall i = 1, 2, \ \forall \alpha\beta \end{cases} , \tag{3.9}$$

where we have used the notations

$$\begin{cases} \boldsymbol{D} = \mathrm{diag}\{\omega_{\alpha\beta} e^{-\omega_{\alpha\beta}\Delta t}\} \in \mathbb{R}^{K \times K} \\ \boldsymbol{m}_i = \{m_i^{\alpha\beta}\} \in \mathbb{R}^K, \ i = 1, 2 \end{cases} . \tag{3.10}$$

From linear algebra, it is well-known that the objective function in (3.9) is maximized when $\boldsymbol{m}_1 = \boldsymbol{m}_2 = \boldsymbol{e}_{\alpha'\beta'}$, the unit vector, such that

$$\alpha'\beta' = \arg\max_{\alpha\beta}\{\omega_{\alpha\beta}e^{-\omega_{\alpha\beta}\Delta t}\}. \tag{3.11}$$

The maximum of $\omega e^{-\omega\Delta t}$ is achieved when $\omega = \frac{1}{\Delta t}$. Hence the solution of the problem (3.8) corresponds to the pair with self-excitation timescale closest to Δt.

Recall that the self-excitation timescale represents the average time until a repeat event occurs. Thus, since all background rates are equal to zero, and therefore the second event must be an offspring of the first one, our method indeed gives the most likely participants. Of course, for prediction purposes the distribution of the weights is not very realistic, because it rules out the possibility for all other pairs to be involved. But, as we will see further, there are other mechanisms that make the solution more regularized, which we do not see here due to a specific and, in fact, unrealistic structure of the example. Indeed, this example is in some sense pathological, as there is no way to explain the occurrence of the first event. However, we can think of it as a limiting case when

$$\sum_{\alpha\beta}\mu_{\alpha\beta} \ll \min_{\alpha\beta}\{\omega_{\alpha\beta}e^{-\omega_{\alpha\beta}\Delta t}\}. \tag{3.12}$$

Then the first event is a background one, which happened after waiting for sufficiently long time, and the second one is due to self-excitation, because the probability of it being a background event *from some timeline* is much less than the probability of it being an offspring of the previous event, as follows from (3.12).

Consider now the alternative energy functional (3.4) with normalization at each time point, introduced in Section 3.1. Clearly, the maximum value it can achieve, for the example in question, is 1. It happens whenever both events completely belong to the same pair of agents. Thus, this model does not "see" the dependence of clustering density on self-excitation timescale, and leads to a degenerate solution.

3.2.2 Example 2: Regularization

Suppose $N = n = 1$, so we have only one event which is incomplete, and suppose we do not have any information at all about the participants. Then the problem to be solved according to (3.5) is

$$\begin{cases} \max \sum_{\alpha\beta} \mu_{\alpha\beta} m^{\alpha\beta} \\ \sum_{\alpha\beta} \left(m^{\alpha\beta}\right)^2 = 1 \\ m^{\alpha\beta} \geq 0, \ \forall \alpha\beta \end{cases} . \tag{3.13}$$

Problem (3.13) can be written conveniently in vector form as

$$\begin{cases} \max \boldsymbol{\mu}^T \boldsymbol{m} \\ \|\boldsymbol{m}\|_2 = 1 \\ m^{\alpha\beta} \geq 0, \ \forall \alpha\beta \end{cases} , \tag{3.14}$$

where we have used the notations

$$\begin{cases} \boldsymbol{\mu} = \{\mu^{\alpha\beta}\} \in \mathbb{R}^K \\ \boldsymbol{m} = \{m^{\alpha\beta}\} \in \mathbb{R}^K \end{cases} . \tag{3.15}$$

The maximizer of (3.14) is well-known from linear algebra to be

$$\boldsymbol{m} = \frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|_2}. \tag{3.16}$$

Thus, the optimal weights, according to our method, are proportional to the corresponding background intensity rates. This is exactly what follows from the probabilistic approach. Indeed, we are dealing with the case where no self-excitation takes place, since there is only one event. Therefore the probability of a pair to be involved in the event is proportional to its background intensity rate.

Consider now the alternative model (3.6) with l^1-normalization, mentioned in Section 3.1. For this example it gives the following optimization problem

$$\begin{cases} \max \boldsymbol{\mu}^T \boldsymbol{m} \\ \|\boldsymbol{m}\|_1 = 1 \end{cases} . \tag{3.17}$$

Clearly, the objective function in (3.17) is maximized when $\boldsymbol{m} = \boldsymbol{e}_{\alpha'\beta'}$, the unit vector, such that

$$\alpha'\beta' = \arg\max_{\alpha\beta}\{\mu_{\alpha\beta}\}. \tag{3.18}$$

We see that the model picks the pair with the highest background rate, assigns weight 1 to it and 0 to the others. However, this is not the most desirable solution. Suppose, for instance, that all background rates are approximately the same. Then, it is not reasonable to choose one pair over the others, since all of them are almost equally likely to be involved. Unfortunately, this is a general property of model (3.6). It will always either assign all the weight to one pair for each incomplete event, never creating any distributions, or will give a degenerate solution. Indeed, the normalization constraints and the objective function, in each of its arguments, are all linear.

Model (3.5) does not have such a drawback for this example. It does not just pick the most likely participants of the event, but assigns weights to all pairs indicating how likely each of them is to be involved. This can be thought of as some sort of regularization property.

3.2.3 Discussion

As we mentioned in Section 3.1, the objective function in (3.5) can be thought of as a sum of the energies of the events. Formally, if we ignore constant terms, it consists of two parts: quadratic terms, corresponding to the interaction of the incomplete events, and linear terms, corresponding to the energy of the incomplete events in the presence of the complete events and background rate values. The examples above were targeted to examine these parts separately to reveal their roles in the reconstruction process.

In the first example we considered the quadratic part of the energy. We have seen that the incomplete events tend to gather on those timelines where their interaction energy is the highest, which leads to aggressive cluster formation up to assigning all the weights to the same pair of agents.

On the other hand, the linear terms express the influence of the complete events and

background rates, and do not allow the incomplete events to deviate too much from already existing clustering structure. Moreover they regularize the solution, which represents the degree of uncertainty in the prediction, as demonstrated in the second example.

The methods arising from l^1-normalization (3.6) and from the alternative energy functional (3.4) have each shown some undesirable properties in these examples, and we will not consider them further. Of course, one is not restricted to only using l^1 or l^2 normalization, and one could consider a general l^p normalization of the weights or look at a hybrid constraint consisting of both l^1 and l^2 terms (or l^p terms). In the hybrid case, a constraint of the form

$$\sum_{\alpha\beta} f\left(m^{\alpha\beta}\right)^2 + (1-f)m^{\alpha\beta} = 1$$

could be employed, where $f \leq 1$ would represent how much emphasis to put on the l^2 term or the l^1 term. Though, for simplicity, we do not consider such a constraint in this work, it remains a potential avenue for future exploration.

3.3 Analysis

Note from Figure (2.1) that the white circles naturally form a $K \times n$ matrix and our goal is to determine its entries. We denote the matrix as $\boldsymbol{X} = \{x_{ij}\}$. For future reference it will be useful to express $\boldsymbol{X}$ in terms of its rows and columns

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{r}_1^T \\ \vdots \\ \boldsymbol{r}_K^T \end{pmatrix} = \begin{pmatrix} \boldsymbol{c}_1 & \cdots & \boldsymbol{c}_n \end{pmatrix}. \tag{3.19}$$

Using these notations, problem (3.5) can be written as

$$\begin{cases} \sum_{i=1}^K \boldsymbol{r}_i^T \boldsymbol{A}_i \boldsymbol{r}_i + \boldsymbol{r}_i^T \boldsymbol{b}_i \to \max \\ \boldsymbol{c}_j^T \boldsymbol{c}_j = 1, \ \forall j = 1, \ldots, n \\ x_{ij} \geq 0, \ \forall i = 1, \ldots, K, \ \forall j = 1, \ldots, n \end{cases}. \tag{3.20}$$

Here $\boldsymbol{A}_i = \{a_i^{jl}\}$ is the symmetric $n \times n$ matrix of the interaction coefficients between the incomplete events on the i^{th} timeline, and $\boldsymbol{b}_i = \{b_i^j\}$ is the column of size n of the energy coefficients for the incomplete events in the presence of the complete events and background rate on the i^{th} timeline. Clearly, the entries of $\boldsymbol{A}_i$ and $\boldsymbol{b}_i$ are nonnegative, for all $i = 1, \ldots, K$.

Theorem 3.3.1. *For the problem (3.20):*

(i) There exists a global maximizer.

(ii) Every local maximizer (or even a stationary point) is a global maximizer.

(iii) If all b_i^j are strictly positive then the maximizer is unique.

Proof. The objective function is continuous and the admissible set, given by the constraints, is compact. This proves (i). Define $y_{ij} = x_{ij}^2$. Then the problem (3.20) becomes

$$\begin{cases} \sum_{i=1}^K \left[\sum_{j,l=1}^n a_i^{jl} \sqrt{y_{ij} y_{il}} + \sum_{j=1}^n b_i^j \sqrt{y_{ij}} \right] \to \max \\ \sum_{i=1}^K y_{ij} = 1, \ \forall j = 1, \ldots, n \\ y_{ij} \geq 0, \ \forall i = 1, \ldots, K, \ \forall j = 1, \ldots, n \end{cases} . \tag{3.21}$$

The admissible set in (3.21), given by the constraints, is convex. We will show that the objective function is concave on it, and strictly concave if all b_i^j are strictly positive, which implies (ii) and (iii).

Note that $a_i^{jl} \sqrt{y_{ij} y_{il}}$ is concave for all $i = 1, \ldots, K$ and $j, l = 1, \ldots, n$. This follows from the fact that for all $a, b, c, d \geq 0$ and $0 < \lambda < 1$ we have

$$\sqrt{\big(\lambda a + (1-\lambda)c\big)\big(\lambda b + (1-\lambda)d\big)} \geq \lambda\sqrt{ab} + (1-\lambda)\sqrt{cd}. \tag{3.22}$$

Indeed, squaring both sides of (3.22) gives a Cauchy-type inequality

$$cb + ad \geq 2\sqrt{abcd}, \tag{3.23}$$

after simplification. Now it suffices to show that the function

$$f_j(y_{1j}, \ldots, y_{Kj}) = \sum_{i=1}^K b_i^j \sqrt{y_{ij}} \tag{3.24}$$

is concave for all $j = 1, \ldots, n$. That is

$$\sum_{i=1}^{K} b_i^j \sqrt{\lambda \hat{y}_{ij} + (1-\lambda)\check{y}_{ij}} \geq \sum_{i=1}^{K} b_i^j \left[\lambda \sqrt{\hat{y}_{ij}} + (1-\lambda)\sqrt{\check{y}_{ij}}\right] \tag{3.25}$$

for all admissible distinct $\{\hat{y}_{ij}\}_{i=1}^K$, $\{\check{y}_{ij}\}_{i=1}^K$ and $0 < \lambda < 1$. We further wish to show that (3.24) is strictly concave, that is the inequality (3.25) must be strict, if all b_i^j are strictly positive. But both are true since the function $\sqrt{x}$ is strictly concave on $\{x : x \geq 0\}$. *This completes the proof.* □

If all pairs are active, then all background rates are nonzero, and we automatically have all b_i^j strictly positive, which implies the uniqueness of prediction in accordance with the Theorem 3.3.1. When some pairs are inactive part (iii) of the Theorem 3.3.1 is not applicable directly. Indeed, if for example timeline i is inactive, then there are no complete events on it and the corresponding background rate is 0, hence $\boldsymbol{b}_i = \mathbf{0}$. Note however that in this case adding the constraint $\boldsymbol{r}_i = \mathbf{0}$, or simply excluding the timeline i from consideration, gives a problem with a smaller unknown matrix equivalent to (3.20). Thus, if we eliminate all inactive pairs in this way, we get a problem with all pairs in question being active, which guarantees the uniqueness of prediction.

So far we implicitly assumed that we had no information at all about the participants of the incomplete events and each pair was considered as a possible candidate for prediction. Of course, if one of the participants of an event is known, then the pairs without this agent can not be in involved, and the corresponding entries of $\boldsymbol{X}$ must be equal to 0, which means we have additional constraints of the form $x_{ij} = 0$ for the problem (3.20). These constraints however do not affect the convexity of the admissible set in the coordinates $y_{ij} = x_{ij}^2$. Therefore all results of the Theorem 3.3.1 remain valid.

3.4 Results

In this section we present and discuss the results of various tests of the proposed reconstruction method. Since the data from the Los Angeles gang network is incomplete, and the

ground truth and interaction parameters for it are unavailable, it is not quite suitable for this purpose. Instead, we generate synthetic data using a Hawkes process (2.3), throw out some of the data at random, and then apply our algorithm to reconstruct it.

To evaluate the performance of our algorithm, we only focus on the *ordering* of the various $m^{\alpha\beta}$ for each incomplete event i. Specifically, we determine for each incomplete event i the weights m_i for that event on the various timelines, order them from lowest to highest, and find the corresponding rank of the ground truth timeline for that event. This is done for two major reasons. First, our method (3.5) does not assign proper probabilities to the various timelines, only weights that should be interpreted as being related to probability in a monotonic way. Second, from an operational point of view, the authorities are not very concerned with the actual probabilities with which each gang committed a given crime, but rather with a simple ranking of gangs from most likely to least likely, to prioritize their investigation.

As a first step, we compare our continuous method (3.5) to two others: one derived from the likelihood function (3.1) and one using the discrete model (3.7). However, note that the methods (3.1) and (3.7) provide likelihoods (or energies) only for full allocations $\mathcal{A}$ of incomplete events, rather than one likelihood for each timeline per event. To bypass this issue, we simply define the likelihood $\hat{m}_i^{\alpha\beta}(f)$ that incomplete event i belongs to timeline $\alpha\beta$ under metric f to be

$$\hat{m}_i^{\alpha\beta}(f) = \sum_{\mathcal{A}_i^{\alpha\beta}} f(\mathcal{A}) \ , \tag{3.26}$$

where $\mathcal{A}_i^{\alpha\beta}$ is meant to represent only those allocations in which incomplete event i is attributed to timeline $\alpha\beta$, and $f = \mathcal{L}$ for (3.1) and $f = \Lambda$ for (3.7).

As mentioned previously, the methods (3.1) and (3.7) are of combinatoric complexity, so we limit our testing here to a relatively small system with $N = 40$, $n = 4$, $k = 4$, $K = 6$. Here, we assume no knowledge of the participants in incomplete events, so that each may be assigned to any of the $K = 6$ timelines. Simulations were run 10,000 times using parameters $\mu = 10^{-2}$ days^{-1}, $\omega = 10^{-1}$ days^{-1}, and $\theta = 0.5$ for each pair of agents, which

have approximately the order of magnitude estimated in [22]. Each simulation generated a ranking of the timelines for each incomplete event, and the percentages of incomplete events whose ground truth timelines were given certain ranks are shown in Table 3.1. Note

Table 3.1: Continuous method (3.5) compared to methods (3.1) and (3.7), for $N = 40$, $n = 4$, $k = 4$, $K = 6$, $\mu = 10^{-2}$ days^{-1}, $\omega = 10^{-1}$ days^{-1}, and $\theta = 0.5$.

Method	Top 1	Top 2	Top 3	Top 4	Top 5
(3.1)	47.3%	68.1%	79.8%	87.7%	94.0%
(3.5)	47.1%	68.1%	79.7%	87.6%	94.1%
(3.7)	47.0%	68.1%	79.7%	87.6%	94.0%

that the three methods perform almost identically, each placing the correct timeline at top likelihood approximately 47% of the time, in the top two likelihoods approximately 68% of the time, and in the top three likelihoods approximately 80% of the time. Since method (3.5) yields nearly indistinguishable solutions to those of (3.1) and (3.7), but is vastly more computationally effective, we focus only on this continuous method for the remainder of this section.

We next test our continuous method using datasets that more closely mimic the gang rivalry data. In all the experiments below, we have exactly one participant unknown for every incomplete event, which is the case for most of the gang data. Also, unless specified otherwise, we assume full connectedness of the network graph and use the same interaction parameters for each pair of agents as used above: $\mu = 10^{-2}$ days^{-1}, $\omega = 10^{-1}$ days^{-1}, $\theta = 0.5$.

Table 3.2 demonstrates the performance of the continuous method (3.5). It is organized as follows. The first three columns describe the dimensions of the network and the data the method was applied to, and the last three indicate how often, on average, a ground-truth unknown pair was in the top one, top two, and top three weights of the predicted distribution. The $\star$ value of k corresponds to the real Los Angeles gang network (see Figure (2.2)), which is not a fully connected graph. The "Guessing" rows show the results that would be obtained by random guessing.

First we note that, in terms of prediction quality, the Los Angeles gang network roughly corresponds to a fully connected 6-nodes graph. This actually makes sense, since each gang has about 5 rivalries on average. Second, the prediction results depend rather mildly on the fraction of incomplete events, which implicitly confirms the fact that reconstruction model (3.5) captures the qualitative features of interaction process (2.3) rather well.

As for the results themselves, we can see that they are significantly better than those obtained by just random guessing. At the same time they are not perfect. To see why this is so we need to have a closer look at how they depend on the parameters of the system: μ, ω, and θ. If self-excitation is too weak, that is $\omega/\mu \ll 1$ and $\theta \ll 1$, then the rate (2.3) will always stay near μ and the clusters will be vague and widespread. Hence the method will give almost uniform distributions of weights, and choosing the pair with the biggest weight will be equivalent to random guessing. On the other hand, if self-excitation is very strong, that is $\omega/\mu \gg 1$ and $\theta \simeq 1$, then the clusters will be sharp, the distribution vectors will be sparse, and choosing the pair with the biggest weight will give a reliable prediction.

Figure (3.3) confirms the above reasoning. Here we applied our method to a fully-connected 6-agents network, with $N = 400$, $n = 100$, varying the values of θ and $\tau = \log_{10}(\omega/\mu)$. For each distribution vector of weights, we simply picked the timeline with the highest weight and plotted average percentage of correct predictions obtained in this way.

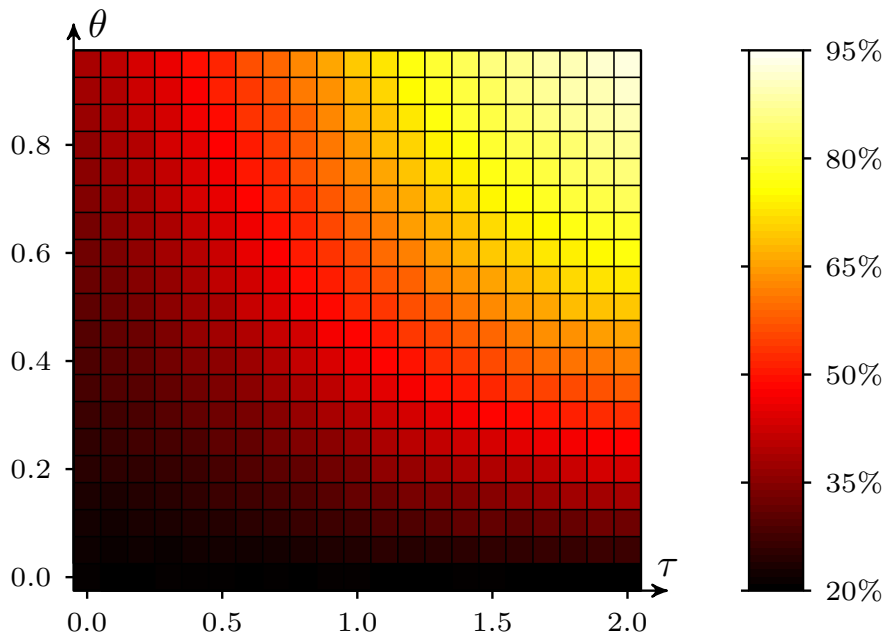

Figure 3.3: Dependence of the average percentage of correct predictions, obtained by choosing the pair with the highest weight for each distribution vector, on θ and $\tau = \log_{10}(\omega/\mu)$, for a fully-connected 6-agents network, with $N = 400$, $n = 100$.

Table 3.2: Continuous model (3.5) performance results. The first three columns describe the dimensions of the network and the data the method was applied to, and the last three indicate how often, on average, a ground-truth unknown pair was in the top one, top two, and top three weights of the predicted distribution. The $\star$ value of k corresponds to the real Los Angeles gang network, see Figure (2.2), which is not a fully connected graph. The "Guessing" rows show the results that would be obtained by random guessing.

k	N	n	Top 1	Top 2	Top 3
5	400	50	57%	80%	92%
5	400	100	56%	79%	91%
5	400	200	54%	76%	90%
5	Guessing		25%	50%	75%
7	400	50	47%	69%	82%
7	400	100	46%	68%	80%
7	400	200	45%	65%	77%
7	Guessing		17%	33%	50%
9	400	50	42%	62%	73%
9	400	100	41%	60%	72%
9	400	200	39%	57%	69%
9	Guessing		13%	25%	38%
$\star$	400	50	50%	72%	83%
$\star$	400	100	49%	71%	82%
$\star$	400	200	48%	68%	80%

CHAPTER 4

Conclusion

Retaliatory gang violence is a large problem in many metropolitan areas around the globe, and to curtail such violence, law enforcement agencies need to know who the participants were in a given altercation. We have shown that, under the assumptions that retaliatory violence on a gang network follows a Hawkes process of the form (2.3), incomplete data on the participants of the offenses can be reconstructed using a computationally effective algorithm that maximizes an energy functional under a set of constraints – method (3.5). Moreover, when focusing on the likelihood rankings of gangs for incomplete events, method (3.5) seems to perform on par with a more probability-based algorithm (3.1) that is too complex to use on realistically sized datasets. Finally, we have shown that the performance of our method is deeply connected to the parameters of the Hawkes process in question, and in certain regimes may predict the correct participants with very high likelihood.

Of course, there are issues to overcome if our method is to be used on actual gang violence data, rather than on simulated events. Firstly, for real datasets, the parameters of the process must be estimated from the events, rather than being known *a priori*. One could imagine accomplishing this in an iterative way: use the complete events to estimate parameters, use these parameters to estimate participants in unknown events, then use these estimates to re-estimate the parameters, continuing the cycle until convergence (if convergence is indeed obtained). To implement this, however, one must choose how to use the estimated participants of events when re-estimating the interaction parameters, something that is not entirely clear given that our estimates of the participants are not probabilities. This could perhaps be accomplished via the expectation-maximization algorithm [19]. In this

case one would first have to consider the real likelihood function, incorporating this time both missing events and unknown parameters, and then come up with a suitable approximation for it that would make the problem less computationally expensive and yield plausible results.

Secondly, in real datasets one must be concerned with systematic deviation between the data and actual occurrences. Certain types of gang violence may be chronically under-reported in ways that will skew the detection of self-excitation or cause events to be allocated in an improper way. A thorough understanding of how this might affect our estimates should be had before trusting the results completely.

Part II

Constitutive Modeling in Solid Mechanics for Graphics Applications

CHAPTER 5

Introduction

Physically based simulation of deformable solids is an indispensable tool for creating realistic virtual environments [3, 45, 24]. Typically, it involves numerical methods that evolve discrete geometric models over time. In addition, one needs to specify the material properties of an object to be simulated by defining a *constitutive model*: an equation relating stress arising in the material with the deformation it undergoes. It determines the look and dynamics of the material and requires careful consideration when a certain type of behavior needs to be produced.

Constitutive modeling for different kinds of materials has been thoroughly studied in physics and mechanical engineering literature [8, 92, 74, 14]. Those models however may not be directly applicable to simulation in graphics, for a number of reasons. Firstly, physics is usually concerned with simple tests that can be experimentally verified. In graphics, on the other hand, it is often necessary to demonstrate the full richness of a certain phenomena. Needless to say, a constitutive model that can handle the whole range of required behaviors might not have been completely developed. Examples usually include materials having either complex inner structure or phase transition, e.g. *snow*. Secondly, while simulation in graphics considers many of the same problems that computational physics does, it also has a unique perspective. A fundamental measure of quality in graphics is whether or not the final image looks plausible. Of course physical accuracy helps achieve that to large extent, but by no means is it the main goal. Naturally, practitioners are less interested in the root causes of physical phenomena but rather seek to have direct and intuitive control over the final result, which is the opposite of how material models get designed and used in physics and

mechanical engineering. Finally, a good constitutive model must be capable of providing robust and stable results in all possible scenarios, including extremely large deformations and even inversion of the material [40, 86].

In what follows we present our work that addresses some of the issues in constitutive modeling for graphics applications described above. In Chapter 6 we consider large strain deformable object simulation which was introduced to computer graphics by [88]. Unfortunately, simulation of large deformations with a Lagrangian mesh is notoriously unstable and error-prone. Although many researchers have shown the effectiveness of adaptive refinement [78, 18, 31, 11] and hybrid Lagrangian/Eulerian approaches [4, 95, 96, 93], computer graphics researchers tend to use purely Lagrangian methods with a topologically static mesh. The primary problem for Lagrangian methods is the inversion of mesh elements that poorly approximate highly deformed regions. This motivated the development of models that are well defined when the deformation mapping has negative Jacobian. Irving et al. developed the invertible finite element (IFE) framework in [40] and [86] to extend arbitrary elastic constitutive models to inverted configurations. The "warped stiffness" [61, 76, 99, 62, 23] and corotated hyperelasticity [13, 56] models are also meaningfully defined through inversion. Another notable model defined through inversion was developed in [89].

We build on the IFE framework in [40] and [86] to provide a method for the practical extension of an arbitrary isotropic hyperelastic energy density to inverted configurations. Hyperelasticity refers to constitutive models for which the stress is determined as the gradient of an underlying scalar energy density. Our extension matches the original model for singular values on the uninverted side of a convex extrapolation threshold surface. In fact, we provide a heuristic that prevents the need for the costly SVD whenever the material is in this uninverted region. The smoothly extended energy allows for accurate and unambiguous definition of the stress and stress derivatives needed for force computation and implicit time integration. We show that this extension provides significantly superior behavior through inversion than both the original IFE and the corotated models in [13] and [56]. Lastly, we show that our new notion of a primary contour provides useful analysis of a model's robustness to

large deformation.

In Chapter 7 we develop a novel snow simulation method utilizing a user-controllable constitutive model defined by an elasto-plastic energy density function integrated with a hybrid Eulerian/Lagrangian Material Point Method (MPM).

Snow dynamics are amazingly beautiful yet varied. Whether it is *powder snow* fluttering in a skier's wake, foot steps shattering an *icy snow* crust or even *packing snow* rolled into balls to make a snowman, it is snow's rich repertoire that makes it simultaneously compelling for storytelling and infuriatingly difficult to model on a computer. Artists typically use simpler techniques combined in various ways to achieve snow effects [47, 15, 48], but these approaches do not produce the full richness of snow and are labor intensive. This suggests the need for a specialized solver that is adept at the full range of snow phenomena.

Specialized solvers for specific phenomena are frequently used in graphics and computational physics because achieving maximum resolution (and thus visual quality) requires efficiency. While a fluid simulator can produce solid-like elastic effects (and vice versa), it is not the most optimal strategy. When solids and fluids are needed simultaneously, researchers have developed two-way coupled systems to get good accuracy and performance for both phenomena. Unfortunately, snow has continuously varying phase effects, sometimes behaving as a rigid/deforming solid and sometimes behaving as a fluid. Thus, instead of discrete coupling we must simultaneously handle a continuum of material properties efficiently in the same domain, even though such a solver may not be most efficient for a single discrete phenomenon.

We present two main contributions that achieve these aims. First, we develop a semi-implicit Material Point Method (MPM) [84] specifically designed to efficiently treat the wide range of material stiffnesses, collisions and topological changes arising in complex snow scenes. To our knowledge, this is the first time MPM has been used in graphics. MPM methods combine Lagrangian material particles (points) with Eulerian Cartesian grids. Notably, there is no inherent need for Lagrangian mesh connectivity. Many researchers in graphics have experimented with hybrid grid and particle methods. For example, [97] simulate sand

as a fluid using a PIC/FLIP incompressible fluid technique. In fact, MPMs were designed as a generalization of the PIC/FLIP solvers to computational solids. As with PIC/FLIP, MPMs implicitly handle self-collision and fracture with the use of the background Eulerian grid. This is essential given the many topological changes exhibited by practical snow dynamics. Our second contribution is a novel snow constitutive model designed for intuitive user control of practical snow behavior. This is also designed to achieve our goal of describing the many phases of snow behavior with one constitutive relation. To do this, we borrow from the vast engineering literature on snow and demonstrate that an elasto-plastic treatment is an effective means of handling the transition between many different behaviors including flowing, clumping, breaking and more.

Note: For quick reference on solid mechanics and the notations used in the following chapters we refer the reader to Appendix A.

CHAPTER 6

Energetically Consistent Invertible Elasticity

6.1 Isotropic Hyperelasticity

We limit our discussion of elasticity to the constitutive stress/strain relationship. We provide enough detail that any spatial or temporal discretization technique can be used in practice. Specifically, we will describe how to compute stress needed for elastic forces and stress linearizations needed for implicit time stepping. Note that our energy-based approach works naturally with variational integrators like those in [43].

We assume a continuum description of the deformation $\boldsymbol{\phi} : \Omega_0 \to \mathbb{R}^3$ that maps initial (or material) points $\boldsymbol{X}$ in the initial configuration Ω_0 to points $\boldsymbol{x} = \boldsymbol{\phi}(\boldsymbol{X})$ in world space. The elastic force per unit volume in the continuum body is $\nabla^{\boldsymbol{X}} \cdot \boldsymbol{P}$, where $\boldsymbol{P}$ is the first Piola-Kirchoff stress [29]. For hyperelasticity, the first Piola-Kirchoff stress is determined from the energy density $\Psi(\boldsymbol{F})$ as $\boldsymbol{P} = \frac{\partial \Psi}{\partial \boldsymbol{F}}$, where $\boldsymbol{F} = \frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{X}}$ is the deformation gradient. We limit our focus to isotropic models for which the energy density can be written as $\Psi(\boldsymbol{F}) = \hat{\Psi}(\hat{\boldsymbol{\Sigma}})$, where $\boldsymbol{F} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ by the singular value decomposition and $\hat{\boldsymbol{\Sigma}} = \text{diag}(\boldsymbol{\Sigma}) = (\sigma_1, \sigma_2, \sigma_3)$. Note that isotropy is equivalent to defining the energy in terms of the principal invariants $I_1 = \sigma_1^2 + \sigma_2^2 + \sigma_3^2$, $I_2 = \sigma_1^2\sigma_2^2 + \sigma_2^2\sigma_3^2 + \sigma_1^2\sigma_3^2$, and $J = \sigma_1\sigma_2\sigma_3$ [29]. This implies that the energy is invariant under permutations of the singular values. In this case, it can be shown that the first Piola-Kirchoff stress has the form $\boldsymbol{P}(\boldsymbol{F}) = \boldsymbol{U}\hat{\boldsymbol{P}}(\hat{\boldsymbol{\Sigma}})\boldsymbol{V}^T$ where

$$\hat{\boldsymbol{P}}(\hat{\boldsymbol{\Sigma}}) = \begin{bmatrix} \hat{P}_1(\hat{\boldsymbol{\Sigma}}) & & \\ & \hat{P}_2(\hat{\boldsymbol{\Sigma}}) & \\ & & \hat{P}_3(\hat{\boldsymbol{\Sigma}}) \end{bmatrix}$$

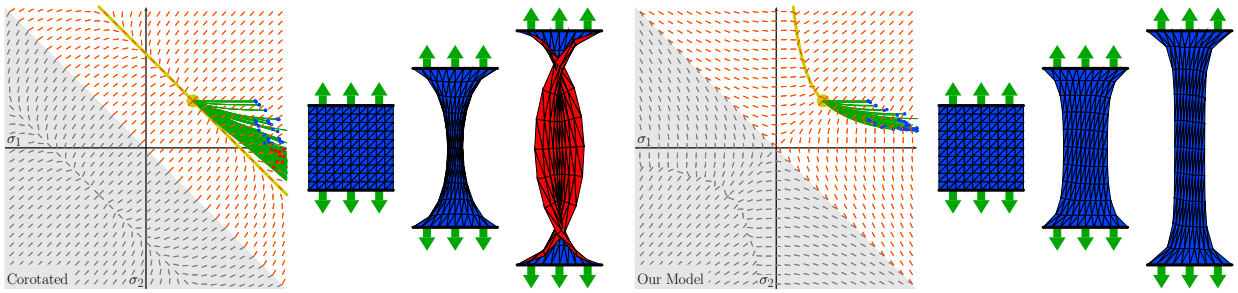

Figure 6.1: A 2D mattress is stretched by two sides and the evolution of its elements is shown in the principal stretches space. The arrows (orange) show the downhill direction of the energy gradient. The gray region is invalid in accordance with the IFE convention [40]. The green curves show singular value trajectories from the undeformed configuration $(1, 1)$ (yellow dot) to the final configuration (colored dots) for each element in the mesh as it is stretched. The trajectories tend to follow the primary contour (yellow). The corotated model primary contour crosses the axes, leading to nonphysical inversion (red) for sufficiently large stretches.

with $\hat{P}_i = \hat{\Psi}_{\sigma_i} = \frac{\partial \hat{\Psi}}{\partial \sigma_i}$. Furthermore, the linearization of the stress around a given $\boldsymbol{F}$ is $\delta \boldsymbol{P} = \frac{\partial \boldsymbol{P}}{\partial \boldsymbol{F}}(\boldsymbol{F}) : \delta \boldsymbol{F}$ and in the case of isotropy, this can be shown to satisfy $\delta \boldsymbol{P} = \boldsymbol{U}\left(\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{F}}(\boldsymbol{\Sigma}) : \left(\boldsymbol{U}^T \delta \boldsymbol{F} \boldsymbol{V}\right)\right) \boldsymbol{V}^T$. Although the term $\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{F}}(\boldsymbol{\Sigma})$ was shown to have a block diagonal structure in terms of the invariants of $\boldsymbol{F}$ in [86], we prefer to express this block structure in terms of the principal stretches as in [81]. If we reorder the $3 \times 3 \times 3 \times 3$ fourth order tensor $\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{F}}(\boldsymbol{\Sigma})$ as a 9×9 matrix using the convention that a 3×3 matrix $\boldsymbol{S}$ is reordered as a 9-vector with components $(s_{11}, s_{22}, s_{33}, s_{12}, s_{21}, s_{13}, s_{31}, s_{23}, s_{32})$, then $\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{F}}(\boldsymbol{\Sigma})$ can be shown (see Appendix B) to have the four diagonal blocks $\boldsymbol{A}$, $\boldsymbol{B}_{12}$, $\boldsymbol{B}_{13}$ and $\boldsymbol{B}_{23}$ with

$$\boldsymbol{A} = \begin{bmatrix} \hat{\Psi}_{\sigma_1\sigma_1} & \hat{\Psi}_{\sigma_1\sigma_2} & \hat{\Psi}_{\sigma_1\sigma_3} \\ \hat{\Psi}_{\sigma_2\sigma_1} & \hat{\Psi}_{\sigma_2\sigma_2} & \hat{\Psi}_{\sigma_2\sigma_3} \\ \hat{\Psi}_{\sigma_3\sigma_1} & \hat{\Psi}_{\sigma_3\sigma_2} & \hat{\Psi}_{\sigma_3\sigma_3} \end{bmatrix}$$

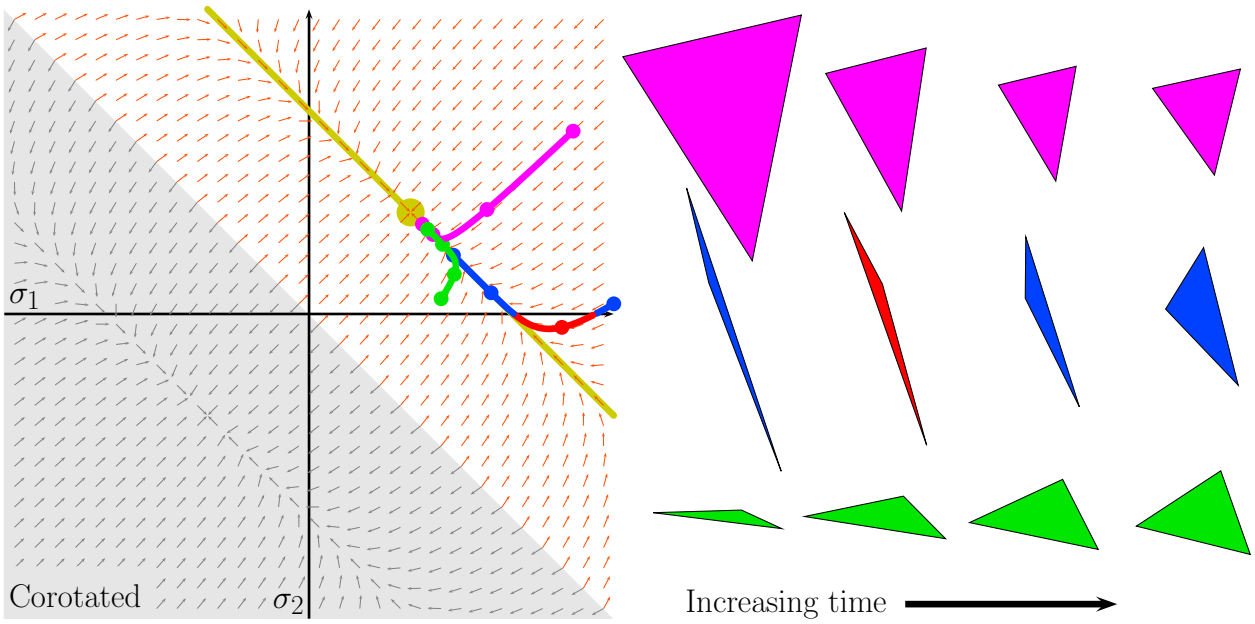

Figure 6.2: Three triangles are allowed to relax from different initially deformed configurations. Their trajectories in singular value space are shown with corresponding colors on the left. Note the strong attraction to the primary contour (yellow) causes the blue triangle to invert (shown in red).

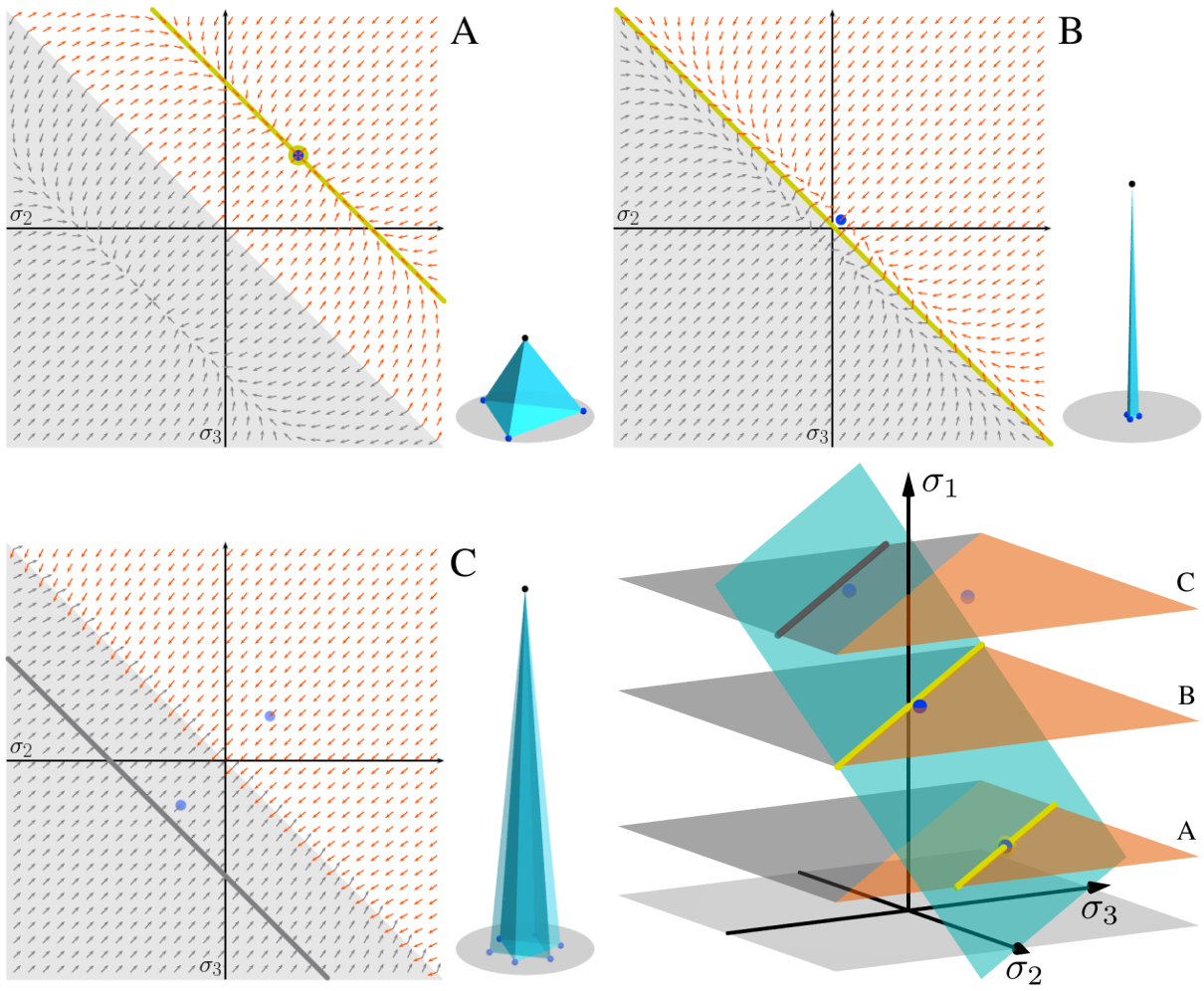

Figure 6.3: We stretch a tetrahedron with its base fixed to a plane. The plots show the corotated energy gradients in constraint planes of increasing σ_1. The equilibrium singular values are shown in blue. The yellow line is the intersection of the constraint plane with the primary contour. As the tetrahedron is stretched, the line shifts. The primary contour does not intersect the third slice, but we show in dark gray where it would intersect if it were extended into the invalid region (gray). The primary contour draws the configuration into a minimum at the energy kink, leading to nonphysical oscillation. For illustrative purposes we slightly abused the IFE convention on the constraint plane C to demonstrate how the configuration is driven towards the kink, although in fact it never can get to the invalid region and keeps "bouncing back."

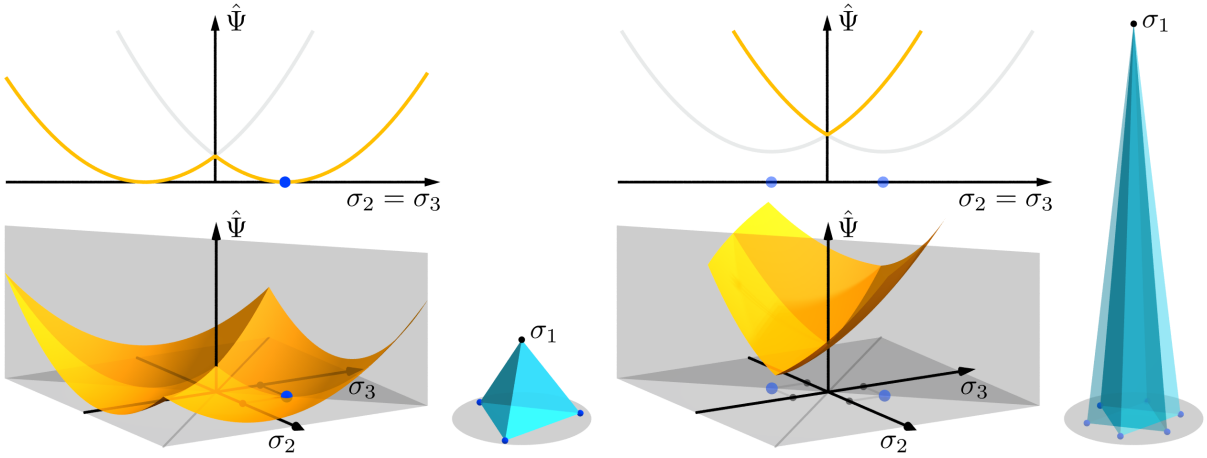

Figure 6.4: We plot the energy surface $\hat{\Psi}(c, \sigma_2, \sigma_3)$ for the example in Figure 6.3 for $c = 1$ (left) and $c = 5$ (right). The plots on top show the energy profile along the line $\sigma_2 = \sigma_3$, which is orthogonal to the kink at $\sigma_2 + \sigma_3 = 0$. The blue dots show the quasistatic solutions that would be obtained assuming a smooth energy profile. In the image at the left, the minimizer is away from the energy kink. However, as the top vertex is stretched the minimum approaches the kink which leads to non-physical oscillations. For illustrative purposes we slightly abused the IFE convention on the right side as in Figure 6.3.

and

$$\boldsymbol{B}_{ij} = \frac{1}{\sigma_i^2 - \sigma_j^2} \begin{bmatrix} \sigma_i \hat{\Psi}_{\sigma_i} - \sigma_j \hat{\Psi}_{\sigma_j} & \sigma_j \hat{\Psi}_{\sigma_i} - \sigma_i \hat{\Psi}_{\sigma_j} \\ \sigma_j \hat{\Psi}_{\sigma_i} - \sigma_i \hat{\Psi}_{\sigma_j} & \sigma_i \hat{\Psi}_{\sigma_i} - \sigma_j \hat{\Psi}_{\sigma_j} \end{bmatrix}.$$

As in [81], care must be taken to robustly treat the possibly small denominators in the components of $\boldsymbol{B}_{ij}$. These expressions for the stress and stress derivatives are used to compute forces and their linearizations directly from our extended hyperelastic energy densities in all configurations (inverted or otherwise). As in [86], the one 3×3 and three 2×2 matrices can be readily projected to their nearest SPD counterpart to guarantee that conjugate gradient can reliably be used for solving the discrete systems that arise with implicit time stepping.

6.2 Invertible Hyperelastic Energy Densities

We propose the invertible extension of $\hat{\Psi}$, rather than the extension of its derivatives $\hat{\boldsymbol{P}}$ as was originally advocated in [40]. That is, these functions are extended to the portion of singular value space where singular values can be negative. In general, an extension of $\hat{\boldsymbol{P}}$ is not guaranteed to be consistent with a hyperelastic strain energy density, and the procedural modification complicates the definition of stress derivatives needed with implicit time stepping. In fact, [86] were forced to evaluate stress derivatives in a nearby uninverted configuration which lead to inconsistency between the stress and its derivatives. We will show that an energetic extension of the constitutive model is far more simplistic, robust and stable than the original stress based extension. However, we first discuss some fundamental properties of hyperelastic energy densities defined over the inverted portion of singular value space.

6.2.1 Energy Kinks

As previously mentioned, isotropy implies that the energy density $\hat{\Psi}$ is invariant under permutations of the singular values. Isotropy is only one source of symmetry in models defined over inverted configurations.

Standard SVD convention dictates that singular values are always nonnegative. However, in order for the $\boldsymbol{U}$ and $\boldsymbol{V}$ matrices to correspond to rotations some singular values might need to be negated. This brings in non-uniqueness in the sense that we are free to choose which singular values get the negative sign. We resolve this with the IFE convention [40] and negate, if needed, the one with the smallest magnitude. As a result, the combinations of singular values that do not obey this convention cannot possibly occur. They form an *invalid region* in the principal stretch space, which we show in gray in Figures 6.1, 6.2 and 6.3. Although such combinations will never be computed, we can consider the energy density as being defined over these excluded combinations by ensuring that combinations corresponding to the same deformation gradient are assigned the same energy densities. This

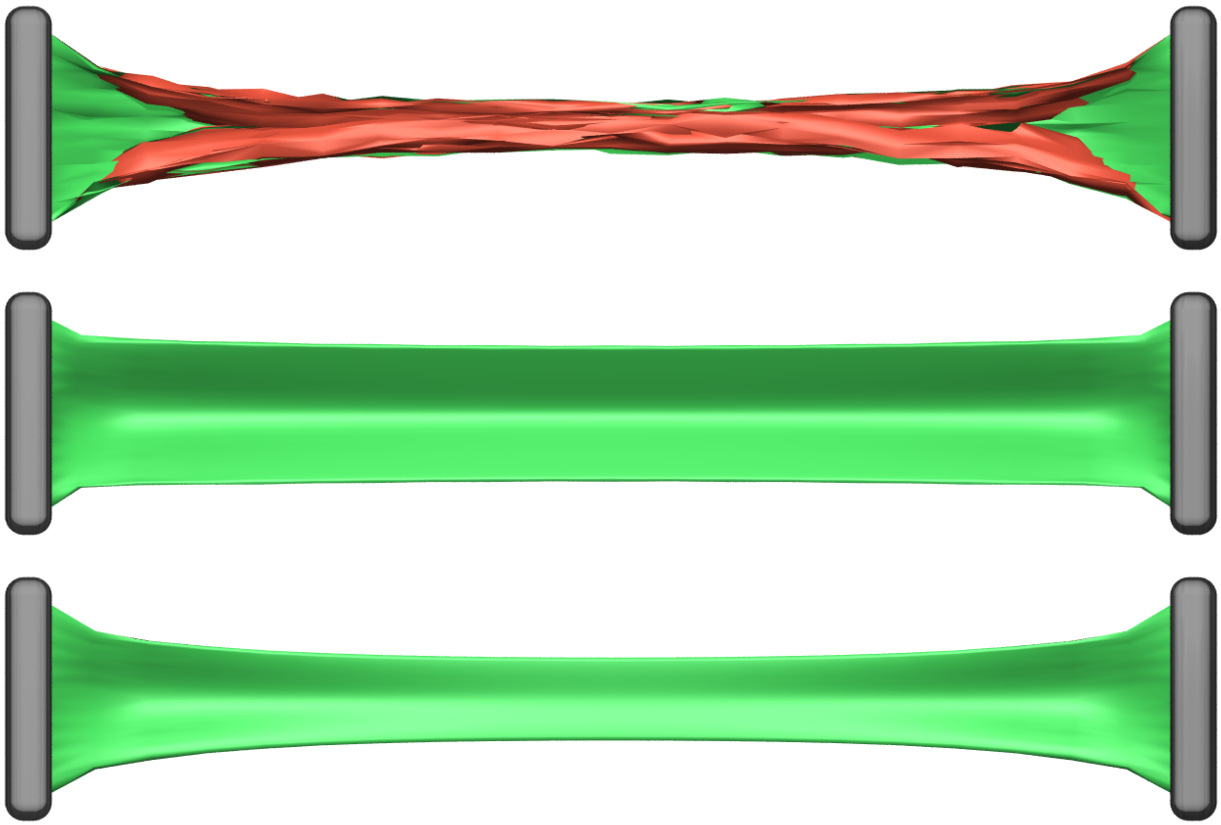

Figure 6.5: Comparison of our C^2 Neo-Hookean-based model (bottom) with corotational elasticity (top) and corotational with our fix (middle). The corotational model is unstable under these stretched configurations, and many inverted elements arise (shown in red). Our fix to the corotational model prevents the instability and inversion but does not look as realistic (the cross section resembles an X) as our C^2 model. Note that unmodified Neo-Hookean will produce the same result as our extension, since the extrapolation threshold was not reached in this example.

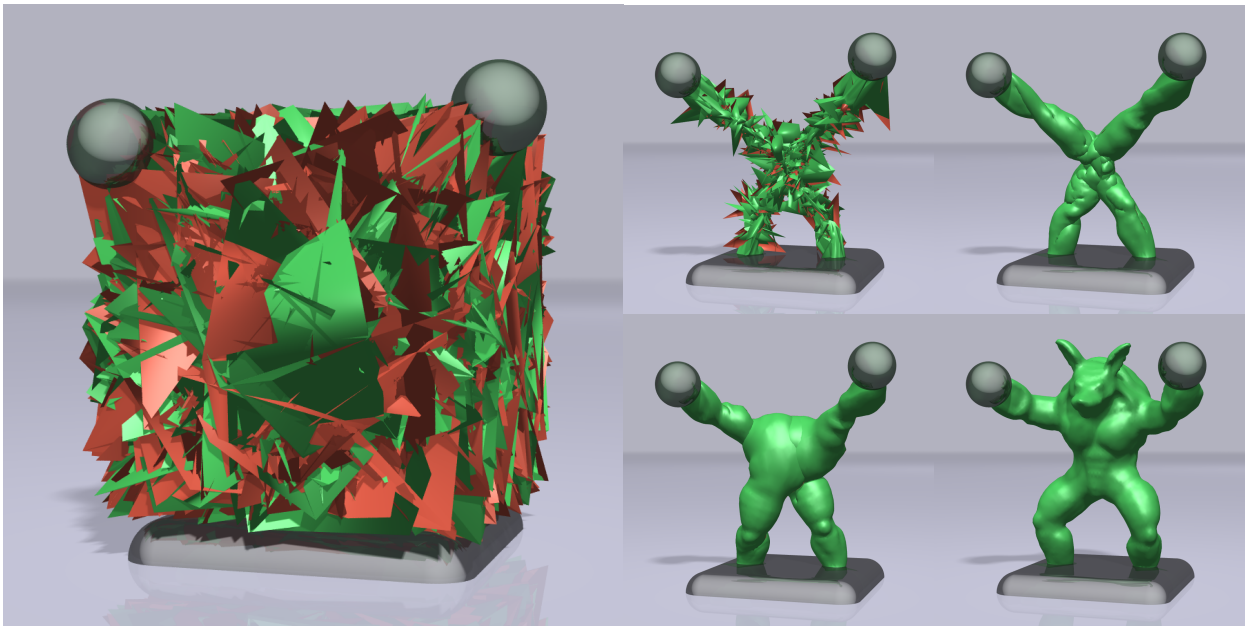

Figure 6.6: We randomly scatter the vertices of an armadillo mesh and let it relax to rest. Our energy based approach robustly handles extremely large deformations with many severely deformed and inverted elements (shown in red).

enforces invariance under pairs of singular value sign flips and results in a second form of symmetry.

The aforementioned symmetries can lead to a kink in the energy density. For example in 2D if we negate the singular values σ_1 and σ_2, then we must have $\hat{\Psi}(\sigma_1, \sigma_2) = \hat{\Psi}(-\sigma_1, -\sigma_2)$ since these configurations correspond to the same deformation gradient. Furthermore, if we then permute these values we must have $\hat{\Psi}(\sigma_1, \sigma_2) = \hat{\Psi}(-\sigma_2, -\sigma_1)$. Consider the energy along the line $\sigma_1(t) = s + t$ and $\sigma_2(t) = -s + t$, where s is arbitrary, but fixed. Then, $\psi(t) = \hat{\Psi}(s + t, -s + t) = \hat{\Psi}(s - t, -s - t) = \psi(-t)$. Then, either $\psi'(0) = 0$ or $\psi(t)$ has a kink at $t = 0$.

Note that $\psi'(0)$ is the component of the stress orthogonal to the line $\sigma_1 + \sigma_2 = 0$ at the point $(s, -s)$. Therefore, any energy density that leads to a nonzero orthogonal stress contribution at the line $\sigma_1 + \sigma_2 = 0$ must have a kink there. Since the orthogonal stress component is required to leave this line, a kink in the energy density profile is actually

desirable. Otherwise, the model would be inherently compliant to inversion with a weak restoring force near this line. Note that energy densities defined in terms of the invariants tend not to produce such a kink. Indeed, consider an energy defined in terms of the invariants $\hat{\Psi}(\sigma_1, \sigma_2) = \tilde{\Psi}(I_1, J) = \tilde{\Psi}(\sigma_1^2 + \sigma_2^2, \sigma_1\sigma_2)$. The component of the stress orthogonal to the line $\sigma_1 + \sigma_2 = 0$ is given as $\hat{\Psi}_{\sigma_1} + \hat{\Psi}_{\sigma_2}$. We can then see from the invariants that $\hat{\Psi}_{\sigma_1} + \hat{\Psi}_{\sigma_2} = (2\sigma_1\tilde{\Psi}_{I_1} + \sigma_2\tilde{\Psi}_{I_2}) + (2\sigma_2\tilde{\Psi}_{I_1} + \sigma_1\tilde{\Psi}_{I_2}) = (\sigma_1 + \sigma_2)(2\tilde{\Psi}_{I_1} + \tilde{\Psi}_{I_2}) = 0$ when $\sigma_1 + \sigma_2 = 0$, provided the partials in the invariants remain bounded. The situation is analogous in 3D with the kink arising along the plane $\sigma_2 + \sigma_3 = 0$. See Figures 6.4 and 6.9 (right) for visualizations of the kinks.

We will show that despite the fact that the kink arises only at the boundary of the valid region, it still plays a fundamental role in the behavior of the model.

6.2.2 Primary Contour

Hyperelastic constitutive models are characterized by a strongly attractive basin which we call the *primary contour* of the model. For example, a model with Poisson's ratio very close to $\frac{1}{2}$ may be more strongly attracted to the submanifold corresponding to volume preservation than to the rest configuration. Let $\boldsymbol{v}$ be the eigenvector of the energy Hessian $H_{ij} = \frac{\partial^2\hat{\Psi}}{\partial\sigma_i\partial\sigma_j}$ with the largest-magnitude eigenvalue, and let $g_i = \frac{\partial\hat{\Psi}}{\partial\sigma_i}$ be the energy gradient. We define the primary contour to be the region where $\boldsymbol{v}\cdot\boldsymbol{g} = 0$. The Hessian describes how $\boldsymbol{g}$ changes, and $\boldsymbol{v}$ describes the direction of greatest change of $\boldsymbol{g}$. Thus, as the configuration moves from the primary contour, the $\boldsymbol{v}$ component of $\boldsymbol{g}$ will dominate. This tends to draw the configuration towards the primary contour. When it gets close, the $\boldsymbol{v}$ component diminishes, and the configuration moves mostly along the contour. This phenomenon is shown in Figures 6.1 and 6.2. Problems arise when the contour crosses over into the inverted regime or when it approaches a kink in the energy.

Failure to define a model with an appropriate primary contour can lead to catastrophic behavior, which we demonstrate in the following section.

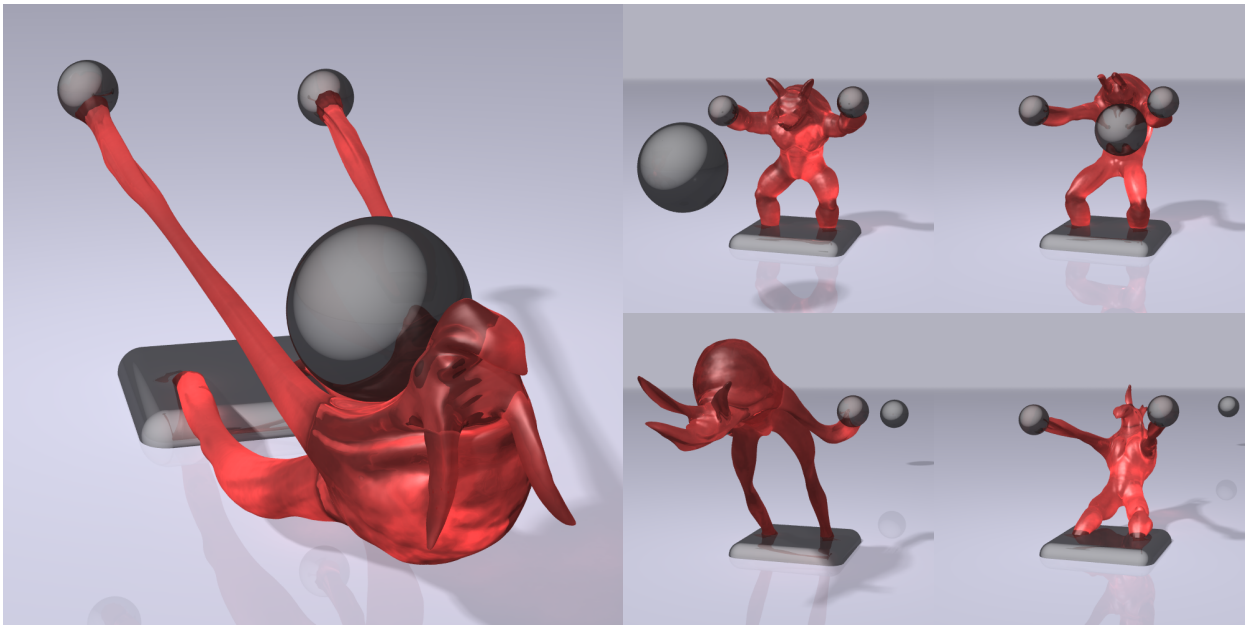

Figure 6.7: **Ouch.** An armadillo is hit with a ball.

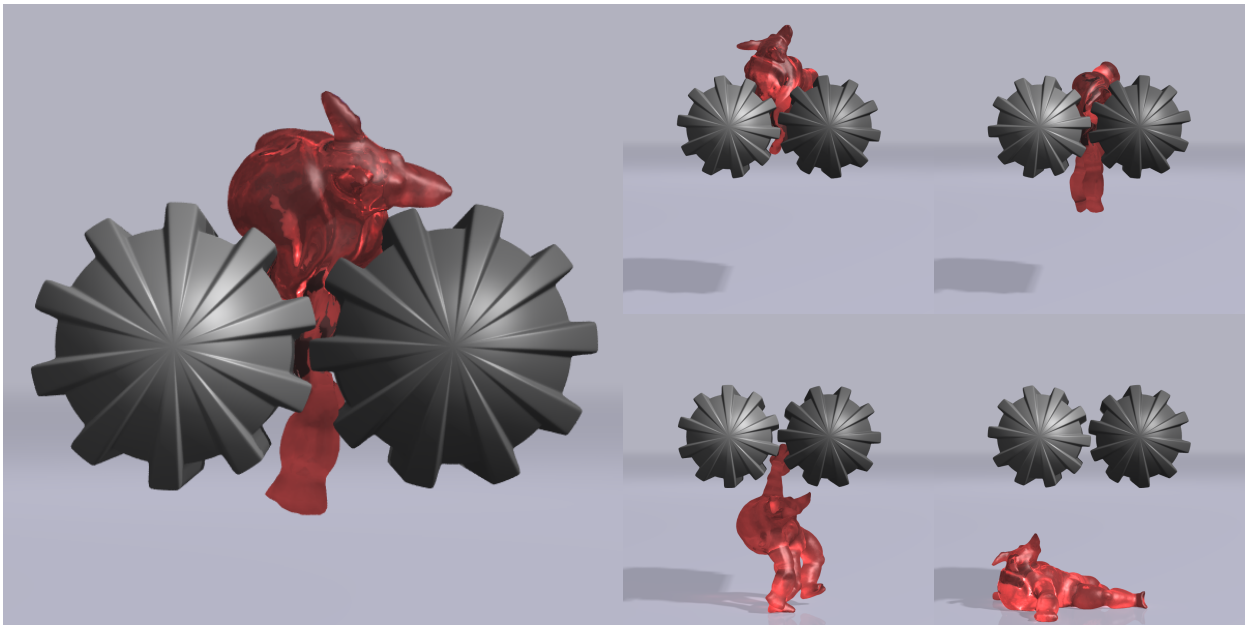

Figure 6.8: **Ouch again.** An armadillo is passed through gears.

6.2.3 Corotational Hyperelasticity

Models designed to correct the rotational artifacts inherent in linear elasticity are very popular in computer graphics [61, 76, 99, 62, 23]. A well-known example is a hyperelastic corotated model [13, 56], which takes the form

$$\hat{\Psi} = \mu \sum_i (\sigma_i - 1)^2 + \frac{\lambda}{2}\Big(\sum_i (\sigma_i - 1)\Big)^2.$$

The gradient is $g_i = 2\mu(\sigma_i - 1) + \lambda \sum_j (\sigma_j - 1)$, and the Hessian is $H_{ij} = 2\mu\delta_{ij} + \lambda$. Its largest eigenvector is $v_i = 1$ with eigenvalue $2\mu + d\lambda$, where d is the dimension. The other eigenvalues are 2μ with eigenvectors orthogonal to $\boldsymbol{v}$. Finally, $\boldsymbol{v} \cdot \boldsymbol{g} = 0$ implies $\sum_i (\sigma_i - 1) = 0$ is the equation for the primary contour. Note that this primary contour crosses into the inverted region.

Consider the 2D examples shown in Figure 6.2, which shows material relaxation from three distinct initial configurations. Note that all three trajectories initially tend towards the primary contour. Unfortunately, the trajectory highlighted in blue passes through the inverted region on its way to the primary contour. This behavior is also observed on a macroscopic scale and under mesh refinement, as shown in Figure 6.1.

The corotated primary contour is also problematic because it intersects the kink in the energy density in 3D. Consider the behavior of the single tetrahedron shown in Figure 6.3 as it undergoes a stretching deformation. The intersection of the primary contour and the plane $\sigma_1 = c$ is shown at different stages. Note that the relevant slice of the primary contour eventually drives the configuration towards the invalid state that violates the sign convention. This ultimately leads to an energy minimum that lies on a kink, as shown in Figure 6.4. The discontinuity of the stress at this minimum causes nonphysical oscillations, the problematic consequences of which are shown on a macroscopic scale in Figure 6.5.

6.2.4 Corotational Correction

Note that the corotated primary contour is determined primarily by the λ term. Further, the corrective behavior of the corotated model can largely be attributed to its μ term. This suggests that a more suitable model can be constructed by replacing the λ term with one that leads to a more favorable primary contour. One such model is

$$\hat{\Psi} = \mu \sum_i (\sigma_i - 1)^2 + \frac{\lambda}{2}(J-1)^2. \tag{6.1}$$

Similar λ terms were used in [70, 89]. This model has the primary contour $J = 1$, which does not intersect the inverted region. While the model still has limitations, it fixes the stretching problems of the corotated model (see Figure 6.5).

6.3 Energy Extrapolation

We provide both C^1 and C^2 extensions to arbitrary isotropic energy densities $\hat{\Psi}$. We show that these extensions can produce models with well-behaved primary contours. The C^1 extension is easier to implement and results in continuous stresses but discontinuous stress derivatives. The C^2 extension has continuous stress and stress derivatives and provides added robustness in some scenarios. Our extension to $\hat{\Psi}$ is accomplished by polynomial extrapolation from a convex contour in the uninverted portion of singular value space that increases the energy density as the configuration inverts. We present only the energy densities below; see Appendix B for derivatives and more details.

6.3.1 C^1 Extension

We define the C^1 extended energy density $\hat{\Psi}^{\text{ext}}$ to coincide with the original $\hat{\Psi}$ whenever the singular values are all above a threshold ϵ. This region is illustrated in blue in the left image of Figure 6.9 for a 2D problem. If one singular value σ_i is less than this threshold (region shown in red), then we extend the energy quadratically in the direction σ_i from the closest point in the blue region. Consider the point $(\bar{\sigma}_1, \bar{\sigma}_2)$ in Figure 6.9. In this case, only $\bar{\sigma}_1$ is below ϵ,

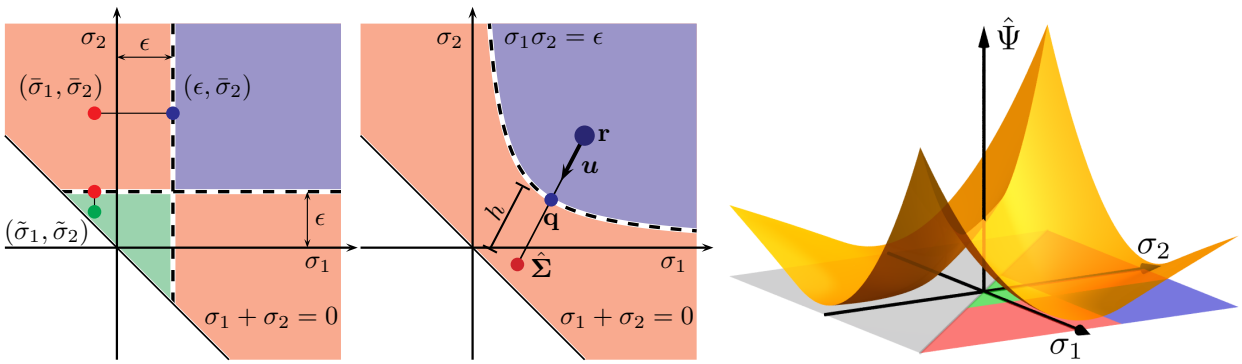

Figure 6.9: The leftmost images show the relevant regions in the 2D C^1 and C^2 extensions respectively. The C^1 $\hat{\Psi}^{\text{ext}}$ is shown at the right. Note the kink discussed in Section 6.2.1 along the line $\sigma_1 + \sigma_2 = 0$. The C^2 $\hat{\Psi}^{\text{ext}}$ looks similar.

and the extension is given as $\hat{\Psi}^{\text{ext}}(\bar{\sigma}_1, \bar{\sigma}_2) = \hat{\Psi}(\epsilon, \bar{\sigma}_2) + \hat{\Psi}_{\sigma_1}(\epsilon, \bar{\sigma}_2)[\bar{\sigma}_1 - \epsilon] + \frac{k}{2}[\bar{\sigma}_1 - \epsilon]^2$. This extension will be C^1 as we transition from the blue region to the red region for all values of the parameter k. This parameter is used to add extra resistance to inversion but does not degrade the C^1 regularity of the extension. For points $(\tilde{\sigma}_1, \tilde{\sigma}_2)$ with both principal stretches below the threshold we define another region (shown in green) and quadratically extend the energy from the nearest point in the red region. It does not matter which red region we extend from, since in both cases we have $\hat{\Psi}^{\text{ext}}(\tilde{\sigma}_1, \tilde{\sigma}_2) = \hat{\Psi}(\epsilon, \epsilon) + \hat{\Psi}_{\sigma_1}(\epsilon, \epsilon)[\bar{\sigma}_1 - \epsilon] + \hat{\Psi}_{\sigma_2}(\epsilon, \epsilon)[\bar{\sigma}_2 - \epsilon] + \hat{\Psi}_{\sigma_1\sigma_2}(\epsilon, \epsilon)[\bar{\sigma}_1 - \epsilon][\bar{\sigma}_2 - \epsilon] + \frac{k}{2}\left([\bar{\sigma}_1 - \epsilon]^2 + [\bar{\sigma}_2 - \epsilon]^2\right)$. The 3D extension is analogous but with an additional type of region. Note that we can avoid the SVD if $I_1/J^2 \geq \epsilon$ (2D) or $I_2/J^2 \geq \epsilon$ (3D), since these imply that $\sigma_i > \epsilon$.

6.3.2 C^2 Extension

We use a different extrapolation surface for our C^2 model, since the one used in C^1 leads to an unfavorable primary contour when extended to C^2. We define the C^2 extension whenever the determinant of the deformation gradient (or product of singular values) is below a threshold ϵ. The base energy density $\hat{\Psi}$ is extended to the extrapolated $\hat{\Psi}^{\text{ext}}$ at a given point $\hat{\boldsymbol{\Sigma}} = (\sigma_1, \sigma_2, \sigma_3)$ by extrapolating along the line to the rest configuration point $\boldsymbol{r} = (1, 1, 1)$.

Extrapolation begins at the intersection of the line and the contour surface $\sigma_1\sigma_2\sigma_3 = \epsilon$ and is chosen so that all first and second derivatives of the extended energy match those of $\hat{\Psi}$ at the intersection point (see Figure 6.9).

Let $\boldsymbol{u} = \frac{\boldsymbol{r}-\hat{\boldsymbol{\Sigma}}}{|\boldsymbol{r}-\hat{\boldsymbol{\Sigma}}|}$ be the direction from the point $\hat{\boldsymbol{\Sigma}}$ to the rest configuration point $\boldsymbol{r}$. Denote the intersection between the line and the threshold surface as $\boldsymbol{q} = \boldsymbol{r} + s\left(\hat{\boldsymbol{\Sigma}} - \boldsymbol{r}\right)$, where the scalar s is given by the roots of the cubic equation $q_1q_2q_3 = \epsilon$. If we denote the distance from $\hat{\boldsymbol{\Sigma}}$ to $\boldsymbol{q}$ by h, then the extended energy has the form

$$\hat{\Psi}^{\text{ext}}(\hat{\boldsymbol{\Sigma}}) = \hat{\Psi}(\boldsymbol{q}) + hu_i\hat{\Psi}_{\sigma_i}(\boldsymbol{q}) + \frac{h^2}{2}u_i\hat{\Psi}_{\sigma_i\sigma_j}(\boldsymbol{q})u_j,$$

where we assume summation on repeated indices. The derivatives of the scalar s needed for stress and stress derivatives can be determined implicitly by differentiating the cubic equation $q_1q_2q_3 = \epsilon$. This $\hat{\Psi}^{\text{ext}}$ gives continuous stress and stress derivatives, but it is complicated by the need for the value and derivatives of the intersection point between the line connecting $\hat{\boldsymbol{\Sigma}}$ to $\boldsymbol{r}$ and the extrapolation contour $\sigma_1\sigma_2\sigma_3 = \epsilon$.

6.4 Examples

We demonstrate our extension methodology with a Neo-Hookean hyperelastic energy density

$$\hat{\Psi} = \frac{\mu}{2}\Big(\sum_i \sigma_i^2 - d\Big) - \mu \ln J + \frac{\lambda}{2}(\ln J)^2.$$

In Figures 6.7, 6.12, 6.13 and 6.14 we show examples run with the C^1 extension using an inversion threshold $\epsilon = 0.4$ and $k = 20 \times E$ where E is the Young's modulus. In general, smaller values of ϵ and larger values of k will increase resistance to extreme compression. In Figures 6.5, 6.6 and 6.8 we show examples run with our C^2 extension with threshold surface $J = \epsilon$ and $\epsilon = 0.9$. As with C^1, smaller values of ϵ increase resistance to extreme compression. Although $\epsilon = 0.9$ is somewhat large, smaller values of ϵ resulted in unnecessarily stiff response to compression due to the high energy barrier of the underlying Neo-Hookean constitutive model.

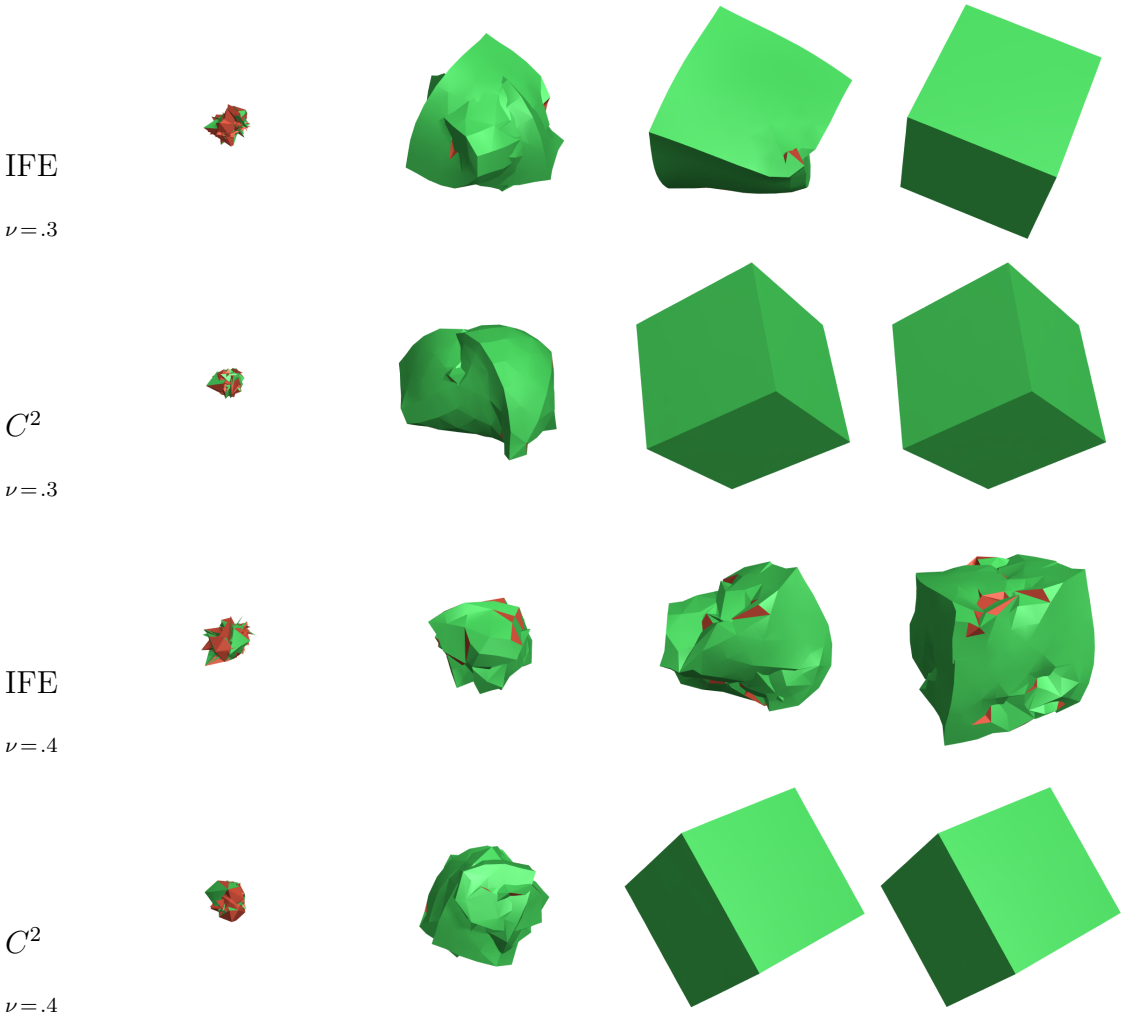

Figure 6.10: In these stress test we initially perturb the vertices of a cube mesh to a point and allow it to recover. Our model resolves itself for a wide range of Lamé coefficients, while a typical IFE implementation fails to recover for large Poisson's ratios (ν).

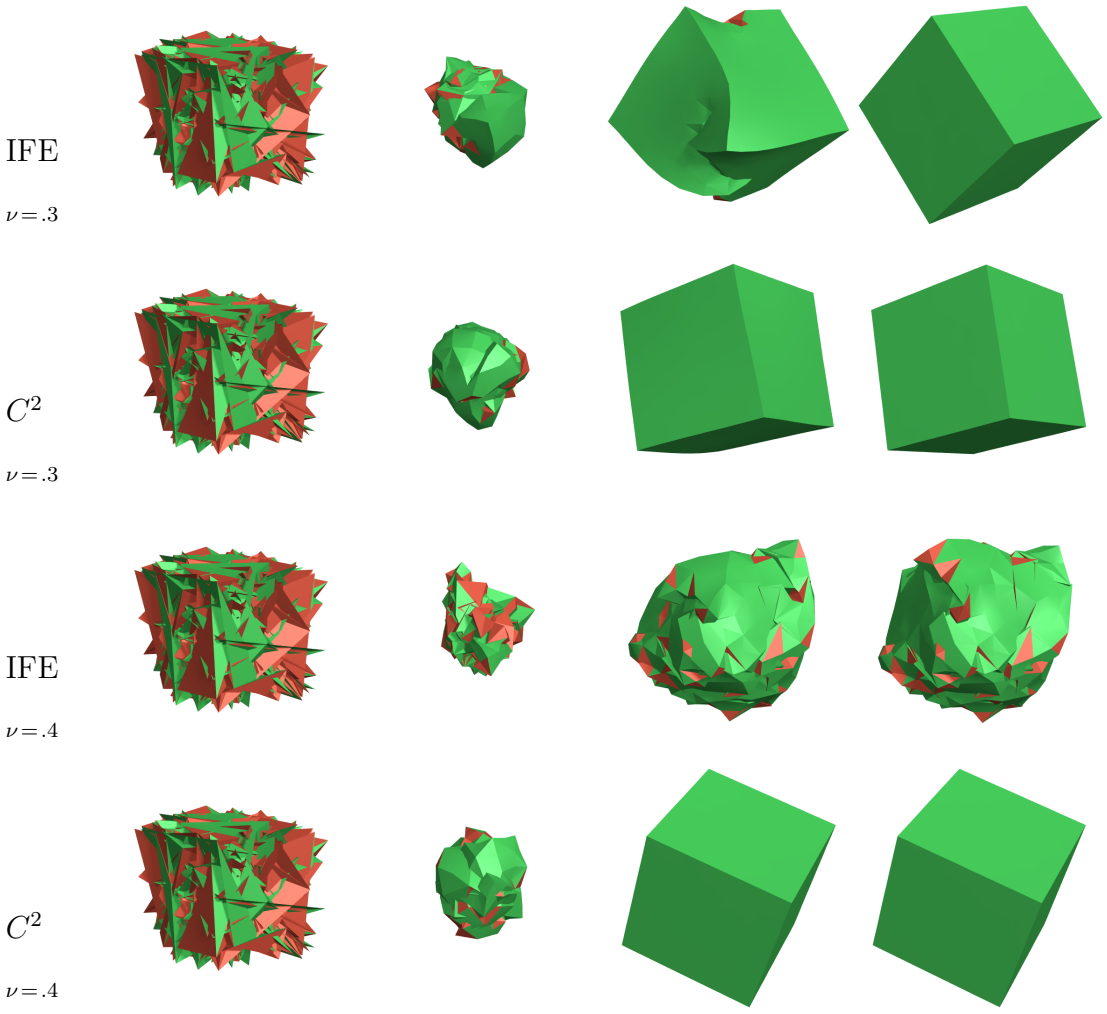

Figure 6.11: In these stress test we randomly perturb the vertices of a cube mesh and allow it to recover. Our model resolves itself for a wide range of Lamé coefficients, while a typical IFE implementation fails to recover for large Poisson's ratios (ν).

6.4.1 Comparison with IFE

Figures 6.10 and 6.11 show a comparison of our C^2 model with IFE using implicit time integration on two common stress tests. In each case, our model performed significantly better with the same Lamé parameters. The IFE extension of the first Piola-Kirchoff stress used the same threshold contour with linear stress extrapolation and derivative clamping as in [40, 86]. Figure 6.1 (right) shows the primary contour for the 2D equivalent of our C^2 model. Note that the contour never extends into the inverted region. Furthermore the figure shows that the curve is a good predictor of the average trajectory of an element in the mesh. Because IFE lacks an energy in the extrapolated region, its primary contour cannot be used to predict its behavior there. These results suggest that our hyperelastic extension methodology allows us to readily design constitutive models that are more robust to extremely large deformation.

6.4.2 Comparison with Corotated

The primary contours for our model and for corotational elasticity are shown in Figure 6.1. Notice that the corotational elasticity primary contour intersects the inverted region and ours does not. This feature prevents the tendency towards inverted equilibrium configurations and it also prevents instabilities that arise when the model drives the configuration towards the energy kink. Figure 6.5 illustrates the consequences of this behavior in a large scale example in 3D. The corotated model is driven towards the inverted region and to unstable minima at the energy kink. This leads to non-physical oscillation and inversion (shown in red). This behavior is prevented with our fix to the λ term in the corotational mode, however it still does not look as realistic as our Neo-Hookean-based extension.

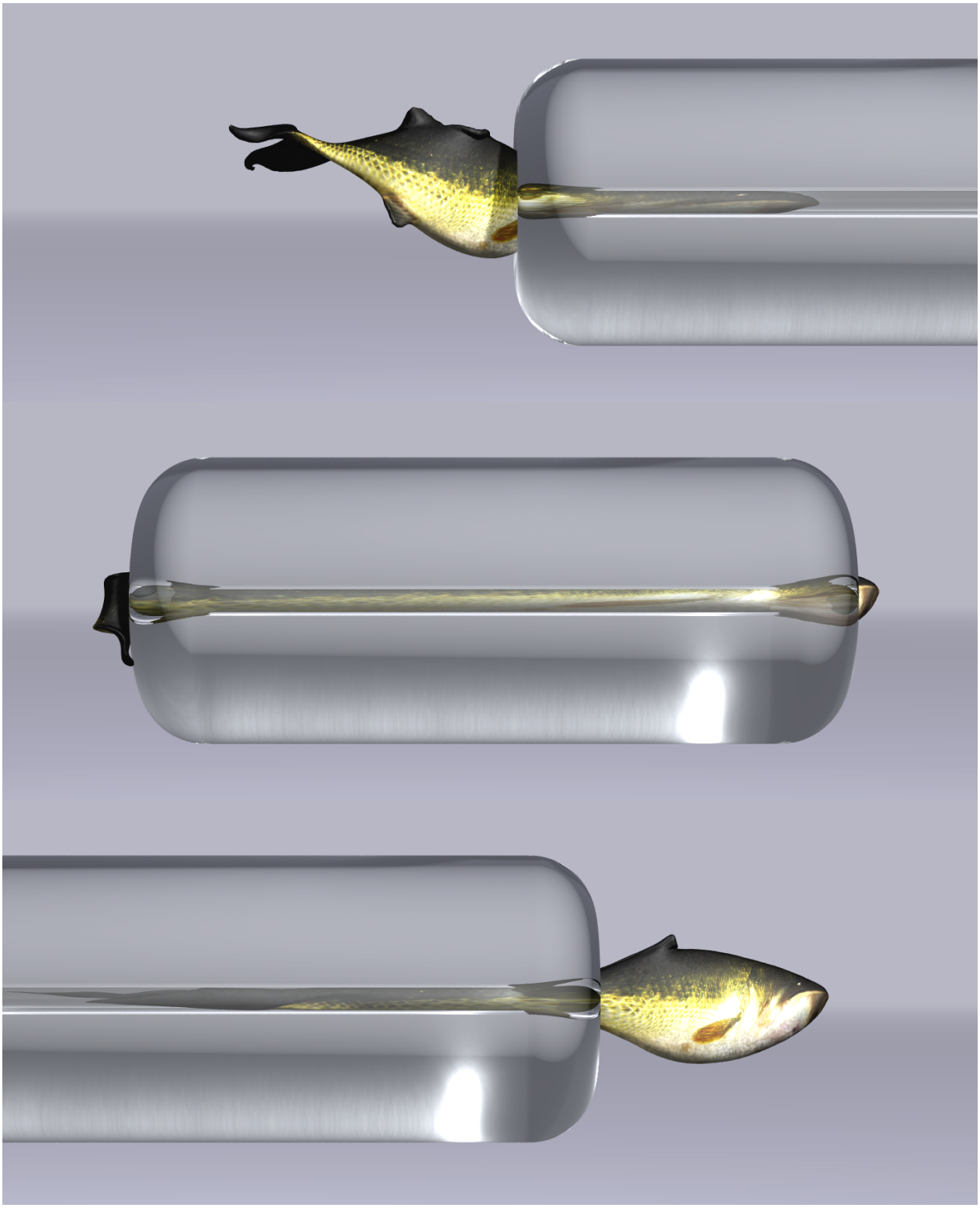

Figure 6.12: **Tight spaces.** A fish passing through a thin tube.

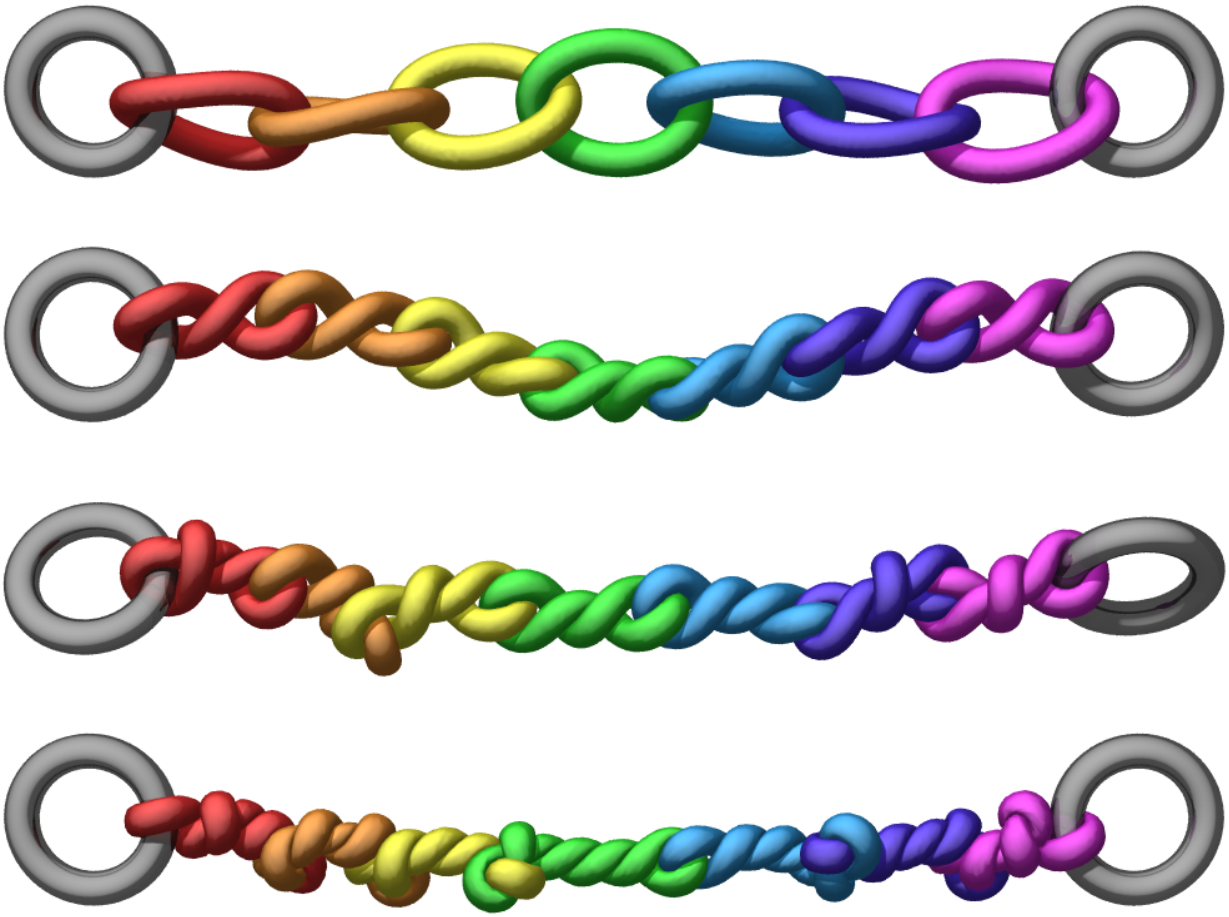

Figure 6.13: **That's twisted.** 7 elastic links in a braiding example.

Figure 6.14: **Yummy.** 25 gelatin cubes falling in a bowl.

CHAPTER 7

A Material Point Method for Snow Simulation

7.1 Overview

As discussed in the introduction, snow dynamics modeling is difficult due to snow's variability, usually stemming from environmental factors (freshness, water/ice content, etc.). Our model ignores root causes, and we instead concentrate on deriving an empirical model based on phenomenological observations. Even so, our snow constitutive model is based on theory and models devised for engineering applications. The model is kept efficient, allowing us to capture sufficient geometric detail with tractable computation time.

The material point method is the center of our technique. At its core, MPM relies on the *continuum approximation*, avoiding the need to model every snow grain. While an MPM method typically uses a Cartesian grid to make topology changes and self-collisions automatic, it outperforms purely Eulerian methods by tracking mass (and other conserved

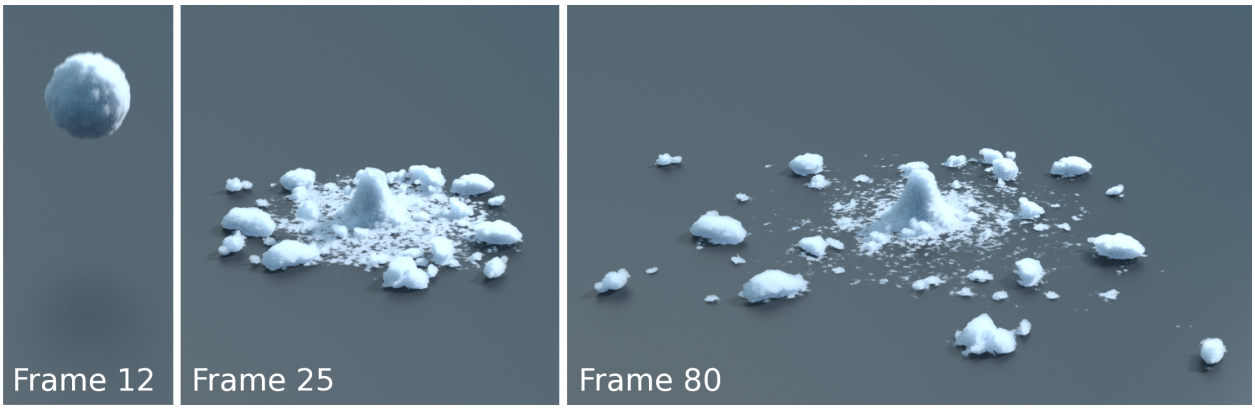

Figure 7.1: **Snowball drop.** A basic snowball hitting the ground. © Disney

Table 7.1: Comparison between various methods of simulation on four properties that are important for snow.

Method	**Volume Preservation**	**Stiffness**	**Plasticity**	**Fracture**
Reeve particles	-	-	-	-
Rigid bodies	⋆⋆	⋆⋆	-	⋆
Mesh-based solids	⋆	⋆⋆⋆	⋆⋆	⋆
Grid-based fluids	⋆⋆⋆	⋆	⋆⋆	⋆⋆⋆
SPH	⋆	⋆	⋆	⋆⋆⋆
MPM	⋆⋆	⋆⋆	⋆⋆⋆	⋆⋆⋆

quantities) through non-dissipative Lagrangian particles (like SPH). Unlike SPH, however, MPM uses the grid as an efficient continuum *scratchpad* which avoids high valence communication patterns derived from nearest-neighbor queries. See Figure 7.4 for an illustration of the interplay between the grid and particles and Section 7.3.1 for details.

Our motivation for choosing MPM is that it is better able to handle the dynamics of snow. This is analogous to how a rigid body simulation is the most efficient way to handle infinite stiffness and incompressibility compared to using a large stiffness directly on a FEM mesh solver. The constitutive properties central to snow are *volume preservation*, *stiffness*, *plasticity*, *fracture*, and we summarize various methods' abilities to handle them in Table 7.1.

Volume preservation in snow is important even though, unlike a liquid, it is compressible. Instead, snow has varying resistance to volume change, which we model in our method similarly to a normal mesh-based solid simulation. Stiffness is also important, and while MPM cannot do this as well as mesh-based elasticity (the deformation gradient is less accurate), it is more effective than grid-based elasticity as the deformation gradient is not dissipative and remains synchronized with positions. Plasticity and fracture are also handled well by MPM, and this is what makes the method desirable for snow simulation. MPM is almost ideal

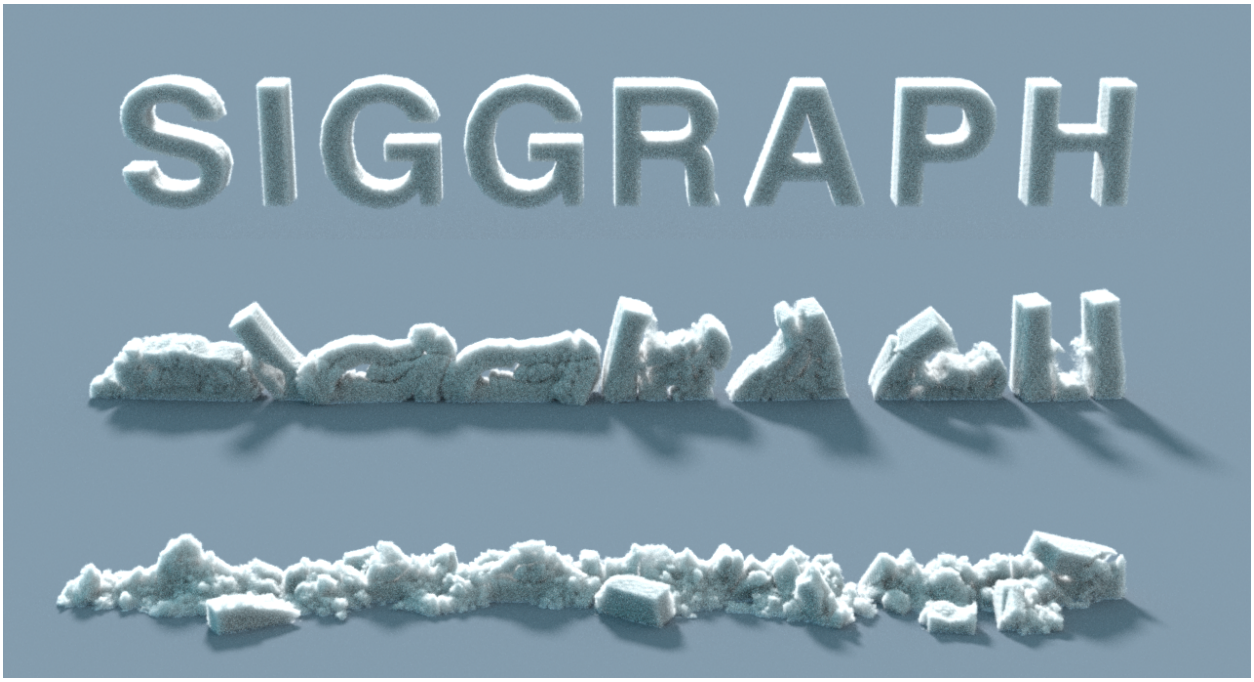

Figure 7.2: **SIGGRAPH.** The MPM method naturally handles fracture, giving us interesting naturalistic shapes. © Disney

for plasticity because the inaccuracies of the deformation gradient accumulate as artificial plasticity. While grid-based fluids work well for plasticity [28], MPM is better at conserving angular momentum. While mesh-based FEM methods can handle plasticity [5], remeshing is required with extreme deformation. By contrast, MPM need only track the unmeshed particles. We note MPM's gains in plasticity and fracture come at the cost of reduced elastic accuracy, a good tradeoff for snow.

7.2 Related work

Geometric snow modeling The bulk of graphics snow research is devoted to modeling accumulation [27, 25, 26, 65]. These techniques can efficiently and accurately create snow-covering effects but neglect treatment of snow dynamics. Some authors have extended these techniques to handle rapid animation and interactions with external objects [72, 97, 12, 55, 34]. Often these methods use simplified modeling primitives like height-fields (e.g. [85]),

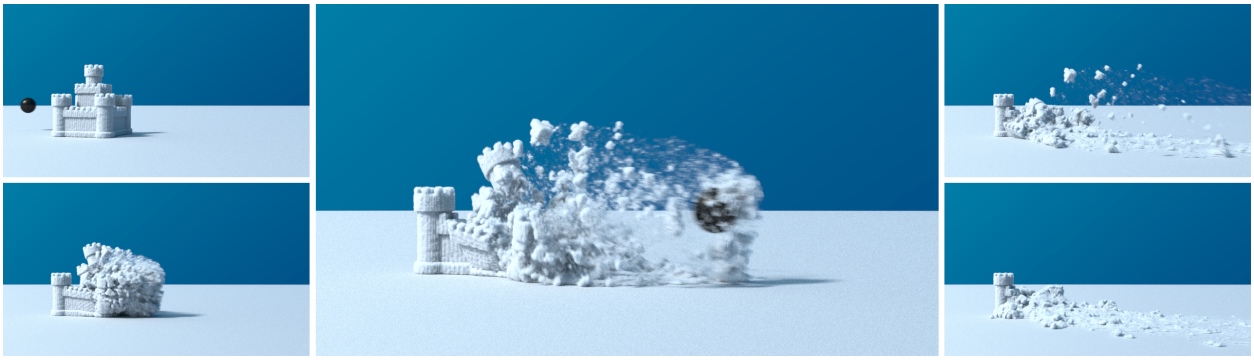

Figure 7.3: **Castle destruction.** Modeled structures like this castle can be destroyed using our method. © Disney

and these are now a popular techniques for games and feature films. Additionally, [46, 44] simulate the related phenomena of ice and frost formation.

Granular materials Snow is often classified as a granular material. Although we are not aware of any graphics paper that specifically simulates snow dynamics, many papers have considered other uses of granular dynamics such as sand. The first graphics papers approximating granular materials consisted of particle or simplified rigid body systems representing each grain [59, 53, 58], and recently researchers have continued to improve such techniques [6, 2]. Even so, the challenge is to retain efficiency as the number of grains increases. Other researchers have successfully applied simplified continuum models [98, 50, 63, 1, 39] to good effect. In particular, [98] introduced a FLIP-based method for simulating sand as an incompressible fluid. Subsequently, [63] modified the incompressibility constraint to avoid cohesion errors. Similarly, the MPM method was designed to extend FLIP to solid mechanics problems that require compressibility.

Elasto-plastic continuum modeling The computer graphics work on elasto-plastic simulation is relevant because of our constitutive model [87, 66, 71, 40, 5, 13, 42, 96, 51, 28]. The Eulerian discretization of elasticity in [51] is particularly relevant as the material point method uses a regular Eulerian grid for discretizing stress-based forces.

Engineering modeling of snow There is extensive engineering literature related to

the modeling and simulation of snow [30]. Although the complex mechanical behavior of snow strongly depends on numerous physical conditions, we found that an elasto-plastic constitutive relation worked well for generating realistic dynamics for a wide range of visual phenomena. This representation, as well as finite-element-based discretization, is very common in the engineering literature [57, 17, 82, 21, 16, 9, 64].

7.3 Material point method

A body's deformation can be described as a mapping from its undeformed configuration $\boldsymbol{X}$ to its deformed configuration $\boldsymbol{x}$ by $\boldsymbol{x} = \boldsymbol{\phi}(\boldsymbol{X})$, which yields the deformation gradient $\boldsymbol{F} = \partial\boldsymbol{\phi}/\partial\boldsymbol{X}$. Deformation $\boldsymbol{\phi}(\boldsymbol{X})$ changes according to conservation of mass, conservation of momentum and the elasto-plastic constitutive relation

$$\frac{D\rho}{Dt} = 0, \quad \rho\frac{D\boldsymbol{v}}{Dt} = \nabla^{\boldsymbol{x}} \cdot \boldsymbol{\sigma} + \rho\boldsymbol{g}, \quad \boldsymbol{\sigma} = \frac{1}{J}\frac{\partial\Psi}{\partial\boldsymbol{F}_E}\boldsymbol{F}_E^T,$$

where ρ is density, t is time, $\boldsymbol{v}$ is velocity, $\boldsymbol{\sigma}$ is the Cauchy stress, $\boldsymbol{g}$ is the gravity, Ψ is the elasto-plastic potential energy density, $\boldsymbol{F}_E$ is the elastic part of the deformation gradient $\boldsymbol{F}$ and $J = \det(\boldsymbol{F})$. We will discuss details of the constitutive model in Section 7.4.

The basic idea behind the material point method is to use particles (material points) to track mass, momentum and deformation gradient. Specifically, particle p holds position $\boldsymbol{x}_p$, velocity $\boldsymbol{v}_p$, mass m_p, and deformation gradient $\boldsymbol{F}_p$. The Lagrangian treatment of these quantities simplifies the discretization of the $\frac{D\rho}{Dt}$ and $\rho\frac{D\boldsymbol{v}}{Dt}$ terms. However, the lack of mesh connectivity between particles complicates the computation of derivatives needed for stress-based force evaluation. This is remedied with the use of a regular background Eulerian grid. Interpolating functions over this grid are used to discretize the $\nabla^{\boldsymbol{x}} \cdot \boldsymbol{\sigma}$ terms in the standard FEM manner using the weak form. We use dyadic products of one-dimensional cubic B-splines as our grid basis functions as in [83]

$$N_{\boldsymbol{i}}^h(\boldsymbol{x}_p) = N(\frac{1}{h}(x_p - ih))N(\frac{1}{h}(y_p - jh))N(\frac{1}{h}(z_p - kh)),$$

where $\boldsymbol{i} = (i, j, k)$ is the grid index, $\boldsymbol{x}_p = (x_p, y_p, z_p)$ is the evaluation position, h is the grid

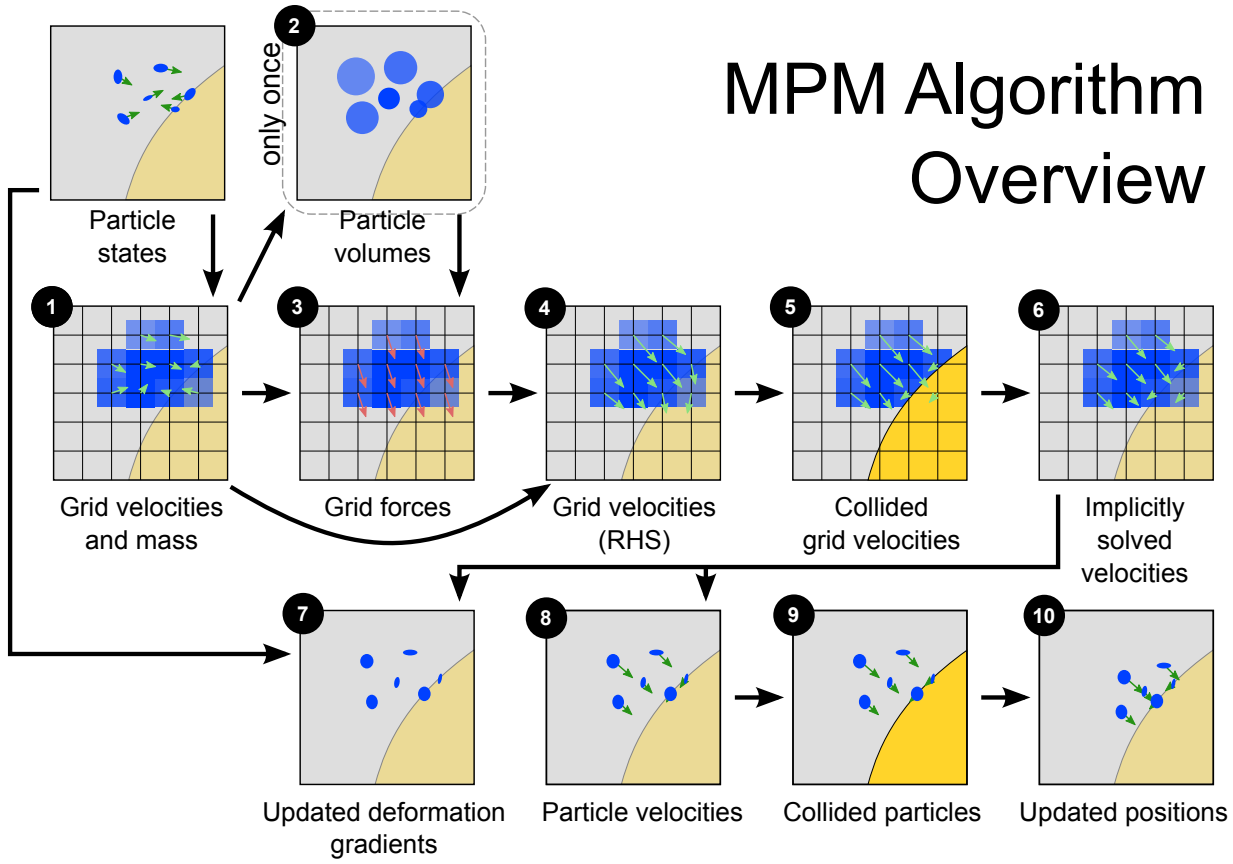

Figure 7.4: An overview of the material point method (MPM). The top and the bottom rows are steps that operate on particles while the middle depicts grid-based operations.

spacing and

$$N(x) = \begin{cases} \frac{1}{2}|x|^3 - x^2 + \frac{2}{3}, & 0 \le |x| < 1 \\ -\frac{1}{6}|x|^3 + x^2 - 2|x| + \frac{4}{3}, & 1 \le |x| < 2 \\ 0, & \text{otherwise} \end{cases} .$$

For more compact notation, we will use $w_{ip} = N_i^h(\boldsymbol{x}_p)$ and $\nabla w_{ip} = \nabla N_i^h(\boldsymbol{x}_p)$. These interpolation functions naturally compute forces at the nodes of the Eulerian grid. Therefore, we must first transfer the mass and momentum from the particles to the grid so that we can update the velocities at the grid nodes. This updated velocity is then transferred back to the particles in either a FLIP or PIC type manner. The transfer process is done using the interpolating weights w_{ip}.

7.3.1 Full method

Here we outline the full update procedure (visually shown in Figure 7.4).

1. **Rasterize particle data to the grid.** The first step is to transfer mass from particles to the grid. The mass is transferred using the weighting functions $m_i^n = \sum_p m_p w_{ip}^n$. Velocity also should be interpolated to the grid, but weighting with w_{ip}^n does not result in interpolation. Instead, we use normalized weights for velocity $\boldsymbol{v}_i^n = \sum_p \boldsymbol{v}_p^n m_p w_{ip}^n / m_i^n$. Note that this conserves the total momentum of the system.

2. **Compute particle volumes and densities.** *First timestep only.* Our force discretization requires a notion of a particle's volume in the initial configuration. We can estimate a cell's density as m_i^0/h^3, which we can weight back to the particle as $\rho_p^0 = \sum_i m_i^0 w_{ip}^0 / h^3$. We can now estimate a particle's volume as $V_p^0 = m_p/\rho_p^0$.

3. **Compute grid forces** using equation (7.6) with $\hat{\boldsymbol{x}}_i = \boldsymbol{x}_i$.

4. **Update velocities on grid** to $\boldsymbol{v}_i^\star$ using equation (7.10).

5. **Grid-based body collisions** on $\boldsymbol{v}_i^\star$ as described in Section 7.7.

6. **Solve the linear system** in equation (7.9) for implicit integration. For explicit time integration, simply let $\boldsymbol{v}_{\boldsymbol{i}}^{n+1} = \boldsymbol{v}_{\boldsymbol{i}}^{\star}$.

7. **Update deformation gradient.** The deformation gradient for each particle is updated as $\boldsymbol{F}_p^{n+1} = (\boldsymbol{I} + \Delta t \nabla \boldsymbol{v}_p^{n+1})\boldsymbol{F}_p^n$, where we have computed $\nabla \boldsymbol{v}_p^{n+1} = \sum_{\boldsymbol{i}} \boldsymbol{v}_{\boldsymbol{i}}^{n+1} (\nabla w_{\boldsymbol{ip}}^n)^T$. Section 7.6 gives a detailed description of the update rule for elastic and plastic parts of $\boldsymbol{F}$.

8. **Update particle velocities.** Our new particle velocities are $\boldsymbol{v}_p^{n+1} = (1-\alpha)(\boldsymbol{v}_P)_p^{n+1} + \alpha(\boldsymbol{v}_F)_p^{n+1}$, where the PIC part is $(\boldsymbol{v}_P)_p^{n+1} = \sum_{\boldsymbol{i}} \boldsymbol{v}_{\boldsymbol{i}}^{n+1} w_{\boldsymbol{ip}}^n$ and the FLIP part is $(\boldsymbol{v}_F)_p^{n+1} = \boldsymbol{v}_p^n + \sum_{\boldsymbol{i}} (\boldsymbol{v}_{\boldsymbol{i}}^{n+1} - \boldsymbol{v}_{\boldsymbol{i}}^n) w_{\boldsymbol{ip}}^n$. We typically used $\alpha = 0.95$.

9. **Particle-based body collisions** on $\boldsymbol{v}_p^{n+1}$ as detailed in Section 7.7.

10. **Update particle positions** using $\boldsymbol{x}_p^{n+1} = \boldsymbol{x}_p^n + \Delta t \boldsymbol{v}_p^{n+1}$.

7.4 Constitutive model

Snow material behavior is complicated by the fact that it contains a combination of water and ice which strongly affects its behavior. One might be tempted to think that snow behaves like sand. However, one major difference is that snow is typically compressible while sand is not. In addition, its behavior changes dramatically with a number of environmental factors including temperature, humidity, density and snow age, making snow constitutive modeling a challenging and open research problem. We refer the reader to [64] for a thorough discussion of the many approaches in the engineering literature and [30] as a general snow reference.

We found the methodology employed in [57] to be most relevant to computer graphics because it is concerned with the large strains typical of visually compelling scenes. In this work, the authors use a specially designed finite-strain multiplicative plasticity law employing the Drucker-Prager plasticity model [20]. They couple this with a hyperelastic dependence of the Kirchoff stress on the elastic part of the multiplicative decomposition of the deformation gradient. While this model is designed to match the stress-strain relation of snow

under a number of realistic conditions, we found that a more simplified treatment of finite-strain multiplicative plasticity coupled with a common graphics model for hyperelasticity was sufficient for visual realism. The most salient features of our approach are the use of principal stretches rather than principal stresses in defining our plastic yield criteria as well as a simplification of hardening behavior that only requires modification of the Lamé parameters in the hyperelastic energy density. While principal-stress-based plasticity is more appropriate for physical accuracy, principal-stretch-based yield gives the user more control over the visual behavior of the simulation.

In multiplicative plasticity theory it is customary to separate $\boldsymbol{F}$ into an elastic part $\boldsymbol{F}_E$ and a plastic part $\boldsymbol{F}_P$ so that $\boldsymbol{F} = \boldsymbol{F}_E\boldsymbol{F}_P$. We define our constitutive model in terms of the elasto-plastic energy density function

$$\Psi(\boldsymbol{F}_E, \boldsymbol{F}_P) = \mu(\boldsymbol{F}_P)\|\boldsymbol{F}_E - \boldsymbol{R}_E\|_F^2 + \frac{\lambda(\boldsymbol{F}_P)}{2}(J_E - 1)^2, \tag{7.1}$$

with the elastic part given by the fixed corotated energy density (6.1) and the Lamé parameters being functions of the plastic deformation gradients

$$\mu(\boldsymbol{F}_P) = \mu_0 e^{\xi(1-J_P)} \quad \text{and} \quad \lambda(\boldsymbol{F}_P) = \lambda_0 e^{\xi(1-J_P)}, \tag{7.2}$$

where $J_E = \det \boldsymbol{F}_E$, $J_P = \det \boldsymbol{F}_P$, $\boldsymbol{F}_E = \boldsymbol{R}_E\boldsymbol{S}_E$ by the polar decomposition, λ_0, μ_0 are the initial Lamé coefficients and ξ is a dimensionless plastic hardening parameter. Additionally we define the portion of deformation that is elastic and plastic using the singular values of the deformation gradient. We define a critical compression θ_c and stretch θ_s as the thresholds to start plastic deformation (or fracture). Namely, the singular values of $\boldsymbol{F}_E$ are restricted to the interval $[1 - \theta_c, 1 + \theta_s]$.

Our material is elastic in the regime of small deformations as dictated by the $\boldsymbol{F}_E$ dependence in (7.1). When the deformation exceeds a critical threshold (either stretch or compress) it starts deforming plastically as described in more detail in Section 7.6. This also affects the material properties in accordance with (7.2), making it stronger under compression (packing) and weaker under stretch (fracture), allowing us to achieve realistic snow phenomena.

Table 7.2: In our model we found these parameters to be a useful starting point for producing simulations.

Parameter	**Notation**	**Value**
Critical compression	θ_c	2.5×10^{-2}
Critical stretch	θ_s	7.5×10^{-3}
Hardening coefficient	ξ	10
Initial density (kg/m^3)	ρ_0	4.0×10^2
Initial Young's modulus (Pa)	E_0	1.4×10^5
Poisson's ratio	ν	0.2

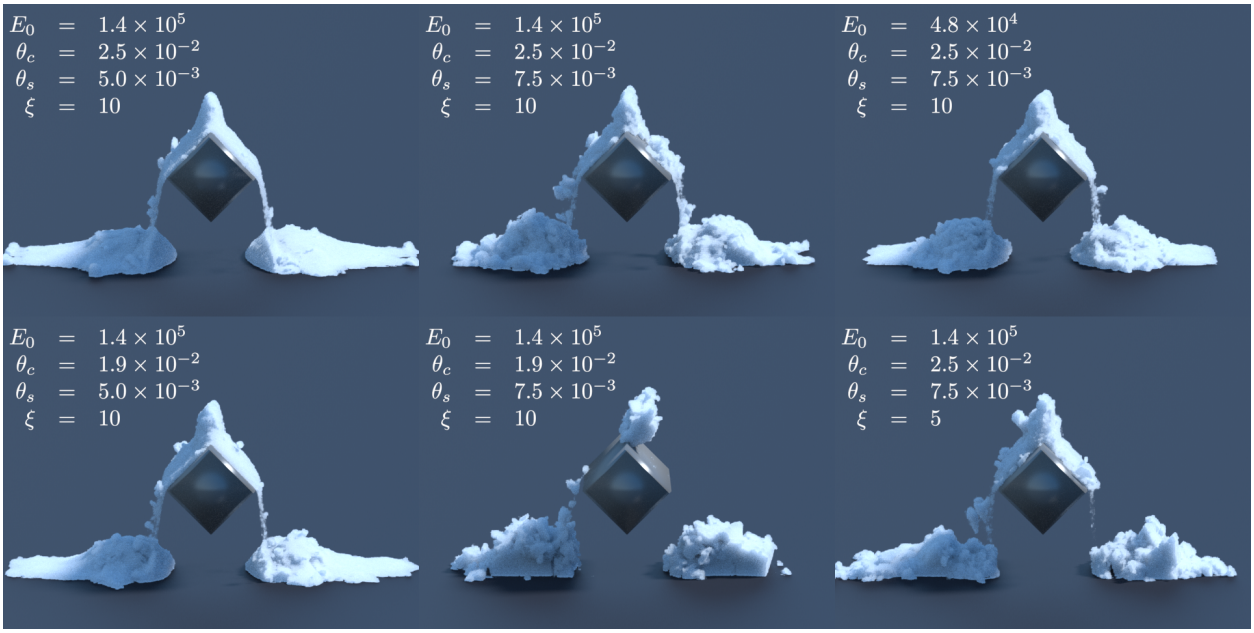

Figure 7.5: A snow block breaks over a wedge. We use the top-center image as a reference and show how changing different parameters in our model affects the look and dynamics of the material. © Disney

To simulate different types of snow, we found the following intuition useful. θ_c and θ_s determine *when* the material starts breaking (larger = chunky, smaller = powdery). The hardening coefficient determines *how fast* the material breaks once it is plastic (larger = brittle, smaller = ductile). *Dry* and *powdery* snow has smaller critical compression and stretch constants, while the opposite is true for *wet* and *chunky* snow. *Icy* snow has a higher hardening coefficient and Young's modulus, with the opposite producing *muddy* snow. See Figure 7.5 for examples of snow variation and Table 7.2 for a list of generic parameters.

7.5 Stress-based forces and linearization

The total elastic potential energy can be expressed in terms of the energy density Ψ as

$$\int_{\Omega^0} \Psi(\boldsymbol{F}_E(\boldsymbol{X}), \boldsymbol{F}_P(\boldsymbol{X}))d\boldsymbol{X}, \tag{7.3}$$

where Ω^0 is the undeformed configuration of the material. The MPM spatial discretization of the stress-based forces is equivalent to differentiation of a discrete approximation of this energy with respect to the Eulerian grid node material positions. However, we do not actually deform the Eulerian grid so we can think of the change in the grid node locations as being determined by the grid node velocities. That is, if $\boldsymbol{x_i}$ is the position of grid node $\boldsymbol{i}$, then $\hat{\boldsymbol{x}}_{\boldsymbol{i}} = \boldsymbol{x_i} + \Delta t \boldsymbol{v_i}$ would be the deformed location of that grid node given the current velocity $\boldsymbol{v_i}$ of the node. If we refer to the vector of all grid nodes $\hat{\boldsymbol{x}}_{\boldsymbol{i}}$ as $\hat{\boldsymbol{x}}$, then the MPM approximation to the total elastic potential can be written as

$$\Phi(\hat{\boldsymbol{x}}) = \sum_p V_p^0 \Psi(\hat{\boldsymbol{F}}_{Ep}(\hat{\boldsymbol{x}}), \boldsymbol{F}_{Pp}^n),$$

where V_p^0 is the volume of material originally occupied by particle p, $\boldsymbol{F}_{Pp}^n$ is the plastic part of $\boldsymbol{F}$ at particle p at time t^n and $\hat{\boldsymbol{F}}_{Ep}$ is the elastic part which is related to $\hat{\boldsymbol{x}}$ as in [84] as

$$\hat{\boldsymbol{F}}_{Ep}(\hat{\boldsymbol{x}}) = \left(\boldsymbol{I} + \sum_{\boldsymbol{i}} (\hat{\boldsymbol{x}}_{\boldsymbol{i}} - \boldsymbol{x_i})(\nabla w_{\boldsymbol{ip}}^n)^T\right) \boldsymbol{F}_{Ep}^n. \tag{7.4}$$

With this convention, the MPM spatial discretization of the stress-based forces is given as

$$-\boldsymbol{f_i}(\hat{\boldsymbol{x}}) = \frac{\partial \Phi}{\partial \hat{\boldsymbol{x}}_{\boldsymbol{i}}}(\hat{\boldsymbol{x}}) = \sum_p V_p^0 \frac{\partial \Psi}{\partial \boldsymbol{F}_E}(\hat{\boldsymbol{F}}_{Ep}(\hat{\boldsymbol{x}}), \boldsymbol{F}_{Pp}^n)(\boldsymbol{F}_{Ep}^n)^T \nabla w_{\boldsymbol{ip}}^n. \tag{7.5}$$

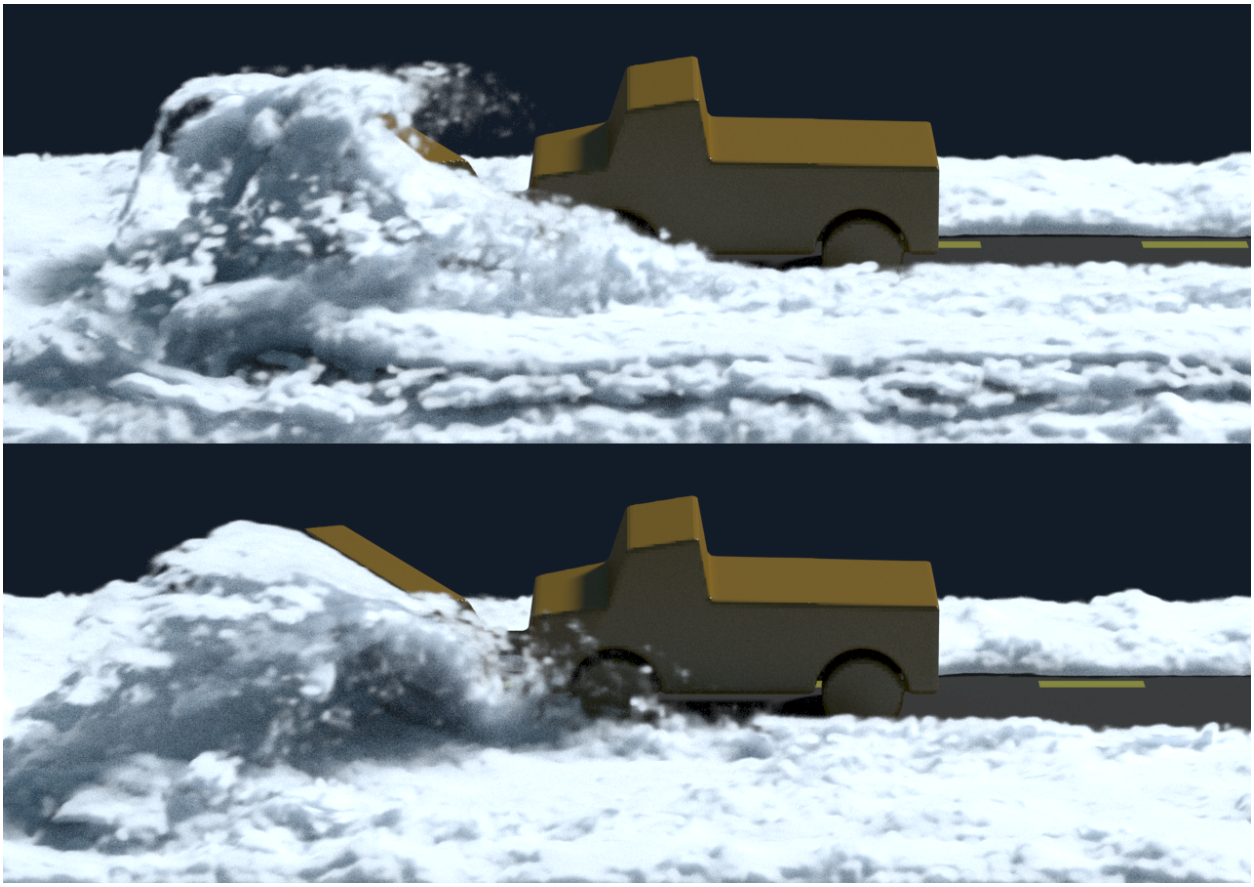

Figure 7.6: **Snowplow.** The characteristic cylindrical tube spray is created by the snowplow. © Disney

That is, $\boldsymbol{f_i}(\hat{\boldsymbol{x}})$ is the force on grid node $\boldsymbol{i}$ resulting from elastic stresses. This is often written in terms of the Cauchy stress $\boldsymbol{\sigma}_p = \frac{1}{J_p^n}\frac{\partial \Psi}{\partial \boldsymbol{F}_E}(\hat{\boldsymbol{F}}_{Ep}(\hat{\boldsymbol{x}}), \boldsymbol{F}_{Pp}^n)(\boldsymbol{F}_{Ep}^n)^T$ as

$$\boldsymbol{f_i}(\hat{\boldsymbol{x}}) = -\sum_p V_p^n \boldsymbol{\sigma}_p \nabla w_{ip}^n, \tag{7.6}$$

where $V_p^n = J_p^n V_p^0$ is the volume of the material occupied by particle p at time t^n.

We highlight this relation of the MPM spatial discretization to the elastic potential because we would like to evolve our grid velocities $\boldsymbol{v_i}$ implicitly in time. With this convention, we can take an implicit step on the elastic part of the update by utilizing the Hessian of the potential with respect to $\hat{\boldsymbol{x}}$. The action of this Hessian on an arbitrary increment $\delta \boldsymbol{u}$ can be

expressed as

$$-\delta \boldsymbol{f_i} = \sum_{\boldsymbol{j}} \frac{\partial^2 \Phi}{\partial \hat{\boldsymbol{x}}_{\boldsymbol{i}} \partial \hat{\boldsymbol{x}}_{\boldsymbol{j}}}(\hat{\boldsymbol{x}}) \delta \boldsymbol{u_j} = \sum_p V_p^0 \mathbf{A}_p (\boldsymbol{F}_{Ep}^n)^T \nabla w_{\boldsymbol{ip}}^n, \tag{7.7}$$

where

$$\mathbf{A}_p = \frac{\partial^2 \Psi}{\partial \boldsymbol{F}_E \partial \boldsymbol{F}_E}(\boldsymbol{F}_E(\hat{\boldsymbol{x}}), \boldsymbol{F}_{P_p}^n) : \left(\sum_{\boldsymbol{j}} \delta \boldsymbol{u_j} (\nabla w_{\boldsymbol{jp}}^n)^T \boldsymbol{F}_{Ep}^n \right) \tag{7.8}$$

and the notation $\boldsymbol{A} = \boldsymbol{\mathcal{C}} : \boldsymbol{D}$ is taken to mean $A_{ij} = \mathcal{C}_{ijkl} D_{kl}$ with summation implied on indices kl. See Appendix B for details of the differentiation.

7.5.1 Implicit update

We think of the elasto-plastic response as defined from the material positions of the Eulerian grid nodes $\hat{\boldsymbol{x}}_{\boldsymbol{i}} = \boldsymbol{x_i} + \Delta t \boldsymbol{v_i}$. However, as noted in the previous section, we never deform this grid. Therefore, we can think of $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}(\boldsymbol{v})$ as defined by $\boldsymbol{v}$. With this in mind, we use the following notation $\boldsymbol{f_i^n} = \boldsymbol{f_i}(\hat{\boldsymbol{x}}(\boldsymbol{0}))$, $\boldsymbol{f_i^{n+1}} = \boldsymbol{f_i}(\hat{\boldsymbol{x}}(\boldsymbol{v}^{n+1}))$ and $\frac{\partial^2 \Phi^n}{\partial \hat{\boldsymbol{x}}_{\boldsymbol{i}} \partial \hat{\boldsymbol{x}}_{\boldsymbol{j}}} = -\frac{\partial \boldsymbol{f_i^n}}{\partial \hat{\boldsymbol{x}}_{\boldsymbol{j}}} = -\frac{\partial \boldsymbol{f_i}}{\partial \hat{\boldsymbol{x}}_{\boldsymbol{j}}}(\hat{\boldsymbol{x}}(\boldsymbol{0}))$.

Using these derivatives, we form our implicit update using $\boldsymbol{v_i^{n+1}} = \boldsymbol{v_i^n} + \Delta t m_{\boldsymbol{i}}^{-1}((1-\beta)\boldsymbol{f_i^n} + \beta \boldsymbol{f_i^{n+1}}) \approx \boldsymbol{v_i^n} + \Delta t m_{\boldsymbol{i}}^{-1}(\boldsymbol{f_i^n} + \beta \Delta t \sum_{\boldsymbol{j}} \frac{\partial \boldsymbol{f_i^n}}{\partial \hat{\boldsymbol{x}}_{\boldsymbol{j}}} \boldsymbol{v_j^{n+1}})$. This leads to a (mass) symmetric system to solve for $\boldsymbol{v_i^{n+1}}$

$$\sum_{\boldsymbol{j}} \left(\boldsymbol{I} \delta_{\boldsymbol{ij}} + \beta \Delta t^2 m_{\boldsymbol{i}}^{-1} \frac{\partial^2 \Phi^n}{\partial \hat{\boldsymbol{x}}_{\boldsymbol{i}} \partial \hat{\boldsymbol{x}}_{\boldsymbol{j}}} \right) \boldsymbol{v_j^{n+1}} = \boldsymbol{v_i^\star}, \tag{7.9}$$

where the right hand side is

$$\boldsymbol{v_i^\star} = \boldsymbol{v_i^n} + \Delta t m_{\boldsymbol{i}}^{-1} \boldsymbol{f_i^n} \tag{7.10}$$

and β chooses between explicit ($\beta = 0$), trapezoidal ($\beta = \frac{1}{2}$), and backward Euler ($\beta = 1$).

7.6 Deformation gradient update

We start of by temporarily defining $\hat{\boldsymbol{F}}_{Ep}^{n+1} = (\boldsymbol{I} + \Delta t \nabla \boldsymbol{v}_p^{n+1}) \boldsymbol{F}_{Ep}^n$ as in (7.4) and $\hat{\boldsymbol{F}}_{Pp}^{n+1} = \boldsymbol{F}_{Pp}^n$, so that initially all the changes get attributed to the elastic part of the deformation gradient

$$\boldsymbol{F}_p^{n+1} = (\boldsymbol{I} + \Delta t \nabla \boldsymbol{v}_p^{n+1}) \boldsymbol{F}_{Ep}^n \boldsymbol{F}_{Pp}^n = \hat{\boldsymbol{F}}_{Ep}^{n+1} \hat{\boldsymbol{F}}_{Pp}^{n+1}. \tag{7.11}$$

The next step is to take the part of $\hat{\boldsymbol{F}}_{Ep}^{n+1}$ that exceeds the critical deformation threshold and push it into $\hat{\boldsymbol{F}}_{Pp}^{n+1}$. We compute the singular value decomposition $\hat{\boldsymbol{F}}_{Ep}^{n+1} = \boldsymbol{U}_p \bar{\boldsymbol{\Sigma}}_p \boldsymbol{V}_p^T$ and then clamp the singular values to the permitted range $\boldsymbol{\Sigma}_p = \text{clamp}\big(\bar{\boldsymbol{\Sigma}}_p, [1-\theta_c, 1+\theta_s]\big)$. The final elastic and plastic components of the deformation gradient are computed as

$$\boldsymbol{F}_{Ep}^{n+1} = \boldsymbol{U}_p \boldsymbol{\Sigma}_p \boldsymbol{V}_p^T \quad \text{and} \quad \boldsymbol{F}_{Pp}^{n+1} = \boldsymbol{V}_p \boldsymbol{\Sigma}_p^{-1} \boldsymbol{U}_p^T \boldsymbol{F}_p^{n+1}. \tag{7.12}$$

It can be easily verified that $\boldsymbol{F}_p^{n+1} = \boldsymbol{F}_{Ep}^{n+1} \boldsymbol{F}_{Pp}^{n+1}$.

7.7 Body Collisions

We process collisions against collision bodies twice each time step. The first time is on the grid velocity $\boldsymbol{v}_i^\star$ immediately after forces are applied to grid velocities. In the case of implicit integration, this contributes to the right hand side of the linear system and degrees of freedom corresponding to the colliding grid nodes are projected out during the solve. We apply collisions once more to particle velocities $\boldsymbol{v}_p^{n+1}$ just before updating positions to account for the minor discrepancies between particle and grid velocities due to interpolation. In each case, collision processing is performed the same way. All of our collisions are inelastic.

Collision objects are represented as level sets, which makes collision detection ($\phi \leq 0$) trivial. In case of a collision the local normal $\boldsymbol{n} = \nabla\phi$ and object velocity $\boldsymbol{v}_{co}$ are computed. First, the particle/grid velocity $\boldsymbol{v}$ is transformed into the reference frame of the collision object, $\boldsymbol{v}_{rel} = \boldsymbol{v} - \boldsymbol{v}_{co}$. If the bodies are separating ($v_n = \boldsymbol{v}_{rel} \cdot \boldsymbol{n} \geq 0$), then no collision is applied. Let $\boldsymbol{v}_t = \boldsymbol{v}_{rel} - \boldsymbol{n} v_n$ be the tangential portion of the relative velocity. If a sticking impulse is required ($\|\boldsymbol{v}_t\| \leq -\mu v_n$), then we simply let $\boldsymbol{v}'_{rel} = \boldsymbol{0}$, where the prime indicates that the collision has been applied. Otherwise, we apply dynamic friction, and $\boldsymbol{v}'_{rel} = \boldsymbol{v}_t + \mu v_n \boldsymbol{v}_t / \|\boldsymbol{v}_t\|$, where μ is the coefficient of friction. Finally, we transform the collided relative velocity back into world coordinates with $\boldsymbol{v}' = \boldsymbol{v}'_{rel} + \boldsymbol{v}_{co}$.

We used two types of collision objects: rigid and deforming. In the rigid case, we store a stationary level set and a potentially time-varying rigid transform, which we can use to

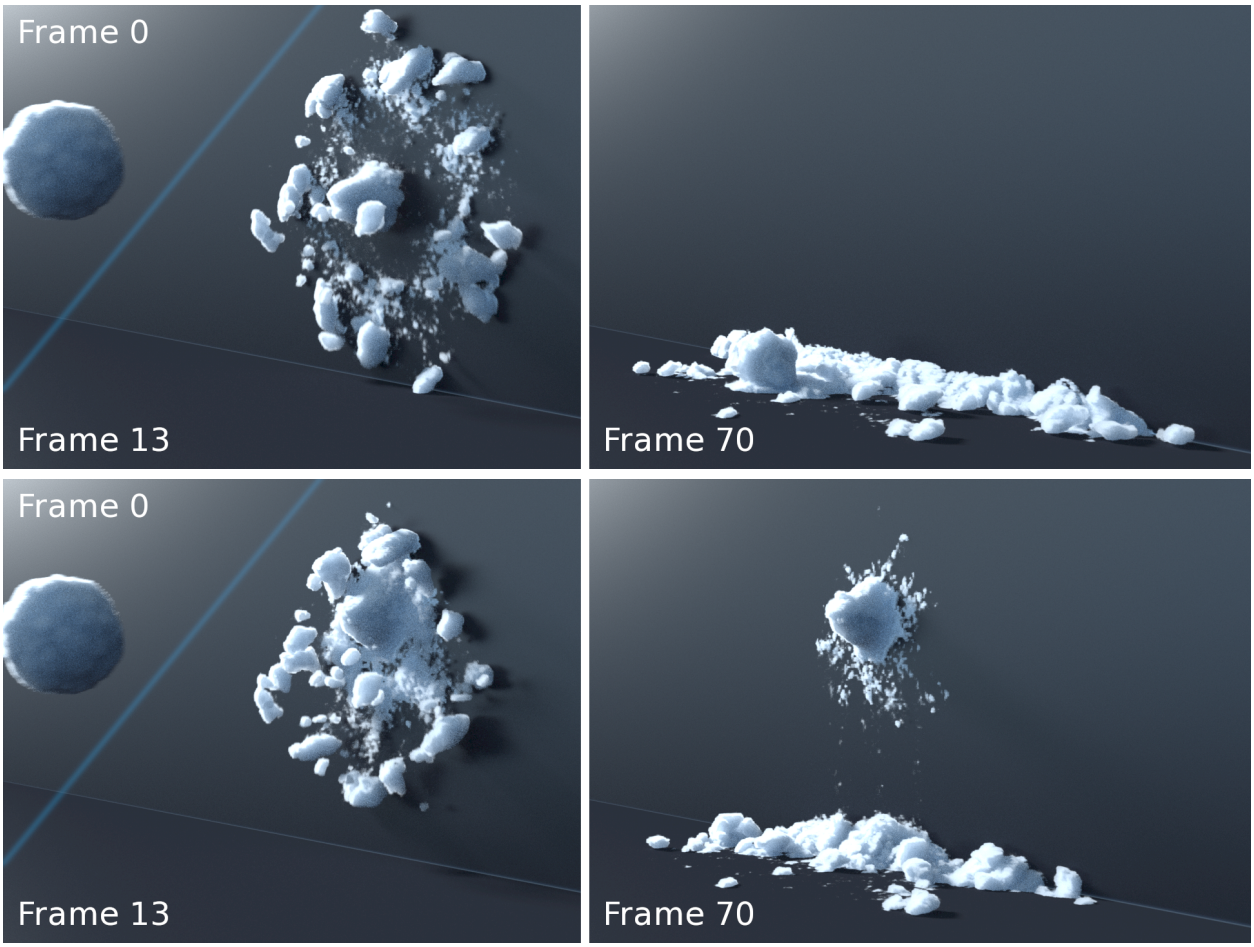

Figure 7.7: **Snowball smash.** A snowball smashes against a wall with sticky (bottom) and non-sticky (top) collisions. © Disney

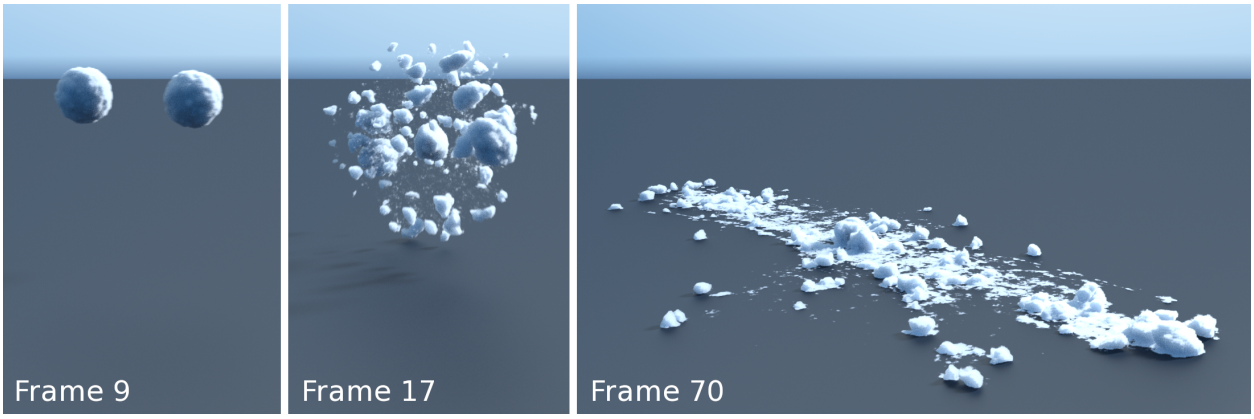

Figure 7.8: **Double smash.** Two snowballs collide and shatter. © Disney

compute ϕ, $\boldsymbol{n}$, and $\boldsymbol{v}_{co}$ at any point. In the deforming case, we load level set key frames and interpolate them similarly to [77] using $\phi(\boldsymbol{x}, t+\gamma\Delta t) = (1-\gamma)\phi(\boldsymbol{x}-\gamma\Delta t\boldsymbol{v}_{co}, t) + \gamma\phi(\boldsymbol{x}+(1-\gamma)\Delta t\boldsymbol{v}_{co}, t+\Delta t)$, except we compute the velocity as $\boldsymbol{v}_{co} = (1-\gamma)\boldsymbol{v}(\boldsymbol{x}, t) + \gamma\boldsymbol{v}(\boldsymbol{x}, t+\Delta t)$ instead of the average velocity.

Finally, we utilize a sort of *sticky* collision in situations where we want snow to stick to vertical or under-hanging surfaces. In this case, Coulomb friction is insufficient since the normal relative velocity would be zero (vertical) or positive (under-hanging and separating due to gravity). We achieve this effect by setting $\boldsymbol{v}'_{rel} = \boldsymbol{0}$ unconditionally for collisions against these surfaces.

7.8 Rendering

Researchers have measured scattering properties by applying scattering and radiative transport theory (see [94]), and this has been popular in graphics as well (see [65]). Our discrete Cartesian grid measures density relative to the material points, giving us a way of showing visual variation between loose and tightly packed snow. This gives us a rendering advantage over a surface or a purely point-based method.

At render time, we rasterize the final simulated material points to the simulation grid. In practice one could use different filter kernels for antialiasing (and even a different grid) but we use those from Section 7.3 for simplicity. We employ a volumetric path tracer to solve the volume scattering equation using a Henyey-Greenstein phase function that approximates the Mie scattering theory of ice crystals. We typically use a mean-cosine of $g = 0.5$ to obtain forward scattering, an extinction coefficient $\sigma_t = 724\text{m}^{-1}$ and scattering albedo $\sigma_s/\sigma_t = [0.9, 0.95, 1.0]$ where σ_s is the scattering coefficient.

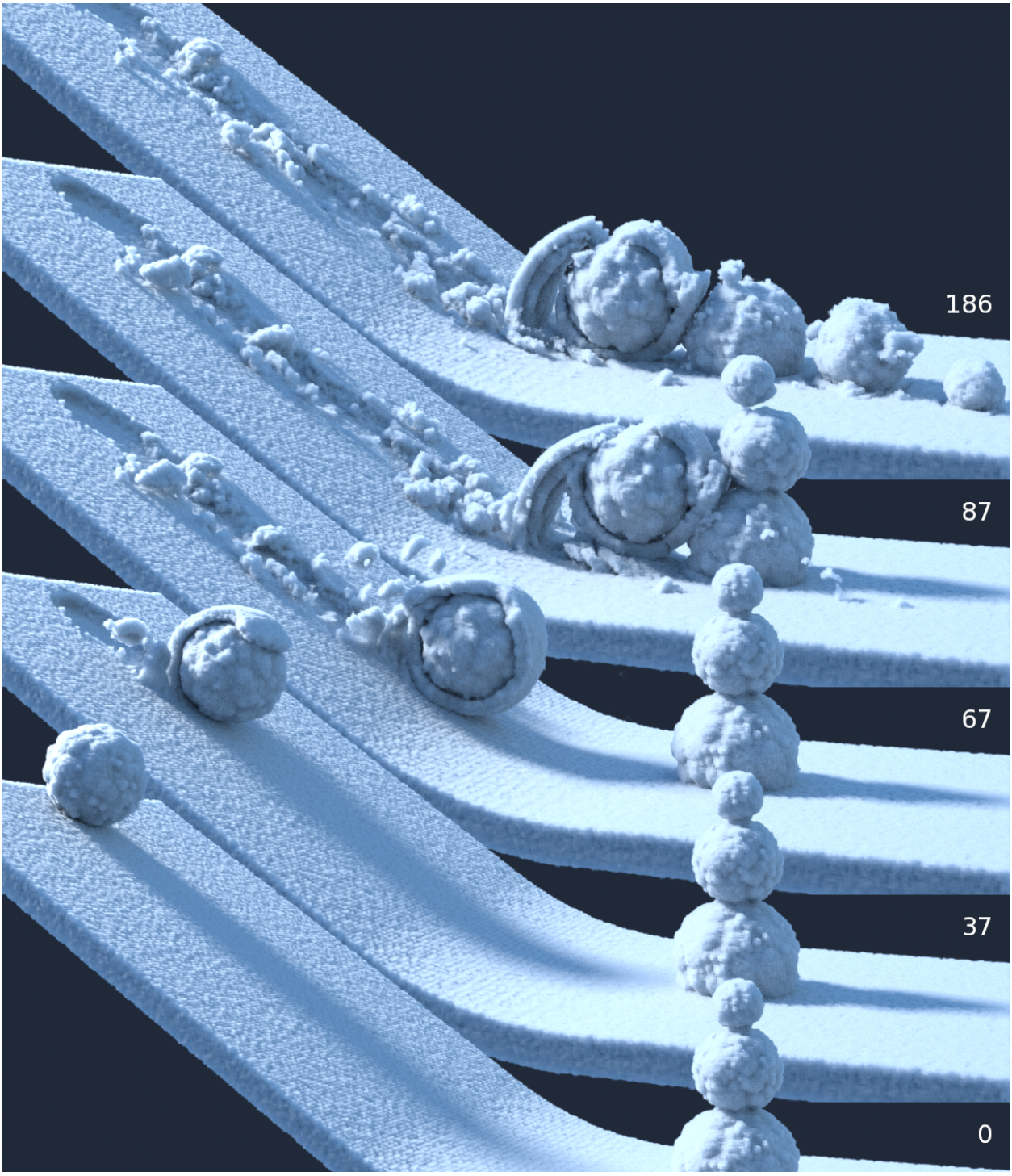

Figure 7.9: **Rolling snowball.** As the snowball moves down the hill, compressed snow sticks, demonstrating that we can handle so-called packing snow effect. © Disney

7.9 Results

We have simulated a variety of examples that demonstrate the power of our method. Our constitutive model's combination of compressibility, plasticity and hardening automatically handles fracture and *packing snow's* characteristic *sticky effect*. In particular, we show a variety of snowball simulations illustrating fracture in Figures 7.7, 7.8 and 7.1. We can also handle sculpted initial snow shapes fracturing as seen in Figures 7.2, 7.3 and 7.10. Snow's *sticky effect* is produced when originally disparate pieces of snow are compressed together which we demonstrate with a growing snowball clobbering a snowman in Figure 7.9.

We also demonstrate the characteristic tumbling motion of snow being plowed in Figure 7.6. The simulation was performed on a translating grid to save computation.

Table 7.3 lists the simulation times and resolutions for each of the examples. For all of our examples we randomly seeded particles into the volumes we needed to fill in with snow and gave them the same initial parameters. We found that using $4 - 10$ particles per grid cell (for initially packed snow) produced plausible results. In addition, we found that we did not need to perform any reseeding of particles, and we also optimized grid operations to occur only on nodes where particles' interpolation radii overlapped. Thus our computation remained proportional to the number of particles and equivalently the number of occupied grid cells.

In some cases we used spatially varying constitutive parameters defined per particle to get more realistic results. For instance, the snowballs were made harder and heavier on the outside with stiffness varied with a noise pattern to get chunky fracture. This mimics our experiments with snowballs in the real world.

Though simple, the explicit update scheme requires very restrictive time steps for stability; it would often require $\Delta t \simeq 10^{-5}$ to get plausible simulation behavior. By contrast, our semi-implicit method is less restrictive, allowing $\Delta t \simeq 0.5 \times 10^{-3}$ for all of the examples presented in this paper. The semi-implicit update step yields a (mass) symmetric system (7.9) which we solved using the conjugate residual method. In practice we found only 10-30

Table 7.3: Example particle counts, resolutions and simulation times. Simulations were performed on an 8-core Intel Xeon X5550 2.67GHz machine.

Example	**Particles**	**Grid**	**min/frame**
Snowball drop	3.0×10^5	$600 \times 300 \times 600$	5.2
Snowball smash	3.0×10^5	$200 \times 240 \times 600$	7.3
Double smash	6.0×10^5	$800 \times 300 \times 800$	13.3
Snowplow	3.9×10^6	$150 \times 50 \times 300$	2.1
Rolling snowball	7.2×10^6	$200 \times 240 \times 470$	35.7
SIGGRAPH	7.5×10^5	$780 \times 120 \times 220$	4.7
The end	5.8×10^5	$700 \times 120 \times 210$	3.8
Castle destruction	1.6×10^6	$360 \times 160 \times 560$	6.0

iterations were necessary with no preconditioning independent of grid resolution or number of particles.

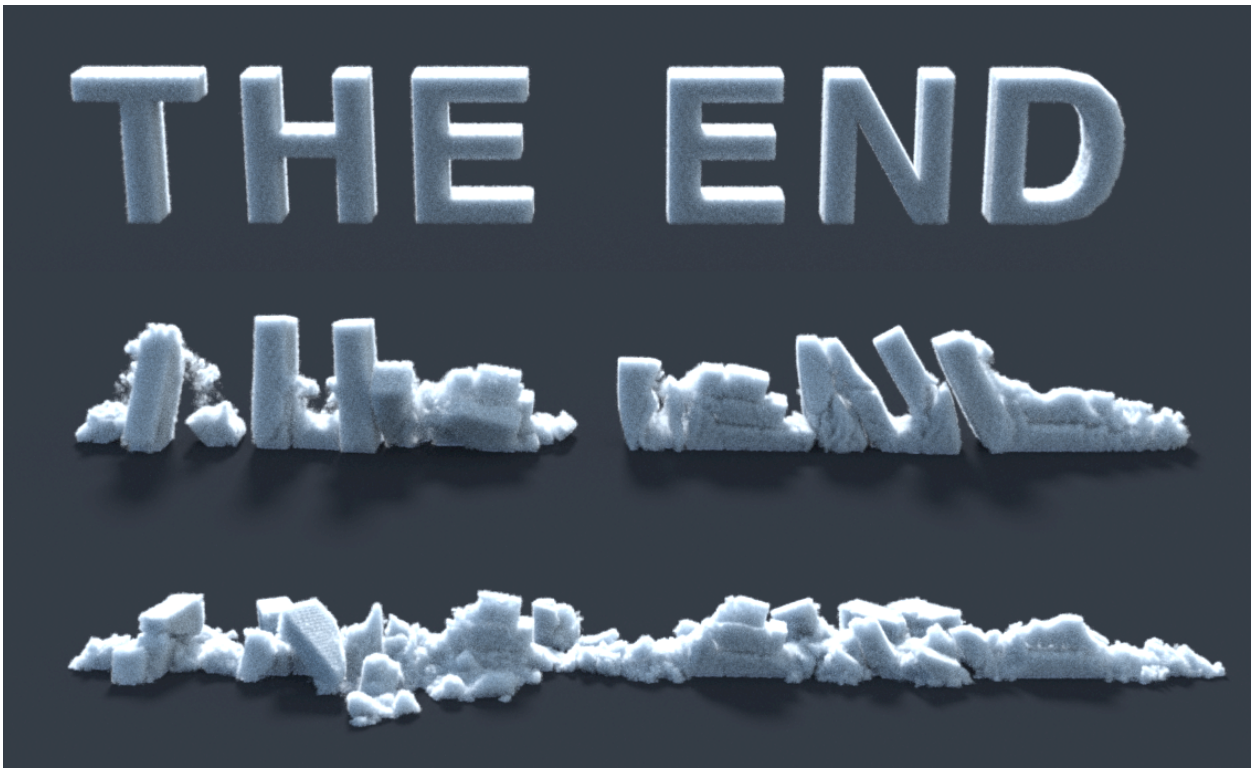

Figure 7.10: **The end.** We can simulate other words, too. © Disney

CHAPTER 8

Conclusion

We demonstrated a number of practical techniques and examples of designing elasto-plastic constitutive models for deformable solids. We targeted issues relevant primarily to graphics applications, which include robustness to extreme deformations, ability to cover a wide range of physical phenomena and providing user with intuitive controls over the material's resulting behavior.

In particular, we developed a method for extending arbitrary isotropic hyperelastic constitutive models to inverted configurations. Our energy based approach gives continuous stress, unambiguous stress derivatives and is significantly more robust to inversion and large deformations than traditional IFE [40, 86]. In addition, we introduced a novel concept of a constitutive model's primary contour. We showed how this concept can be efficiently used to predict model's stability and we also showed how to use the primary contour methodology to improve the robustness of a popular corotated model [13, 56] to large deformations.

Further, we presented a novel snow constitutive model designed for intuitive user control of practical snow behavior. This was also designed to achieve our goal of describing the many phases of snow behavior with one constitutive relation. We borrowed from the vast engineering literature on snow and demonstrated that an elasto-plastic treatment is an effective means of handling the transition between many different behaviors including flowing, clumping, breaking and more. Finally, we developed a semi-implicit Material Point Method (MPM) [84] to efficiently treat the wide range of material stiffnesses, collisions and topological changes arising in complex snow scenes. It presents and interesting new technique for continuum mechanics that will likely inspire additional research in graphics.

APPENDIX A

Solid Mechanics Reference

This chapter covers notations and main concepts from continuum mechanics, focusing primarily on the mechanics of solids. We follow closely [8, 29] in our developments, and refer the reader to those manuscripts for more details.

A.1 Kinematics

A.1.1 Notations

The deformation of a body from *initial* or *reference* configuration Ω_0 to *current* configuration Ω_t at time t is described by a function $\boldsymbol{\phi} : \Omega_0 \to \Omega_t$, which maps each point $\boldsymbol{X} \in \Omega_0$ to a point $\boldsymbol{x}(t) \in \Omega_t$ (see Figure A.1)

$$\boldsymbol{x}(t) = \boldsymbol{\phi}(\boldsymbol{X}, t) \tag{A.1}$$

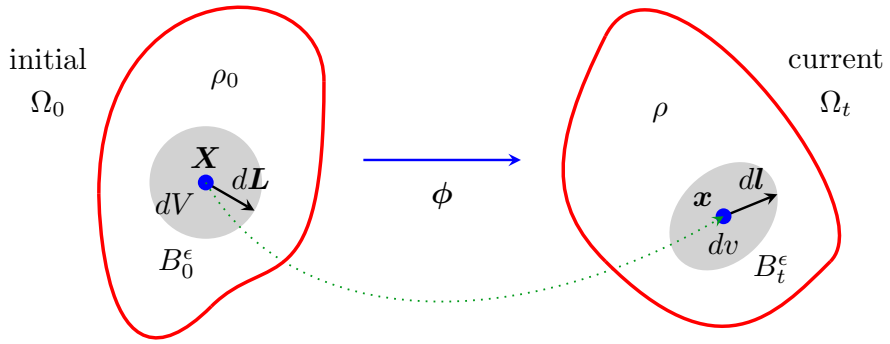

Figure A.1: Illustration of deformation. Ω_0 denotes the reference configuration and Ω_t denotes the deformed configuration at time t. The dot in Ω_0 and Ω_t represents one and the same material particle.

The velocity and acceleration of a particle $\boldsymbol{X}$ at time t, located at $\boldsymbol{x} = \boldsymbol{\phi}(\boldsymbol{X}, t)$, are given respectively by

$$\boldsymbol{V}(\boldsymbol{X}, t) = \frac{\partial \boldsymbol{\phi}(\boldsymbol{X}, t)}{\partial t}, \tag{A.2}$$

$$\boldsymbol{A}(\boldsymbol{X}, t) = \frac{\partial^2 \boldsymbol{\phi}(\boldsymbol{X}, t)}{\partial t^2}. \tag{A.3}$$

This representation is often referred to as *material space* description, since everything is written in terms of material coordinates $\boldsymbol{X}$. In addition, one can consider *world space* description of the same quantities: $\boldsymbol{v}(\boldsymbol{x}, t)$ and $\boldsymbol{a}(\boldsymbol{x}, t)$. The relationship between the two is given by

$$\boldsymbol{V}(\boldsymbol{X}, t) = \boldsymbol{v}(\boldsymbol{\phi}(\boldsymbol{X}, t), t), \tag{A.4}$$

$$\boldsymbol{A}(\boldsymbol{X}, t) = \boldsymbol{a}(\boldsymbol{\phi}(\boldsymbol{X}, t), t). \tag{A.5}$$

A.1.2 Deformation Gradient

A natural way to quantify strain is through the *deformation gradient*, which is defined by

$$\boldsymbol{F}(\boldsymbol{X}, t) = \nabla^{\boldsymbol{X}} \boldsymbol{\phi}(\boldsymbol{X}, t), \tag{A.6}$$

or in index notations

$$F_{ij} = \frac{\partial x_i}{\partial X_j}. \tag{A.7}$$

Similarly, one can introduce the deformation gradient of the inverse transform

$$F^{-1}_{ij} = \frac{\partial X_j}{\partial x_i}. \tag{A.8}$$

The field $\boldsymbol{F}$ provides information on the local behavior of a deformation $\boldsymbol{\phi}$. If we consider an infinitesimal neighborhood $B_0^\epsilon(\boldsymbol{X})$ and its image $B_t^\epsilon(\boldsymbol{x})$, then we have

$$d\boldsymbol{l} = \boldsymbol{F} d\boldsymbol{L}, \tag{A.9}$$

for any vector $d\boldsymbol{L}$ within the neighborhood and its image $d\boldsymbol{l}$ (see Figure A.1). The *Jacobian* field

$$J(\boldsymbol{X}, t) = \det \boldsymbol{F}(\boldsymbol{X}, t) \tag{A.10}$$

characterizes the local volume change

$$dv = JdV, \tag{A.11}$$

where $dV = volume(B_0^\epsilon)$, $dv = volume(B_t^\epsilon)$. We also note here that the time derivative of the deformation gradient can be written as

$$\dot{\boldsymbol{F}} = \frac{\partial}{\partial t}\nabla^{\boldsymbol{X}}\boldsymbol{\phi}(\boldsymbol{X},t) = \nabla^{\boldsymbol{X}}\frac{\partial \boldsymbol{\phi}(\boldsymbol{X},t)}{\partial t} = \nabla^{\boldsymbol{X}}\boldsymbol{v} \tag{A.12}$$

A.1.3 Surface Normals Transformation

Consider an infinitesimal surface element of area dA and its normal $\boldsymbol{N}$ in the reference configuration, and let its image in the current configuration have area da and normal $\boldsymbol{n}$. The goal is to determine the relationship between $\boldsymbol{N}dA$ and $\boldsymbol{n}da$. Let dV be the volume of a cylinder built with the surface element in the reference configuration and some infinitesimal vector $d\boldsymbol{L}$, and let dv be the volume of a cylinder built with the surface element in the current configuration and vector $d\boldsymbol{l}$, which is the image of $d\boldsymbol{L}$. Then

$$dV = d\boldsymbol{L}\cdot\boldsymbol{N}dA, \tag{A.13}$$

$$dv = d\boldsymbol{l}\cdot\boldsymbol{n}da. \tag{A.14}$$

Also from (A.9) and (A.11) we have

$$d\boldsymbol{l} = \boldsymbol{F}d\boldsymbol{L}, \tag{A.15}$$

$$dv = JdV. \tag{A.16}$$

Substituting (A.13) and (A.14) into (A.16) and using (A.15) gives

$$\boldsymbol{F}d\boldsymbol{L}\cdot\boldsymbol{n}da = Jd\boldsymbol{L}\cdot\boldsymbol{N}dA \quad \Longleftrightarrow \quad d\boldsymbol{L}\cdot\boldsymbol{F}^T\boldsymbol{n}da = d\boldsymbol{L}\cdot J\boldsymbol{N}dA \tag{A.17}$$

Since $d\boldsymbol{L}$ was chosen arbitrarily, we get

$$\boldsymbol{n}da = J\boldsymbol{F}^{-T}\boldsymbol{N}dA, \tag{A.18}$$

or in index notation

$$n_i da = JF_{ji}^{-1}N_j dA. \tag{A.19}$$

A.2 Dynamics

A.2.1 Conservation of Mass

Let $\rho_0(\boldsymbol{X})$ and $\rho(\boldsymbol{x}, t)$ be the densities in the reference and current configurations respectively (see Figure A.1). Note that at $t = 0$ we have $\boldsymbol{x} = \boldsymbol{X}$ and hence $\rho \equiv \rho_0$.

A.2.1.1 Lagrangian Formulation

For any $B_0 \subset \Omega_0$ and its image $B_t = \boldsymbol{\phi}(B_0) \subset \Omega_t$ (not necessarily small) we have

$$\int_{B_0} \rho_0(\boldsymbol{X}) d\boldsymbol{X} = \int_{B_t} \rho(\boldsymbol{x}, t) d\boldsymbol{x}, \tag{A.20}$$

which is dictated by the conservation of mass. Applying the change of variables to the second integral gives

$$\int_{B_t} \rho(\boldsymbol{x}, t) d\boldsymbol{x} = \int_{B_0} R(\boldsymbol{X}, t) J(\boldsymbol{X}, t) d\boldsymbol{X}, \tag{A.21}$$

where $R(\boldsymbol{X}, t)$ is the *material* description of $\rho(\boldsymbol{x}, t)$

$$R(\boldsymbol{X}, t) = \rho(\boldsymbol{\phi}(\boldsymbol{X}, t), t). \tag{A.22}$$

Since (A.20) and (A.21) hold for any $B_0 \subset \Omega_0$ we have

$$R(\boldsymbol{X}, 0) = JR(\boldsymbol{X}, t). \tag{A.23}$$

A.2.1.2 Eulerian Formulation

Equality (A.20) can be equivalently written as

$$\frac{\partial}{\partial t} \int_{B_t} \rho(\boldsymbol{x}, t) d\boldsymbol{x} = 0. \tag{A.24}$$

Differentiating the integral and applying divergence theorem gives

$$\frac{\partial}{\partial t} \int_{B_t} \rho(\boldsymbol{x}, t) d\boldsymbol{x} = \int_{B_t} \frac{\partial \rho}{\partial t} d\boldsymbol{x} + \int_{\partial B_t} \rho \boldsymbol{v} \cdot \boldsymbol{n} da = \int_{B_t} \Big(\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) \Big) d\boldsymbol{x}, \tag{A.25}$$

Since $B_t \subset \Omega_t$ is arbitrary, combining (A.24) and (A.25) gives

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) = 0. \tag{A.26}$$

Note: It can also be shown that (A.23) and (A.26) are equivalent by directly performing the change of variables (A.1) and using the chain rule.

A.2.2 Cauchy Stress Tensor

Let point $\boldsymbol{x}$ and a unit vector $\boldsymbol{n}$ be given. Consider an element of area Δa with normal $\boldsymbol{n}$ and containing $\boldsymbol{x}$. Let the resultant force on this area from the part of the material that has $\boldsymbol{n}$ as the outer normal be $\Delta \boldsymbol{q}$. Then the *traction* vector field is defined as

$$\boldsymbol{t}(\boldsymbol{x}, \boldsymbol{n}) = \lim_{\Delta a \to 0} \frac{\Delta \boldsymbol{q}}{\Delta a}. \tag{A.27}$$

By the 3rd Newton's Law we must have

$$\boldsymbol{t}(\boldsymbol{x}, -\boldsymbol{n}) = -\boldsymbol{t}(\boldsymbol{x}, \boldsymbol{n}). \tag{A.28}$$

It can be shown, given the existence of volumetric force density, that $\boldsymbol{t}$ is linear in $\boldsymbol{n}$, so there exists a second order tensor $\boldsymbol{\sigma}(\boldsymbol{x})$ such that for any $\boldsymbol{n}$

$$\boldsymbol{t}(\boldsymbol{x}, \boldsymbol{n}) = \boldsymbol{\sigma}(\boldsymbol{x})\boldsymbol{n}. \tag{A.29}$$

It also can be shown, from the conservation of the angular momentum, that $\boldsymbol{\sigma}(\boldsymbol{x})$ is a symmetric tensor.

A.2.3 Balance of Linear Momentum

Let $B_0 \subset \Omega_0$ and $B_t = \boldsymbol{\phi}(B_0) \subset \Omega_t$. Then the integral form of balance of linear momentum can be written as

$$\frac{d}{dt} \int_{B_t} \rho(\boldsymbol{x}, t) \boldsymbol{v}(\boldsymbol{x}, t) d\boldsymbol{x} = \int_{\partial B_t} \boldsymbol{\sigma}(\boldsymbol{x}, t) \boldsymbol{n} da + \int_{B_t} \rho(\boldsymbol{x}, t) \boldsymbol{b}(\boldsymbol{x}, t) d\boldsymbol{x}, \tag{A.30}$$

where $\boldsymbol{b}(\boldsymbol{x}, t)$ is external force per unit mass.

A.2.3.1 Eulerian Formulation

Lemma A.2.1. *Let $\Phi(\boldsymbol{x},t)$ be any spatial scalar, vector or tensor field. Then*

$$\frac{d}{dt}\int_{B_t}\Phi(\boldsymbol{x},t)\rho(\boldsymbol{x},t)d\boldsymbol{x}=\int_{B_t}\dot{\Phi}(\boldsymbol{x},t)\rho(\boldsymbol{x},t)d\boldsymbol{x} \tag{A.31}$$

Proof. Switching to material coordinates and using conservation of mass equations (A.22) and (A.23) we get

$$\begin{aligned}\frac{d}{dt}\int_{B_t}\Phi(\boldsymbol{x},t)\rho(\boldsymbol{x},t)d\boldsymbol{x} &= \frac{d}{dt}\int_{B_0}\Phi(\boldsymbol{\phi}(\boldsymbol{X},t),t)\rho(\boldsymbol{\phi}(\boldsymbol{X},t),t)Jd\boldsymbol{X} = \\ &= \frac{d}{dt}\int_{B_0}\Phi(\boldsymbol{\phi}(\boldsymbol{X},t),t)R(\boldsymbol{X},0)d\boldsymbol{X} = \\ &= \int_{B_0}\dot{\Phi}(\boldsymbol{\phi}(\boldsymbol{X},t),t)R(\boldsymbol{X},0)d\boldsymbol{X} = \\ &= \int_{B_t}\dot{\Phi}(\boldsymbol{x},t)\rho(\boldsymbol{x},t)d\boldsymbol{x}.\end{aligned}$$

□

Applying Lemma A.2.1 and divergence theorem to (A.30) gives

$$\int_{B_t}\rho(\boldsymbol{x},t)\boldsymbol{a}(\boldsymbol{x},t)d\boldsymbol{x}=\int_{B_t}\nabla^{\boldsymbol{x}}\cdot\boldsymbol{\sigma}(\boldsymbol{x},t)d\boldsymbol{x}+\int_{B_t}\rho(\boldsymbol{x},t)\boldsymbol{b}(\boldsymbol{x},t)d\boldsymbol{x}.$$

And since $B_t\subset\Omega_t$ is arbitrary we have

$$\rho\boldsymbol{a}=\nabla^{\boldsymbol{x}}\cdot\boldsymbol{\sigma}+\rho\boldsymbol{b}. \tag{A.32}$$

A.2.3.2 Lagrangian Formulation

Performing change of variables on the left-hand-side of (A.30) gives

$$\frac{d}{dt}\int_{B_t}\rho(\boldsymbol{x},t)\boldsymbol{v}(\boldsymbol{x},t)d\boldsymbol{x}=\frac{d}{dt}\int_{B_0}\rho_0(\boldsymbol{X})\boldsymbol{V}(\boldsymbol{X},t)d\boldsymbol{X}=\int_{B_0}\rho_0(\boldsymbol{X})\boldsymbol{A}(\boldsymbol{X},t)d\boldsymbol{X}.$$

Similarly for the external force on the right-hand-side

$$\int_{B_t}\rho(\boldsymbol{x},t)\boldsymbol{b}(\boldsymbol{x},t)d\boldsymbol{x}=\int_{B_0}\rho_0(\boldsymbol{X})\boldsymbol{B}(\boldsymbol{X},t)d\boldsymbol{X}.$$

where we introduced $\boldsymbol{B}(\boldsymbol{X},t) = \boldsymbol{b}(\boldsymbol{\phi}(\boldsymbol{X},t),t)$. For the surface integral we use (A.18) and divergence theorem to obtain

$$\int_{\partial B_t} \boldsymbol{\sigma}\boldsymbol{n}da = \int_{\partial B_0} \boldsymbol{\sigma}J\boldsymbol{F}^{-T}\boldsymbol{N}dA = \int_{B_0} \nabla^{\boldsymbol{X}}\boldsymbol{P}d\boldsymbol{X},$$

where we introduced the 1st Piola Kirchoff stress tensor

$$\boldsymbol{P} = \boldsymbol{\sigma}J\boldsymbol{F}^{-T}, \tag{A.33}$$

or in index notations

$$P_{ij} = \sigma_{ik}JF^{-1}_{jk}. \tag{A.34}$$

Combining all of the above results and recalling that B_0 is arbitrary we get

$$\rho_0\boldsymbol{A} = \nabla^{\boldsymbol{X}} \cdot \boldsymbol{P} + \rho\boldsymbol{B}. \tag{A.35}$$

or in index notations

$$\rho_0 A_i = P_{ij,j} + \rho B_i. \tag{A.36}$$

A.2.4 Eulerian vs Lagrangian Representations

The graph on Figure A.2 shows the relationship between different quantities in the initial and current configurations. The only quantity that has not been introduced yet is 2nd Piola Kirchoff stress tensor

$$\boldsymbol{S} = J\boldsymbol{F}^{-1}\sigma\boldsymbol{F}^{-T}. \tag{A.37}$$

Given an element of area $\boldsymbol{N}dA$ in the reference configuration it gives the force

$$d\boldsymbol{Q} = \boldsymbol{S}\boldsymbol{N}dA.$$

This is the material force in the sense that if multiplied by the deformation gradient we get the actual force on the image $\boldsymbol{n}da$ of $\boldsymbol{N}dA$

$$d\boldsymbol{q} = \boldsymbol{F}d\boldsymbol{Q}.$$

The same quantity can be computed using 1st Piola Kirchoff or Cauchy stress tensors as

$$d\boldsymbol{q} = \boldsymbol{P}\boldsymbol{N}dA = \boldsymbol{\sigma}\boldsymbol{n}da.$$

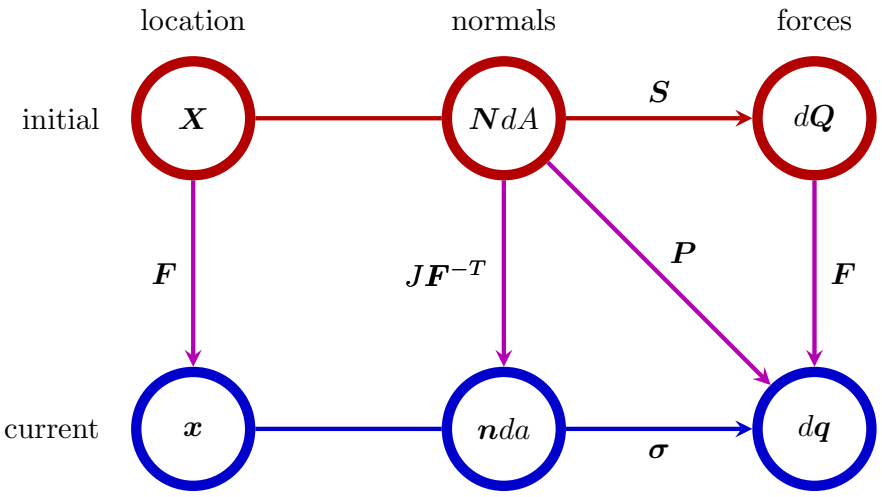

Figure A.2: The graph shows the relationship between different quantities in the initial and current configurations.

A.3 Virtual Work Principle

It is customary, especially in the finite element applications, to consider a weak formulation of the differential equation in question. In the context of solid mechanics this leads to the *virtual work equation* which is a weak form of the balance of linear momentum. Let $\delta\boldsymbol{\nu}$ be an arbitrary velocity field associated with the current configuration of the body. Then (A.32) implies

$$\rho\boldsymbol{a}\cdot\delta\boldsymbol{\nu} = (\nabla^{\boldsymbol{x}}\cdot\boldsymbol{\sigma} + \rho\boldsymbol{b})\cdot\delta\boldsymbol{\nu}, \tag{A.38}$$

which is the *local virtual work equation*. Integrating over Ω_t and applying the divergence theorem yields

$$\int_{\Omega_t}\rho\boldsymbol{a}\cdot\delta\boldsymbol{\nu}d\boldsymbol{x} = \int_{\partial\Omega_t}\boldsymbol{t}\cdot\delta\boldsymbol{\nu}da - \int_{\Omega_t}\boldsymbol{\sigma}:\nabla^{\boldsymbol{x}}\delta\boldsymbol{\nu}d\boldsymbol{x} + \int_{\Omega_t}\rho\boldsymbol{b}\cdot\delta\boldsymbol{\nu}d\boldsymbol{x}, \tag{A.39}$$

which is the *global virtual work equation*. In case of $\delta\boldsymbol{\nu} = \boldsymbol{v}$ the left hand side is the rate of change of the total kinetic energy of the system (recall Lemma A.2.1), while the right hand side is the total power of the forces acting on the system: both internal and external. Thus, (A.39) can be viewed as a generalization of the energy conservation law. We will now focus

on the internal elastic term

$$\delta W_E = \int_{\Omega_t} \boldsymbol{\sigma} : \nabla^{\boldsymbol{x}} \delta \boldsymbol{\nu} d\boldsymbol{x}, \tag{A.40}$$

and derive a material space description for it. We first note that the deformation gradient change rate due to the virtual displacement $\delta\boldsymbol{\nu}$ is given by

$$\delta \dot{\boldsymbol{F}} = \nabla^{\boldsymbol{X}} \delta \boldsymbol{\nu} = \nabla^{\boldsymbol{x}} \delta \boldsymbol{\nu} \boldsymbol{F}$$

where we used relationship (A.12). Hence

$$\begin{aligned} \delta W_E &= \int_{\Omega_t} \boldsymbol{\sigma} : \nabla^{\boldsymbol{x}} \delta \boldsymbol{\nu} d\boldsymbol{x} = \int_{\Omega_0} J\boldsymbol{\sigma} : (\dot{\boldsymbol{F}}\boldsymbol{F}^{-1}) d\boldsymbol{x} = \int_{\Omega_0} \mathrm{tr}(J\boldsymbol{\sigma}\dot{\boldsymbol{F}}\boldsymbol{F}^{-1}) d\boldsymbol{x} \\ &= \int_{\Omega_0} \mathrm{tr}(J\boldsymbol{F}^{-1}\boldsymbol{\sigma}\dot{\boldsymbol{F}}) d\boldsymbol{x} = \int_{\Omega_0} (J\boldsymbol{\sigma}\boldsymbol{F}^{-T}) : \dot{\boldsymbol{F}} d\boldsymbol{x} = \int_{\Omega_0} \boldsymbol{P} : \dot{\boldsymbol{F}} d\boldsymbol{x} \end{aligned}$$

A.4 Constitutive Modeling and Hyperelasticity

The dynamics equations derived in the previous sections were written using only stresses inside the body. Those stresses result from material deformation and hence it is necessary to specify a relationship between the two. This relationship, also known as a *constitutive model*, is typically determined by the physical properties of the material. Materials for which the constitutive behavior is only a function of the current state are known as *elastic*, and can be fully characterized by specifying the dependency $\boldsymbol{P} = \boldsymbol{P}(\boldsymbol{F}, \boldsymbol{X})$. In the special case when the work done by the stresses is dependent only on the initial and current configurations of body, the material is called *hyperelastic* and one can define its elastic energy density as

$$\Psi(\boldsymbol{F}(\boldsymbol{X}), \boldsymbol{X}) = \int_0^t \boldsymbol{P}(\boldsymbol{F}(\boldsymbol{X}), \boldsymbol{X}) : \dot{\boldsymbol{F}} dt \tag{A.41}$$

This elastic energy density $\Psi(\boldsymbol{F}, \boldsymbol{X})$ is often used to define a hyperelastic material. We note that (A.41) implies

$$\boldsymbol{P}(\boldsymbol{F}, \boldsymbol{X}) = \frac{\partial \Psi(\boldsymbol{F}, \boldsymbol{X})}{\partial \boldsymbol{F}} \tag{A.42}$$

APPENDIX B

Energy and Stress Derivatives

B.1 Isotropic Stress Derivatives in Terms of Singular Values

When specifying a constitutive model energy density Ψ in terms of singular values σ_i, it is convenient to compute the first Piola-Kirchoff stress tensor $\boldsymbol{P} = \frac{\partial \Psi}{\partial \boldsymbol{F}}$ and its derivatives $\mathbf{M} = \frac{\partial \boldsymbol{P}}{\partial \boldsymbol{F}}$ directly in terms of $\frac{\partial \Psi}{\partial \sigma_i}$ and $\frac{\partial^2 \Psi}{\partial \sigma_i \partial \sigma_j}$. To do this we parametrized $\boldsymbol{F}$ in terms of the singular values using the singular value decomposition $\boldsymbol{F} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$. Let $K = \{\sigma_1, \sigma_2, \sigma_3, u_1, u_2, u_3, v_1, v_2, v_3\}$ be the degrees of freedom parametrization $\boldsymbol{U}$, $\boldsymbol{\Sigma}$, and $\boldsymbol{V}$. We parametrize the rotations using Rodrigues' rotation formula, though any parametrization that is well-behaved around the identity would suffice. Let $C_{i\alpha} = \frac{\partial F_i}{\partial K_\alpha}$, where Latin characters (i, j) are used to represent the degrees of freedom of $\boldsymbol{F}$ (flattened into a 9-vector) and Greek letters (α, β, γ) are used to represent the degrees of freedom parametrizing the singular value decomposition. Then, $C_{i\alpha}$ is the Jacobian matrix (in terms of K) for the change of variables. Let $D_{\alpha j}$ be the inverse of the Jacobian (also in terms of K), so that $C_{i\alpha}D_{\alpha j} = \delta_{ij}$. Let Ψ be the energy in terms of the degrees of freedom of $\boldsymbol{F}$ and $\hat{\Psi}$ be the energy in terms of the degrees of freedom K. Using commas to indicate partial differentiation,

$$\begin{aligned}
\Psi_{,i}C_{i,\alpha} &= \hat{\Psi}_{,\alpha} \\
\left(\Psi_{,i}C_{i,\alpha}\right)_{,\beta} &= \hat{\Psi}_{,\alpha\beta} \\
\Psi_{,ij}C_{i,\alpha}C_{j,\beta} + \Psi_{,i}C_{i,\alpha\beta} &= \hat{\Psi}_{,\alpha\beta} \\
C_{i,\alpha}D_{\alpha j} &= \delta_{ij}
\end{aligned}$$

$$
\begin{aligned}
\Psi_{,i} &= \hat{\Psi}_{,\alpha} D_{\alpha i} \\
\Psi_{,ij} C_{i,\alpha} D_{\alpha k} C_{j,\beta} &= \hat{\Psi}_{,\alpha\beta} D_{\alpha k} - \Psi_{,i} C_{i,\alpha\beta} D_{\alpha k} \\
\Psi_{,kj} C_{j,\beta} &= \hat{\Psi}_{,\alpha\beta} D_{\alpha k} - \Psi_{,i} C_{i,\alpha\beta} D_{\alpha k} \\
\Psi_{,ij} &= \hat{\Psi}_{,\alpha\beta} D_{\alpha i} D_{\beta j} - \hat{\Psi}_{,\gamma} D_{\gamma k} C_{k,\alpha\beta} D_{\alpha i} D_{\beta j}
\end{aligned}
$$

The Piola-Kirchoff stress tensor $\Psi_{,i}$ in diagonal space can be computed as $\Psi_{,i}(\boldsymbol{F}(K))\big|_{\boldsymbol{U}=\boldsymbol{V}=\boldsymbol{I}}$. It is a diagonal matrix whose diagonals are $\frac{\partial \hat{\Psi}}{\partial \sigma_i}$ and corresponds to $\hat{\boldsymbol{P}}(\hat{\boldsymbol{\Sigma}})$ from Section 6.1. The stress derivatives in diagonal space are similarly given by $\Psi_{,ij}(\boldsymbol{F}(K))\big|_{\boldsymbol{U}=\boldsymbol{V}=\boldsymbol{I}}$. This corresponds to the $\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{F}}(\boldsymbol{\Sigma})$ from Section 6.1. When this computation is performed, one finds that the 9×9 matrix can be permuted into a block diagonal matrix with diagonal blocks $\mathbf{A}^{3\times3}$, $\mathbf{B}_{12}^{2\times2}$, $\mathbf{B}_{13}^{2\times2}$, $\mathbf{B}_{23}^{2\times2}$.

$$
\mathbf{A} = \begin{bmatrix} M_{1111} & M_{1122} & M_{1133} \\ M_{2211} & M_{2222} & M_{2233} \\ M_{3311} & M_{3322} & M_{3333} \end{bmatrix} = \begin{bmatrix} \hat{\Psi}_{,\sigma_1\sigma_1} & \hat{\Psi}_{,\sigma_1\sigma_2} & \hat{\Psi}_{,\sigma_1\sigma_3} \\ \hat{\Psi}_{,\sigma_2\sigma_1} & \hat{\Psi}_{,\sigma_2\sigma_2} & \hat{\Psi}_{,\sigma_2\sigma_3} \\ \hat{\Psi}_{,\sigma_3\sigma_1} & \hat{\Psi}_{,\sigma_3\sigma_2} & \hat{\Psi}_{,\sigma_3\sigma_3} \end{bmatrix}
$$

and

$$
\mathbf{B}_{ij} = \begin{bmatrix} M_{ijij} & M_{ijji} \\ M_{jiij} & M_{jiji} \end{bmatrix} = \frac{1}{\sigma_i^2 - \sigma_j^2} \begin{bmatrix} \sigma_i \hat{\Psi}_{,\sigma_i} - \sigma_j \hat{\Psi}_{,\sigma_j} & \sigma_j \hat{\Psi}_{,\sigma_i} - \sigma_i \hat{\Psi}_{,\sigma_j} \\ \sigma_j \hat{\Psi}_{,\sigma_i} - \sigma_i \hat{\Psi}_{,\sigma_j} & \sigma_i \hat{\Psi}_{,\sigma_i} - \sigma_j \hat{\Psi}_{,\sigma_j} \end{bmatrix}
$$

for $(ij) \in \{(12), (13), (23)\}$.

The division by $\sigma_i^2 - \sigma_j^2$ is problematic when two singular values are nearly equal or when two singular values nearly sum to zero. The latter is possible with a convention for permitting negative singular values. Expanding $\mathbf{B}_{ij}$ in terms of partial fractions yields the useful decomposition

$$
\mathbf{B}_{ij} = \frac{1}{2} \frac{\hat{\Psi}_{,\sigma_i} - \hat{\Psi}_{,\sigma_j}}{\sigma_i - \sigma_j} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + \frac{1}{2} \frac{\hat{\Psi}_{,\sigma_i} + \hat{\Psi}_{,\sigma_j}}{\sigma_i + \sigma_j} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.
$$

Note that if $\hat{\Psi}$ is invariant under permutation of the singular values, then $\hat{\Psi}_{,\sigma_i} \to \hat{\Psi}_{,\sigma_j}$ as $\sigma_i \to \sigma_j$. Thus, the first term can normally be computed robustly for an isotropic model if implemented carefully. The other fraction has deeper implications. This term can be computed robustly if $\hat{\Psi}_{,\sigma_i} + \hat{\Psi}_{,\sigma_j} \to 0$ as $\sigma_i + \sigma_j \to 0$. This property is unfavorable, as it means

the constitutive model will have difficulty recovering from many inverted configurations. This corresponds to the kink described in Section 6.2.1. Since we are specifically interested in models with robust behavior under inversion, this term will necessarily be unbounded when $\sigma_i + \sigma_j \approx 0$. The best that we can hope to do in this case is avoid numerical problems by modifying the derivatives. We do this by clamping the magnitude of the denominator to not be smaller than 10^{-6} before division. Since this change does not affect the stresses, it does not affect the constitutive behavior, and its consequences will primarily be numerical in nature. We have not observed any ill effects from this alteration.

B.2 C^1 Model

In this section, we construct the energy density and its derivatives for the C^1 extrapolation model. We begin by presenting the model in 3D. This model has four regions, depending on how many singular values are below the cutoff $\sigma_i = a$. If all of the singular values are above this threshold, then the energy is just the base model Ψ. Next, assume one singular value crosses this threshold ($\sigma_3 < a$), and let $\Delta\sigma_3 = \sigma_3 - a < 0$. Let

$$\phi = \Psi|_q \qquad g_i = \left.\frac{\partial \Psi}{\partial \sigma_i}\right|_q \qquad H_{ij} = \left.\frac{\partial^2 \Psi}{\partial \sigma_i \partial \sigma_j}\right|_q$$

$$T_{ijk} = \left.\frac{\partial^3 \Psi}{\partial \sigma_i \partial \sigma_j \partial \sigma_k}\right|_q \qquad A_{ijkl} = \left.\frac{\partial^4 \Psi}{\partial \sigma_i \partial \sigma_j \partial \sigma_k \partial \sigma_l}\right|_q .$$

Then we extrapolate the energy across the threshold, add a quadratic term, and compute the derivatives

$$\hat{\Psi} = \phi + g_3 \Delta\sigma_3 + k\Delta\sigma_3^2 \qquad \frac{\partial \hat{\Psi}}{\partial \sigma_1} = g_1 + H_{13}\Delta\sigma_3 \qquad \frac{\partial \hat{\Psi}}{\partial \sigma_3} = g_3 + 2k\Delta\sigma_3$$

$$\frac{\partial^2 \hat{\Psi}}{\partial \sigma_1^2} = H_{11} + T_{113}\Delta\sigma_3 \qquad \frac{\partial^2 \hat{\Psi}}{\partial \sigma_3^2} = 2k \qquad \frac{\partial^2 \hat{\Psi}}{\partial \sigma_1 \partial \sigma_2} = H_{12} + H_{12}\Delta\sigma_3 \qquad \frac{\partial^2 \hat{\Psi}}{\partial \sigma_1 \partial \sigma_3} = H_{13}$$

The remaining terms are obtained by exchanging the indices 1 and 2.

If we instead assume two singular values cross this threshold ($\sigma_2 < a$, $\sigma_3 < a$), and let $\Delta\sigma_2 = \sigma_2 - a < 0$ and $\Delta\sigma_3 = \sigma_3 - a < 0$. Then the extrapolated energy is

$$\hat{\Psi} = \phi + g_2\Delta\sigma_2 + g_3\Delta\sigma_3 + H_{23}\Delta\sigma_2\Delta\sigma_3 + k\Delta\sigma_2^2 + k\Delta\sigma_3^2$$

$$\frac{\partial\hat{\Psi}}{\partial\sigma_1} = g_1 + H_{12}\Delta\sigma_2 + H_{13}\Delta\sigma_3 + T_{123}\Delta\sigma_2\Delta\sigma_3 \qquad \frac{\partial\hat{\Psi}}{\partial\sigma_2} = g_2 + H_{23}\Delta\sigma_3 + 2k\Delta\sigma_2$$

$$\frac{\partial^2\hat{\Psi}}{\partial\sigma_1^2} = H_{11} + T_{112}\Delta\sigma_2 + T_{113}\Delta\sigma_3 + A_{1123}\Delta\sigma_2\Delta\sigma_3 \qquad \frac{\partial^2\hat{\Psi}}{\partial\sigma_2^2} = 2k$$

$$\frac{\partial^2\hat{\Psi}}{\partial\sigma_1\partial\sigma_2} = H_{12} + T_{123}\Delta\sigma_3 \qquad \frac{\partial^2\hat{\Psi}}{\partial\sigma_2\partial\sigma_3} = H_{23}$$

The remaining terms are obtained by exchanging the indices 2 and 3.

Finally, if all three singular values cross the threshold ($\sigma_1 < a$, $\sigma_2 < a$, $\sigma_3 < a$), let $\Delta\sigma_1 = \sigma_1 - a < 0$, $\Delta\sigma_2 = \sigma_2 - a < 0$ and $\Delta\sigma_3 = \sigma_3 - a < 0$. Then the extrapolated energy is then

$$\begin{aligned}
\hat{\Psi} &= \phi + g_1\Delta\sigma_1 + g_2\Delta\sigma_2 + g_3\Delta\sigma_3 + H_{12}\Delta\sigma_1\Delta\sigma_2 + H_{13}\Delta\sigma_1\Delta\sigma_3 + H_{23}\Delta\sigma_2\Delta\sigma_3 \\
&\quad + T_{123}\Delta\sigma_1\Delta\sigma_2\Delta\sigma_3 + k\Delta\sigma_1^2 + k\Delta\sigma_2^2 + k\Delta\sigma_3^2 \\
\frac{\partial\hat{\Psi}}{\partial\sigma_1} &= g_1 + H_{12}\Delta\sigma_2 + H_{13}\Delta\sigma_3 + T_{123}\Delta\sigma_2\Delta\sigma_3 + 2k\Delta\sigma_1 \\
\frac{\partial^2\hat{\Psi}}{\partial\sigma_1^2} &= 2k \\
\frac{\partial^2\hat{\Psi}}{\partial\sigma_1\partial\sigma_2} &= H_{12} + T_{123}\Delta\sigma_3
\end{aligned}$$

The remaining derivatives are obtained by cycling the indices.

B.2.1 Continuity

To see that the model is in fact C^1, we need to show that the energy and first derivatives match at the interfaces between regions. In the case of one singular value right at the extrapolation surface, $\Delta\sigma_3 \to 0$, and

$$\hat{\Psi} = \phi + g_3\Delta\sigma_3 + k\Delta\sigma_3^2 \to \phi \qquad \frac{\partial\hat{\Psi}}{\partial\sigma_1} = g_1 + H_{13}\Delta\sigma_3 \to g_1 \qquad \frac{\partial\hat{\Psi}}{\partial\sigma_3} = g_3 + 2k\Delta\sigma_3 \to g_3.$$

These are just the base model. At the transition from two to one singular values outside the extrapolation surface, $\Delta\sigma_2 \to 0$ and

$$\hat{\Psi} = \phi + g_2\Delta\sigma_2 + g_3\Delta\sigma_3 + H_{23}\Delta\sigma_2\Delta\sigma_3 + k\Delta\sigma_2^2 + k\Delta\sigma_3^2 \to \phi + g_3\Delta\sigma_3 + k\Delta\sigma_3^2$$

$$\frac{\partial\hat{\Psi}}{\partial\sigma_1} = g_1 + H_{12}\Delta\sigma_2 + H_{13}\Delta\sigma_3 + T_{123}\Delta\sigma_2\Delta\sigma_3 \rightarrow g_1 + H_{13}\Delta\sigma_3$$

$$\frac{\partial\hat{\Psi}}{\partial\sigma_2} = g_2 + H_{23}\Delta\sigma_3 + 2k\Delta\sigma_2 \rightarrow g_2 + H_{23}\Delta\sigma_3 \qquad \frac{\partial\hat{\Psi}}{\partial\sigma_3} = g_3 + H_{23}\Delta\sigma_2 + 2k\Delta\sigma_3 \rightarrow g_3 + 2k\Delta\sigma_3$$

These agree with the values obtained when only one singular value was extrapolated. Finally, in the transition from three to two singular values outside the extrapolation surface, $\Delta\sigma_1 \rightarrow 0$ and

$$\begin{aligned}
\hat{\Psi} &= \phi + g_1\Delta\sigma_1 + g_2\Delta\sigma_2 + g_3\Delta\sigma_3 + H_{12}\Delta\sigma_1\Delta\sigma_2 + H_{13}\Delta\sigma_1\Delta\sigma_3 + H_{23}\Delta\sigma_2\Delta\sigma_3 \\
&\quad + T_{123}\Delta\sigma_1\Delta\sigma_2\Delta\sigma_3 + k\Delta\sigma_1^2 + k\Delta\sigma_2^2 + k\Delta\sigma_3^2 \\
&\rightarrow \phi + g_2\Delta\sigma_2 + g_3\Delta\sigma_3 + H_{23}\Delta\sigma_2\Delta\sigma_3 + k\Delta\sigma_2^2 + k\Delta\sigma_3^2 \\
\frac{\partial\hat{\Psi}}{\partial\sigma_1} &= g_1 + H_{12}\Delta\sigma_2 + H_{13}\Delta\sigma_3 + T_{123}\Delta\sigma_2\Delta\sigma_3 + 2k\Delta\sigma_1 \\
&\rightarrow g_1 + H_{12}\Delta\sigma_2 + H_{13}\Delta\sigma_3 + T_{123}\Delta\sigma_2\Delta\sigma_3 \\
\frac{\partial\hat{\Psi}}{\partial\sigma_2} &= g_2 + H_{23}\Delta\sigma_3 + H_{12}\Delta\sigma_1 + T_{123}\Delta\sigma_1\Delta\sigma_3 + 2k\Delta\sigma_2 \\
&\rightarrow g_2 + H_{23}\Delta\sigma_3 + 2k\Delta\sigma_2
\end{aligned}$$

These match the expressions obtained for the case where two singular values are beyond the extrapolation surface, so C^1 continuity is established.

B.3 C^2 Model

This section provides a detailed derivation of the energy density and its derivatives for the C^2 model described in Section 6.3.2.

In this section, index notation is used for conciseness and clarity. We follow the convention that letters $(i, j, k, \ldots)$ are used for indices with the Einstein summation assumed. In a few places, this convention does not fit will. For those cases, we use Greek letters $(\alpha, \beta, \gamma, \ldots)$ for the index to indicate that summation over that index is never implied. Indices that occur after a comma are differentiated. Thus, $\Psi_{,i} = \frac{\partial\Psi}{\partial\sigma_i}$ and $u_{i,jk} = \frac{\partial^2 u_i}{\partial\sigma_j\sigma_k}$. Summation limits are not stated and should go up to the dimension (that is, 2 or 3). The derivation that follows is valid in any dimension, except where noted.

The base energy Ψ is the be extended to the extrapolated energy $\hat{\Psi}$ at the point $\hat{\boldsymbol{\Sigma}}$ by extrapolating along the line to the rest configuration $\boldsymbol{r}$ ($r_i = 1$ for all i). The direction of the line is $u_i = m(\sigma_i - r_i)$, with $m = \|\hat{\boldsymbol{\Sigma}} - \boldsymbol{r}\|^{-1}$. This line intersects the contour $J = a$ at $q_i = r_i + (\sigma_i - r_i)s$. The distance along this line from the contour to $\hat{\boldsymbol{\Sigma}}$ is then $h = (\sigma_i - q_i)u_i$. The extrapolated energy is $\hat{\Psi} = \phi + hg_ju_j + \frac{1}{2}h^2H_{lj}u_lu_j$, where $\phi = \Psi\big|_q$, $g_i = \Psi_{,i}\big|_q$, and $H_{ij} = \Psi_{,ij}\big|_q$. The scalar s is given by the polynomial equation $a = \prod_\alpha(r_\alpha + (\sigma_\alpha - r_\alpha)s)$.

The differentiation of $\hat{\Psi}$ at first may seem like an impossible task, particularly in terms of debugging. We compute the extrapolated energy in many small intermediate steps, and then we differentiate each of those steps along the way to construct the extrapolated energy derivatives. This breaks the task down into many simpler quantities, which simplifies the implementation. This has the added advantage that the derivatives of each intermediate quantity can be checked numerically, which drastically simplifies the debugging process. See section B.4 for suggestions on testing derivatives numerically.

B.3.1 Simple Quantities

First, we start with a few simple quantities. The quantity $\delta_{ij} = 1$ if $i = j$, and $\delta_{ij} = 0$ otherwise. The scalar m is the reciprocal of the distance between $\boldsymbol{r}$ and $\hat{\boldsymbol{\Sigma}}$, which is a convenient intermediate in computing u_i, the direction along with extrapolation occurs.

$$
\begin{aligned}
\sigma_{i,j} &= \delta_{ij} \\
\delta_{ij,k} &= 0 \\
r_{i,j} &= 0 \\
m &= \|\hat{\boldsymbol{\Sigma}} - \boldsymbol{r}\|^{-1} \\
m_{,i} &= -(\sigma_i - r_i)m^3 \\
m_{,ij} &= -\delta_{ij}m^3 + 3(\sigma_i - r_i)(\sigma_j - r_j)m^5
\end{aligned}
$$

$$
\begin{aligned}
u_i &= m(\sigma_i - r_i) \\
u_{i,k} &= m_{,k}(\sigma_i - r_i) + m\delta_{ik} \\
u_{i,kj} &= m_{,kj}(\sigma_i - r_i) + (m_{,k}\delta_{ij} + m_{,j}\delta_{ik})
\end{aligned}
$$

These quantities do not depend on anything else. Throughout this derivation, we will group pairs of terms in Hessians that are symmetric as we have done in the expression for $u_{i,kj}$ with $(m_{,k}\delta_{ij} + m_{,j}\delta_{ik})$. In practice, one of these terms should be computed and then transposed to obtain the other.

B.3.2 Point on Extrapolation Surface

Next, we define $\boldsymbol{q}$ as the location where extrapolation begins. It is the location on the segment connecting the rest configuration $\boldsymbol{r}$ and the current configuration $\hat{\boldsymbol{\Sigma}}$ that intersects the extrapolation surface. The energy density will be extrapolated along the segment from $\boldsymbol{q}$ to $\hat{\boldsymbol{\Sigma}}$. The length of this segment is denoted h.

$$
\begin{aligned}
q_i &= r_i + (\sigma_i - r_i)s \\
q_{i,j} &= \delta_{ij}s + (\sigma_i - r_i)s_{,j} \\
q_{i,jk} &= (\delta_{ij}s_{,k} + \delta_{ik}s_{,j}) + (\sigma_i - r_i)s_{,jk} \\
h &= (\sigma_i - q_i)u_i \\
h_{,j} &= (\delta_{ij} - q_{i,j})u_i + (\sigma_i - q_i)u_{i,j} \\
h_{,jk} &= -q_{i,jk}u_i + ((\delta_{ij} - q_{i,j})u_{i,k} + (\delta_{ik} - q_{i,k})u_{i,j}) + (\sigma_i - q_i)u_{i,jk}
\end{aligned}
$$

Note that these quantities depend on an interpolation fraction s (with $0 < s < 1$) and its derivatives, which we compute next.

B.3.3 Interpolation Fraction

The interpolating fraction is constrained to lie on the extrapolation surface given by $\prod_\alpha q_\alpha = a$ for some constant $0 < a < 1$. This leads to the equation

$$a = \prod_\alpha q_\alpha = \prod_\alpha (r_\alpha + (\sigma_\alpha - r_\alpha)s).$$

This is a cubic equation in the scalar variable s. Note that when $s = 0$

$$\prod_\alpha (r_\alpha + (\sigma_\alpha - r_\alpha)s) - a = \prod_\alpha r_\alpha - a = 1 - a > 0,$$

and at $s = 1$

$$\prod_\alpha (r_\alpha + (\sigma_\alpha - r_\alpha)s) - a = \prod_\alpha \sigma_\alpha - a = J < 0.$$

Thus, we are guaranteed that there will exist a solution s to the cubic in the interval $(0, 1)$. We compute this intersection using the bisection method since it is efficient and robust.

To compute the derivatives, we use implicit differentiation. We introduce two intermediate scalars ζ and ξ (with no particular physical interpretation) to simplify these computations.

$$\begin{aligned}
\zeta &= \sum_\alpha \frac{\sigma_\alpha - r_\alpha}{q_\alpha} \\
\zeta_{,\beta} &= \frac{1}{q_\beta} - \sum_\alpha \frac{\sigma_\alpha - r_\alpha}{q_\alpha^2} q_{\alpha,\beta} \\
\zeta_{,\beta\gamma} &= -\left(\frac{q_{\beta,\gamma}}{q_\beta^2} + \frac{q_{\gamma,\beta}}{q_\gamma^2} \right) + 2 \sum_\alpha \frac{\sigma_\alpha - r_\alpha}{q_\alpha^3} q_{\alpha,\beta} q_{\alpha,\gamma} - \sum_\alpha \frac{\sigma_\alpha - r_\alpha}{q_\alpha^2} q_{\alpha,ik} \\
\xi &= \zeta^{-1} \\
\xi_{,k} &= -\xi^2 \zeta_{,k} \\
\xi_{,ki} &= 2\xi^3 \zeta_{,k} \zeta_{,i} - \xi^2 \zeta_{,ki}
\end{aligned}$$

Now, we can proceed with the differentiation of s.

$$\begin{aligned}
0 &= \sum_\alpha \frac{\delta_{\alpha k} s + (\sigma_\alpha - r_\alpha) s_{,k}}{r_\alpha + (\sigma_\alpha - r_\alpha) s} \\
s_{,\beta} &= -s \left(\sum_\alpha \frac{\sigma_\alpha - r_\alpha}{r_\alpha + (\sigma_\alpha - r_\alpha) s} \right)^{-1} \sum_\alpha \frac{\delta_{\alpha\beta}}{r_\alpha + (\sigma_\alpha - r_\alpha) s} \\
&= -\frac{s}{q_\beta} \left(\sum_\alpha \frac{\sigma_\alpha - r_\alpha}{q_\alpha} \right)^{-1} \\
&= -\frac{s\xi}{q_\beta} \\
s_{,\beta\gamma} &= -\frac{s_{,\gamma}\xi}{q_\beta} - \frac{s\xi_{,\gamma}}{q_\beta} + \frac{s\xi q_{\beta,\gamma}}{q_\beta^2} \\
&= \frac{s\xi^2}{q_\beta q_\gamma} - \frac{s\xi_{,\gamma}}{q_\beta} + \frac{s\xi q_{\beta,\gamma}}{q_\beta^2}
\end{aligned}$$

All the quantities introduced so far can now be computed. The expression for $s_{,\beta\gamma}$ can be shown to be symmetric.

B.3.4 Base Model

The base model and its first four derivatives are required on the extrapolation surface to compute the stress derivatives. These quantities are all evaluated at the point q_i.

$$\begin{aligned}
\phi &= \Psi\big|_q \\
g_i &= \Psi_{,i}\big|_q \\
H_{ij} &= \Psi_{,ij}\big|_q \\
T_{ijk} &= \Psi_{,ijk}\big|_q \\
A_{ijkl} &= \Psi_{,ijkl}\big|_q
\end{aligned}$$

These quantities are symmetric in all of their indices. We will also use some of the derivatives of these quantities. Note that the point q_i is constrained to the extrapolation surface, so the

derivatives of these will depend on the derivatives of q_i.

$$\begin{aligned}
\phi_{,i} &= g_k q_{k,i} \\
\phi_{,ij} &= g_{k,j} q_{k,i} + g_k q_{k,ij} \\
g_{k,i} &= H_{km} q_{m,i} \\
g_{k,ij} &= T_{kmn} q_{m,i} q_{n,j} + H_{km} q_{m,ij} \\
H_{kl,i} &= T_{klm} q_{m,i} \\
H_{kl,ij} &= A_{klmn} q_{m,i} q_{n,j} + T_{klm} q_{m,ij}
\end{aligned}$$

B.3.5 Extrapolated Energy

We now have all of the quantities we need to compute $\hat{\Psi}$ and its derivatives. The scalars $g_k u_k$ and $H_{kl} u_k u_l$ are required for interpolation, and we differentiate them separately first.

$$\begin{aligned}
b &= g_k u_k \\
b_{,i} &= g_{k,i} u_k + g_k u_{k,i} \\
b_{,ij} &= g_{k,ij} u_k + (g_{k,i} u_{k,j} + g_{k,j} u_{k,i}) + g_k u_{k,ij} \\
c &= H_{kl} u_k u_l \\
c_{,i} &= H_{kl,i} u_k u_l + 2 H_{kl} u_{k,i} u_l \\
c_{,ij} &= H_{kl,ij} u_k u_l + (2 H_{kl,i} u_{k,j} u_l + 2 H_{kl,j} u_{k,i} u_l) + 2 H_{kl} u_{k,ij} u_l + 2 H_{kl} u_{k,i} u_{l,j} \\
d &= T_{klj} u_k u_l u_j
\end{aligned}$$

Finally, we compute the extrapolated energy and its derivatives.

$$\begin{aligned}
\hat{\Psi} &= \phi + hb + \frac{1}{2} h^2 c \\
\hat{\Psi}_{,i} &= \phi_{,i} + h_{,i} b + h b_{,i} + h h_{,i} c + \frac{1}{2} h^2 c_{,i} \\
\hat{\Psi}_{,ik} &= \phi_{,ik} + h_{,ik} b + (h_{,i} b_{,k} + h_{,k} b_{,i}) + h b_{,ik} + h_{,k} h_{,i} c + h h_{,ik} c + (h h_{,i} c_{,k} + h h_{,k} c_{,i}) + \frac{1}{2} h^2 c_{,ik}
\end{aligned}$$

B.3.6 Robustness

The formula for $\frac{\partial P_{ij}}{\partial F_{km}}$ requires that terms of the form

$$\frac{\hat{\Psi}_{,i} - \hat{\Psi}_{,t}}{\sigma_i - \sigma_t}$$

be computed robustly. To work out a robust way to do this, it will be very convenient to introduce some new notation. We take the index $[it]$ to indicate quantities like

$$B_{[it]} = \frac{B_i - B_t}{\sigma_i - \sigma_t} \qquad C_{k,[it]} = \frac{C_{k,i} - B_{k,t}}{\sigma_i - \sigma_t}$$

where it is assumed that $i \neq t$. We are after the quantity $\hat{\Psi}_{,[it]}$. We will also reuse notation slightly. Since $r_i = 1$, we will use it for this purpose even when the usage is unrelated to the rest configuration. With this, we can say $\sigma_{[it]} = r_i r_t$ and $r_{[it]} = 0$. Note that $hm + s = 1$ and $B_k \delta_{k[it]} = B_{[it]}$.

$$\begin{aligned}
u_{[it]} &= m(\sigma_{[it]} - r_{[it]}) = m r_i r_t \\
m_{,[it]} &= -(\sigma_{[it]} - r_{[it]}) m^3 = -m^3 r_i r_t \\
u_{j,[it]} &= m^{-1} m_{,[it]} u_j + m \delta_{j[it]} \\
&= -m^2 u_j r_i r_t + m \delta_{j[it]} \\
q_{[it]} &= r_{[it]} + (\sigma_{[it]} - r_{[it]}) s = s r_i r_t \\
s_{,[it]} &= \frac{s_{,i} - s_{,s}}{\sigma_i - \sigma_t} \\
&= -s\xi \frac{q_i^{-1} - q_t^{-1}}{\sigma_i - \sigma_t} \\
&= -s\xi q_i^{-1} q_t^{-1} \frac{q_t - q_i}{\sigma_i - \sigma_t} \\
&= s\xi q_i^{-1} q_t^{-1} q_{[it]} \\
&= s^2 \xi q_i^{-1} q_t^{-1} \\
q_{m,[it]} &= \delta_{m[it]} s + (\sigma_m - r_m) s_{,[it]} \\
&= \delta_{m[it]} s + m^{-1} u_m s_{,[it]}
\end{aligned}$$

$$
\begin{aligned}
h_{,[it]} &= u_{[it]} - q_{k,[it]}u_k + (\sigma_k - q_k)u_{k,[it]} \\
&= mr_i r_t - su_{[it]} - m^{-1}u_k s_{,[it]}u_k + hu_k(m^{-1}m_{,[it]}u_k + m\delta_{k[it]}) \\
&= mr_i r_t + (hm - s)u_{[it]} - m^{-1}s_{,[it]} + hu_k m^{-1}m_{,[it]}u_k \\
&= (hm - s + 1)mr_i r_t - m^{-1}s_{,[it]} - hm^2 r_i r_t \\
&= hm^2 r_i r_t - m^{-1}s_{,[it]} \\
\phi_{,[it]} &= g_k q_{k,[it]} \\
&= g_{[it]}s + m^{-1}bs_{,[it]} \\
g_{j,[it]} &= H_{jm}q_{m,[it]} \\
&= H_{j[it]}s + m^{-1}H_{jm}u_m s_{,[it]} \\
H_{kl,[it]} &= T_{klm}q_{m,[it]} \\
&= sT_{kl[it]} + m^{-1}T_{klm}u_m s_{,[it]} \\
b_{,[it]} &= g_{k,[it]}u_k + g_k u_{k,[it]} \\
&= H_{k[it]}su_k + m^{-1}H_{km}u_m s_{,[it]}u_k - g_k m^2 u_k r_i r_t + g_k m\delta_{k[it]} \\
&= H_{k[it]}su_k + m^{-1}cs_{,[it]} - bm^2 r_i r_t + mg_{[it]} \\
c_{,[it]} &= H_{kl,[it]}u_k u_l + 2H_{kl}u_{k,[it]}u_l \\
&= sT_{kl[it]}u_k u_l + m^{-1}ds_{,[it]} - 2H_{kl}mu_k r_i r_t u_l + 2H_{kl}m\delta_{k[it]}u_l \\
&= sT_{kl[it]}u_k u_l + m^{-1}ds_{,[it]} - 2cmr_i r_t + 2H_{k[it]}mu_k
\end{aligned}
$$

Finally, we can assemble the desired quantity $\hat{\Psi}_{,[it]}$.

$$
\begin{aligned}
h_{,[it]}b + hb_{,[it]} &= -m^{-1}s_{,[it]}b + hm^2 r_i r_t b + hH_{k[it]}su_k + hm^{-1}cs_{,[it]} - hbm^2 r_i r_t + hmg_{[it]} \\
&= m^{-1}(hc - b)s_{,[it]} + hsH_{k[it]}u_k + hmg_{[it]} \\
hh_{,[it]}c + \frac{1}{2}h^2 c_{,[it]} &= -hcm^{-1}s_{,[it]} + h^2 cm^2 r_i r_t + \frac{1}{2}h^2 sT_{kl[it]}u_k u_l \\
&\quad + \frac{1}{2}h^2 m^{-1}ds_{,[it]} - h^2 cmr_i r_t + h^2 H_{k[it]}mu_k \\
&= \frac{1}{2}m^{-1}h(hd - 2c)s_{,[it]} + \frac{1}{2}h^2 sT_{kl[it]}u_k u_l + h^2 H_{k[it]}mu_k
\end{aligned}
$$

$$
\begin{aligned}
\hat{\Psi}_{,[it]} &= \phi_{,[it]} + h_{,[it]}b + hb_{,[it]} + hh_{,[it]}c + \frac{1}{2}h^2 c_{,[it]} \\
&= \phi_{,[it]} + m^{-1}(hc - b)s_{,[it]} + hsH_{k[it]}u_k + hmg_{[it]} + \frac{1}{2}m^{-1}h(hd - 2c)s_{,[it]} \\
&\quad + \frac{1}{2}h^2 sT_{kl[it]}u_k u_l + h^2 H_{k[it]}mu_k \\
&= \phi_{,[it]} + \frac{1}{2}m^{-1}\big(h^2 d - 2b\big)s_{,[it]} + hH_{k[it]}u_k + hmg_{[it]} + \frac{1}{2}h^2 sT_{kl[it]}u_k u_l \\
&= g_{[it]}s + m^{-1}bs_{,[it]} + \frac{1}{2}m^{-1}\big(h^2 d - 2b\big)s_{,[it]} + hH_{k[it]}u_k + hmg_{[it]} + \frac{1}{2}h^2 sT_{kl[it]}u_k u_l \\
&= g_{[it]} + hH_{k[it]}u_k + \frac{1}{2}h^2 sT_{kl[it]}u_k u_l + \frac{1}{2}m^{-1}h^2 ds_{,[it]}
\end{aligned}
$$

This formula is elegant, but unfortunately $H_{k[it]}$ and $T_{kl[it]}$ cannot be computed robustly.

The solution to this problem is to compute $hH_{k[it]}u_k$ and $T_{kl[it]}u_k u_l$, since they can be computed robustly. Consider the computation of $hH_{k[12]}u_k$ (the others can be obtained by cycling indices).

$$
\begin{aligned}
H_{[12]j}u_j &= H_{[12]1}u_1 + H_{[12]2}u_2 + H_{[12]3}u_3 \\
&= \frac{H_{11}u_1 - H_{12}u_1 + H_{12}u_2 - H_{22}u_2}{\sigma_1 - \sigma_2} + H_{[12]3}u_3 \\
&= \frac{H_{11}u_1 - H_{11}u_2}{\sigma_1 - \sigma_2} + \frac{H_{11}u_2 - H_{22}u_2}{\sigma_1 - \sigma_2} - \frac{H_{12}u_1 - H_{12}u_2}{\sigma_1 - \sigma_2} + H_{[12]3}u_3 \\
&= H_{11}u_{[12]} + H_{[11,22]}u_2 - H_{12}u_{[12]} + H_{[12]3}u_3
\end{aligned}
$$

where we have introduced the new notation

$$
H_{[11,22]} = \frac{H_{11} - H_{22}}{\sigma_1 - \sigma_2}.
$$

The resulting terms can each be computed robustly. Note that expanding in this way allows us to isolate the base model $(H_{11}, H_{12}, H_{[12]3}, H_{[11,22]})$ from the details of the extrapolation $(u_2, u_3, u_{[12]})$.

Similarly, we can compute $T_{jk[12]}u_j u_k$ robustly

$$\begin{aligned}
T_{jk[12]}u_j u_k &= T_{33[12]}u_3u_3 + T_{31[12]}u_3u_1 + T_{32[12]}u_3u_2 + T_{13[12]}u_3u_1 + T_{11[12]}u_1u_1 \\
&\quad + T_{12[12]}u_1u_2 + T_{23[12]}u_3u_2 + T_{21[12]}u_2u_1 + T_{22[12]}u_2u_2 \\
&= T_{33[12]}u_3u_3 + 2T_{12[12]}u_1u_2 + 2T_{31[12]}u_3u_1 + 2T_{32[12]}u_3u_2 + T_{11[12]}u_1u_1 + T_{22[12]}u_2u_2 \\
&= T_{33[12]}u_3u_3 + 2T_{12[12]}u_1u_2 + 2u_3(T_{311}u_{[12]} + T_{3[11,22]}u_2 - T_{312}u_{[12]}) \\
&\quad + (T_{[111,222]} - T_{12[12]})u_1^2 + (T_{222} - T_{122})\frac{u_1^2 - u_2^2}{\sigma_1 - \sigma_2} \\
&= T_{33[12]}u_3u_3 + 2T_{12[12]}u_1u_2 + 2u_3(T_{311}u_{[12]} + T_{3[11,22]}u_2 - T_{312}u_{[12]}) \\
&\quad + (T_{[111,222]} - T_{12[12]})u_1^2 + m^2(T_{222} - T_{122})(\sigma_1 + \sigma_2 - 2)
\end{aligned}$$

where we have introduced the new notation

$$T_{3[11,22]} = \frac{T_{311} - T_{322}}{\sigma_1 - \sigma_2} \qquad T_{[111,222]} = \frac{T_{111} - T_{222}}{\sigma_1 - \sigma_2}.$$

As before, these quantities can be computed robustly, and the base model is isolated from the details of the extrapolation. The 2D formulas for $H_{[12]j}u_j$ and $T_{jk[12]}u_j u_k$ are obtained by discarding all terms containing the index 3.

B.3.7 Continuity

To establish C^2 continuity for this model, we need to establish that $\hat{\Psi} = \phi$, $\hat{\Psi}_{,i} = g_i$, and $\hat{\Psi}_{,ik} = H_{ik}$ at the extrapolation surface. At this surface, $s = 1$ and $h = 0$. With these,

$$\begin{aligned}
q_i &= r_i + (\sigma_i - r_i)s \\
q_i &= \sigma_i \\
q_{i,j} &= \delta_{ij}s + (\sigma_i - r_i)s_{,j} \\
&= \delta_{ij} + m^{-1}u_i s_{,j}
\end{aligned}$$

$$
\begin{aligned}
\delta_{ij} - q_{i,j} &= -m^{-1} u_i s_{,j} \\
h_{,j} &= (\delta_{ij} - q_{i,j}) u_i + (\sigma_i - q_i) u_{i,j} \\
&= -m^{-1} s_{,j} \\
h_{,jk} &= -q_{i,jk} u_i + ((\delta_{ij} - q_{i,j}) u_{i,k} + (\delta_{ik} - q_{i,k}) u_{i,j}) + (\sigma_i - q_i) u_{i,jk} \\
&= -q_{i,jk} u_i \\
g_{k,i} &= H_{km} q_{m,i} \\
&= H_{ki} + m^{-1} H_{km} u_m s_{,i} \\
g_{k,ij} &= T_{kmn} q_{m,i} q_{n,j} + H_{km} q_{m,ij} \\
&= T_{kmn} q_{m,i} q_{n,j} + H_{km} q_{m,ij} \\
b_{,i} &= g_{k,i} u_k + g_k u_{k,i} \\
&= H_{ki} u_k + m^{-1} H_{km} u_m s_{,i} u_k + m_{,i} (\sigma_k - r_k) g_k + m g_i \\
&= H_{ki} u_k + m^{-1} c s_{,i} + m_{,i} m^{-1} b + m g_i \\
&= H_{ki} u_k + m^{-1} c s_{,i} + (g_i - u_i b) m
\end{aligned}
$$

With these, C^1 is established readily

$$
\begin{aligned}
\hat{\Psi} &= \phi + hb + \frac{1}{2} h^2 c \\
&= \phi \\
\hat{\Psi}_{,i} &= \phi_{,i} + h_{,i} b + h b_{,i} + h h_{,i} c + \frac{1}{2} h^2 c_{,i} \\
&= \phi_{,i} + h_{,i} b \\
&= g_k q_{k,i} - m^{-1} s_{,i} b \\
&= g_i + g_k u_k m^{-1} s_{,i} - m^{-1} s_{,i} b \\
&= g_i
\end{aligned}
$$

Finally, C^2 can be established with a bit of work.

$$\begin{aligned}
\hat{\Psi}_{,ik} &= \phi_{,ik} + h_{,ik}b + (h_{,i}b_{,k} + h_{,k}b_{,i}) + hb_{,ik} + h_{,k}h_{,i}c + hh_{,ik}c + (hh_{,i}c_{,k} + hh_{,k}c_{,i}) + \frac{1}{2}h^2c_{,ik} \\
&= \phi_{,ik} + h_{,ik}b + (h_{,i}b_{,k} + h_{,k}b_{,i}) + h_{,k}h_{,i}c \\
&= g_{j,k}q_{j,i} + g_jq_{j,ik} - q_{j,ik}u_jb + (h_{,i}b_{,k} + h_{,k}b_{,i}) + h_{,k}h_{,i}c \\
&= g_{j,k}q_{j,i} + (g_j - u_jb)q_{j,ik} + (h_{,i}b_{,k} + h_{,k}b_{,i}) + h_{,k}h_{,i}c \\
&= g_{j,k}q_{j,i} + (g_j - u_jb)(\delta_{ji}s_{,k} + \delta_{jk}s_{,i} + m^{-1}u_js_{,ik}) + (h_{,i}b_{,k} + h_{,k}b_{,i}) + h_{,k}h_{,i}c \\
&= g_{j,k}q_{j,i} + (g_i - u_ib)s_{,k} + (g_k - u_kb)s_{,i} + (h_{,i}b_{,k} + h_{,k}b_{,i}) + h_{,k}h_{,i}c \\
&= g_{j,k}q_{j,i} + (b_{,i} - (g_i - u_ib)m)h_{,k} + (b_{,k} - (g_k - u_kb)m)h_{,i} + h_{,k}h_{,i}c \\
&= g_{j,k}q_{j,i} + (H_{ji}u_j + m^{-1}cs_{,i})h_{,k} + (H_{jk}u_j + m^{-1}cs_{,k})h_{,i} + h_{,k}h_{,i}c \\
&= g_{j,k}q_{j,i} - ch_{,i}h_{,k} + H_{ji}u_jh_{,k} + H_{jk}u_jh_{,i} \\
&= g_{i,k} - g_{j,k}u_jh_{,i} - ch_{,i}h_{,k} + H_{ji}u_jh_{,k} + H_{jk}u_jh_{,i} \\
&= H_{ik} + m^{-1}H_{im}u_ms_{,k} - H_{jk}u_jh_{,i} - m^{-1}H_{jm}u_ms_{,k}u_jh_{,i} - ch_{,i}h_{,k} + H_{ji}u_jh_{,k} + H_{jk}u_jh_{,i} \\
&= H_{ik} - m^{-1}H_{jm}u_ms_{,k}u_jh_{,i} - ch_{,i}h_{,k} \\
&= H_{ik}
\end{aligned}$$

This establishes C^2 continuity for this model.

B.4 Note on Testing Derivatives Numerically

We suggest above that the derivatives can be tested numerically. Here, we present a simple yet effective way to do this. Choose a small random perturbation $\delta\boldsymbol{x}$. Suppose we have a scalar f and its derivative ∇f evaluated at $\boldsymbol{x}$ and $\delta\boldsymbol{x}$. Then,

$$f(\boldsymbol{x} + \delta\boldsymbol{x}) - f(\boldsymbol{x}) - \frac{1}{2}(\nabla f(\boldsymbol{x} + \delta\boldsymbol{x}) + \nabla f(\boldsymbol{x})) \cdot \delta\boldsymbol{x} = O(\|\delta\boldsymbol{x}\|^3).$$

This test compares a second order accurate central difference approximation against a second order average, which makes the test much less ambiguous. When the test fails, the error will generally only be of order $O(\|\delta\boldsymbol{x}\|)$. If the quantities being tested are on the order of one,

then it is most effective to choose $\|\delta \boldsymbol{x}\|^3$ to be around floating point precision. If f were instead a vector quantity, then the error quantity computed on the left hand side would be a vector, which should be nearly zero. Second order derivatives are tested against first order derivatives.

B.5 Energy Derivatives for Elasto-Plastic Snow Model

B.5.1 Total Energy Derivatives

Given an elasto-plastic energy density function $\Psi(\boldsymbol{F}_E, \boldsymbol{F}_P)$ which evaluates to $\Psi_p = \Psi(\hat{\boldsymbol{F}}_{Ep}(\hat{\boldsymbol{x}}), \boldsymbol{F}^n_{Pp})$ at each particle p using its elastic and plastic parts of the deformation gradient $\hat{\boldsymbol{F}}_{Ep}(\hat{\boldsymbol{x}})$ and $\boldsymbol{F}^n_{Pp}$, we define the full potential energy of the system to be

$$\Phi(\hat{\boldsymbol{x}}) = \sum_p V^0_p \Psi(\hat{\boldsymbol{F}}_{Ep}(\hat{\boldsymbol{x}}), \boldsymbol{F}^n_{Pp}) = \sum_p V^0_p \Psi_p,$$

where $\hat{\boldsymbol{F}}_{Ep}(\hat{\boldsymbol{x}})$ is updated as

$$\hat{\boldsymbol{F}}_{Ep}(\hat{\boldsymbol{x}}) = \left(\boldsymbol{I} + \sum_{\boldsymbol{i}} (\hat{\boldsymbol{x}}_{\boldsymbol{i}} - \boldsymbol{x}^n_{\boldsymbol{i}})(\nabla w^n_{\boldsymbol{ip}})^T \right) \boldsymbol{F}^n_{Ep}. \tag{B.1}$$

For the purposes of working out derivatives, we use index notation for differentiation, using Greek indices $\alpha, \beta, \ldots$ for spatial indices, $\Phi_{,(\boldsymbol{j}\sigma)}$ to indicate partial derivatives on $x_{\boldsymbol{j}\sigma}$, $\Phi_{,(\alpha\beta)}$ to indicate partial derivatives on $F_{E\alpha\beta}$, and summation implied over all repeated indices. The derivatives of $\hat{\boldsymbol{F}}_{Ep}$ with respect to $\boldsymbol{x}_{\boldsymbol{i}}$ are

$$\begin{aligned}
\hat{F}_{Ep\alpha\beta} &= \left(\delta_{\alpha\gamma} + (x_{\boldsymbol{i}\alpha} - x^n_{\boldsymbol{i}\alpha}) w^n_{\boldsymbol{ip},\gamma}\right) F^n_{Ep\gamma\beta} \\
\hat{F}_{Ep\alpha\beta,(\boldsymbol{j}\sigma)} &= \delta_{\alpha\sigma} w^n_{\boldsymbol{jp},\gamma} F^n_{Ep\gamma\beta} \\
\hat{F}_{Ep\alpha\beta,(\boldsymbol{j}\sigma)(\boldsymbol{k}\tau)} &= 0
\end{aligned}$$

With these, the derivatives of Φ with respect to $\boldsymbol{x_i}$ can be worked out using the chain rule

$$\begin{aligned}
\Phi &= V_p^0 \Psi_p \\
\Phi_{,(\boldsymbol{j}\sigma)} &= \sum_p V_p^0 \Psi_{p,(\alpha\beta)} \hat{F}_{Ep\alpha\beta,(\boldsymbol{j}\sigma)} \\
&= \sum_p V_p^0 \Psi_{p,(\sigma\beta)} w^n_{\boldsymbol{j}p,\gamma} F^n_{Ep\gamma\beta} \\
\Phi_{,(\boldsymbol{j}\sigma)(\boldsymbol{k}\tau)} &= \sum_p (V_p^0 \Psi_{p,(\sigma\beta)} w^n_{\boldsymbol{j}p,\gamma} F^n_{Ep\gamma\beta})_{,(\boldsymbol{k}\tau)} \\
&= \sum_p V_p^0 \Psi_{p,(\sigma\beta)(\tau\kappa)} w^n_{\boldsymbol{j}p,\gamma} F^n_{Ep\gamma\beta} w^n_{\boldsymbol{k}p,\omega} F^n_{Ep\omega\kappa}
\end{aligned}$$

These can be interpreted without the use of indices as

$$-\boldsymbol{f_i}(\hat{\boldsymbol{x}}) = \frac{\partial \Phi}{\partial \hat{\boldsymbol{x}}_{\boldsymbol{i}}}(\hat{\boldsymbol{x}}) = \sum_p V_p^0 \frac{\partial \Psi}{\partial \boldsymbol{F}_E}(\hat{\boldsymbol{F}}_{Ep}(\hat{\boldsymbol{x}}), \boldsymbol{F}^n_{Pp})(\boldsymbol{F}^n_{Ep})^T \nabla w^n_{\boldsymbol{i}p} \tag{B.2}$$

and

$$-\delta \boldsymbol{f_i} = \sum_{\boldsymbol{j}} \frac{\partial^2 \Phi}{\partial \hat{\boldsymbol{x}}_{\boldsymbol{i}} \partial \hat{\boldsymbol{x}}_{\boldsymbol{j}}}(\hat{\boldsymbol{x}}) \delta \boldsymbol{u_j} = \sum_p V_p^0 \mathbf{A}_p (\boldsymbol{F}^n_{Ep})^T \nabla w^n_{\boldsymbol{i}p} \tag{B.3}$$

where

$$\mathbf{A}_p = \frac{\partial^2 \Psi}{\partial \boldsymbol{F}_E \partial \boldsymbol{F}_E}(\boldsymbol{F}_E(\hat{\boldsymbol{x}}), \boldsymbol{F}^n_{P_p}) : \left(\sum_{\boldsymbol{j}} \delta \boldsymbol{u_j} (\nabla w^n_{\boldsymbol{j}p})^T \boldsymbol{F}^n_{Ep} \right). \tag{B.4}$$

and the notation $\boldsymbol{A} = \boldsymbol{\mathcal{C}} : \boldsymbol{D}$ is taken to mean $A_{ij} = \mathcal{C}_{ijkl} D_{kl}$ with summation implied on indices kl.

B.6 Energy Density Derivatives

For integration, we need to compute $\frac{\partial \Psi}{\partial \boldsymbol{F}_E}$ and $\frac{\partial^2 \Psi}{\partial \boldsymbol{F}_E \partial \boldsymbol{F}_E} : \delta \mathcal{D}$. In this section, we will omit the subscripts E.

$$\Psi = \mu \|\boldsymbol{F} - \boldsymbol{R}\|_F^2 + \frac{\lambda}{2}(J-1)^2$$

$$
\begin{aligned}
\delta\Psi &= \delta\left(\mu\|\boldsymbol{F}-\boldsymbol{R}\|_F^2+\frac{\lambda}{2}(J-1)^2\right)\\
&= \mu\delta\big(\|\boldsymbol{F}-\boldsymbol{R}\|_F^2\big)+\lambda(J-1)\delta J\\
&= \mu\delta\big(\mathrm{tr}(\boldsymbol{F}^T\boldsymbol{F})\big)-2\mu\delta\big(\mathrm{tr}(\boldsymbol{R}^T\boldsymbol{F})\big)+\mu\delta\big(\mathrm{tr}(\boldsymbol{R}^T\boldsymbol{R})\big)+\lambda(J-1)\delta J\\
&= 2\mu\boldsymbol{F}:\delta\boldsymbol{F}-2\mu\delta(\mathrm{tr}(\boldsymbol{S}))+\lambda(J-1)J\boldsymbol{F}^{-T}:\delta\boldsymbol{F}\\
&= 2\mu\boldsymbol{F}:\delta\boldsymbol{F}-2\mu\mathrm{tr}(\delta\boldsymbol{S})+\lambda(J-1)J\boldsymbol{F}^{-T}:\delta\boldsymbol{F}\\
\boldsymbol{F} &= \boldsymbol{R}\boldsymbol{S}\\
\delta\boldsymbol{F} &= \delta\boldsymbol{R}\boldsymbol{S}+\boldsymbol{R}\delta\boldsymbol{S}\\
\mathrm{tr}(\delta\boldsymbol{S}) &= \mathrm{tr}(\boldsymbol{R}^T\delta\boldsymbol{F})-\mathrm{tr}(\boldsymbol{R}^T\delta\boldsymbol{R}\boldsymbol{S})\\
&= \mathrm{tr}(\boldsymbol{R}^T\delta\boldsymbol{F})-(\boldsymbol{R}^T\delta\boldsymbol{R}):\boldsymbol{S}\\
&= \mathrm{tr}(\boldsymbol{R}^T\delta\boldsymbol{F})\\
&= \boldsymbol{R}:\delta\boldsymbol{F}
\end{aligned}
$$

Note that since $\boldsymbol{R}^T\boldsymbol{R}=\boldsymbol{I}$, $\boldsymbol{R}^T\delta\boldsymbol{R}$ must be skew-symmetric. Since $\boldsymbol{S}$ is symmetric, $(\boldsymbol{R}^T\delta\boldsymbol{R}):\boldsymbol{S}=0$. Finally,

$$
\begin{aligned}
\delta\Psi &= 2\mu\boldsymbol{F}:\delta\boldsymbol{F}-2\mu\mathrm{tr}(\delta\boldsymbol{S})+\lambda(J-1)J\boldsymbol{F}^{-T}:\delta\boldsymbol{F}\\
&= 2\mu\boldsymbol{F}:\delta\boldsymbol{F}-2\mu\boldsymbol{R}:\delta\boldsymbol{F}+\lambda(J-1)J\boldsymbol{F}^{-T}:\delta\boldsymbol{F}\\
\frac{\partial\Psi}{\partial\boldsymbol{F}}:\delta\boldsymbol{F} &= \big(2\mu\boldsymbol{F}-2\mu\boldsymbol{R}+\lambda(J-1)J\boldsymbol{F}^{-T}\big):\delta\boldsymbol{F}\\
\frac{\partial\Psi}{\partial\boldsymbol{F}_E} &= 2\mu(\boldsymbol{F}_E-\boldsymbol{R}_E)+\lambda(J_E-1)J_E\boldsymbol{F}_E^{-T}
\end{aligned}
$$

Note that Cauchy stress $\boldsymbol{\sigma}$ and first Piola-Kirchhoff stress $\boldsymbol{P}$ are related to $\frac{\partial\Psi}{\partial\boldsymbol{F}_E}$ by

$$
\boldsymbol{\sigma}=\frac{1}{J}\frac{\partial\Psi}{\partial\boldsymbol{F}_E}\boldsymbol{F}_E^T=\frac{2\mu}{J}(\boldsymbol{F}_E-\boldsymbol{R}_E)\boldsymbol{F}_E^T+\frac{\lambda}{J}(J_E-1)J_E\boldsymbol{I}\qquad\boldsymbol{P}=\frac{\partial\Psi}{\partial\boldsymbol{F}_E}\boldsymbol{F}_P^{-T}
$$

The second derivatives require a bit more care but can be computed relatively easily.

$$
\begin{aligned}
\frac{\partial^2\Psi}{\partial\boldsymbol{F}\partial\boldsymbol{F}}:\delta\boldsymbol{F} &= \delta\left(\frac{\partial\Psi}{\partial\boldsymbol{F}}\right)\\
&= \delta(2\mu(\boldsymbol{F}-\boldsymbol{R})+\lambda(J-1)J\boldsymbol{F}^{-T})\\
&= 2\mu\delta\boldsymbol{F}-2\mu\delta\boldsymbol{R}+\lambda J\boldsymbol{F}^{-T}\delta J+\lambda(J-1)\delta(J\boldsymbol{F}^{-T})\\
&= 2\mu\delta\boldsymbol{F}-2\mu\delta\boldsymbol{R}+\lambda J\boldsymbol{F}^{-T}(J\boldsymbol{F}^{-T}:\delta\boldsymbol{F})+\lambda(J-1)\delta(J\boldsymbol{F}^{-T})
\end{aligned}
$$

Since $J\boldsymbol{F}^{-T}$ is a matrix whose entries are polynomials in the entries of $\boldsymbol{F}$, $\delta(J\boldsymbol{F}^{-T}) = \frac{\partial}{\partial \boldsymbol{F}}(J\boldsymbol{F}^{-T}) : \delta\boldsymbol{F}$ can readily be computed directly. That leaves the task of computing $\delta\boldsymbol{R}$.

$$\begin{aligned} \delta\boldsymbol{F} &= \delta\boldsymbol{R}\boldsymbol{S} + \boldsymbol{R}\delta\boldsymbol{S} \\ \boldsymbol{R}^T\delta\boldsymbol{F} &= (\boldsymbol{R}^T\delta\boldsymbol{R})\boldsymbol{S} + \delta\boldsymbol{S} \\ \boldsymbol{R}^T\delta\boldsymbol{F} - \delta\boldsymbol{F}^T\boldsymbol{R} &= (\boldsymbol{R}^T\delta\boldsymbol{R})\boldsymbol{S} + \boldsymbol{S}(\boldsymbol{R}^T\delta\boldsymbol{R}) \end{aligned}$$

Here we have taken advantage of the symmetry of $\delta\boldsymbol{S}$ and the skew symmetry of $\boldsymbol{R}^T\delta\boldsymbol{R}$. There are three independent components of $\boldsymbol{R}^T\delta\boldsymbol{R}$, which we can solve for directly. The equation is linear in these components, so $\boldsymbol{R}^T\delta\boldsymbol{R}$ can be computed by solving a 3×3 system. Finally, $\delta\boldsymbol{R} = \boldsymbol{R}(\boldsymbol{R}^T\delta\boldsymbol{R})$.

References

[1] I. Alduán and M.A. Otaduy. SPH granular flow with friction and cohesion. In *Proc. of the 2011 ACM SIGGRAPH/Eurographics Symp. on Comp. Anim.*, pages 25–32, 2011.

[2] I. Alduán, A. Tena, and M.A. Otaduy. Simulation of high-resolution granular media. In *Proc. of Congreso Español de Informática Gráfica*, volume 1, 2009.

[3] J. Barbič and Y. Zhao. Real-time large-deformation substructuring. *ACM Trans. Graph.*, 30:91:1–91:8, 2011.

[4] A. Bargteil, C. Wojtan, J. Hodgins, and G. Turk. A finite element method for animating large viscoplastic flow. *ACM Trans. Graph.*, 26, 2007.

[5] A.W. Bargteil, C. Wojtan, J.K. Hodgins, and G. Turk. A finite element method for animating large viscoplastic flow. In *ACM Trans. on Graph.*, volume 26, page 16, 2007.

[6] N. Bell, Y. Yu, and P.J. Mucha. Particle-based simulation of granular materials. In *Proc. of the 2005 ACM SIGGRAPH/Eurographics symposium on Comp. animation*, pages 77–86, 2005.

[7] B. A. Berg. *Markov chain Monte Carlo simulations and their statistical analysis*. World Scientific, 2004.

[8] J. Bonet and R.D. Wood. *Nonlinear continuum mechanics for finite element analysis*. Cambridge University Press, 1997.

[9] R.L. Brown. A volumetric constitutive law for snow based on a neck growth model. *J. of Appl. Phys.*, 51(1):161–165, 1980.

[10] R. S. Burt. A note on missing network data in the general social survey. *Social Networks*, 9:63–73, 1987.

[11] S. Capell, S. Green, B. Duchamp, and Z. Popovic. A multiresolution framework for dynamics deformations. In *Proc. Symp. Comp. Anim.*, pages 131–140, 2004.

[12] B. Chanclou, A. Luciani, and A. Habibi. Physical models of loose soils dynamically marked by a moving object. In *Comp. Anim.'96. Proc.*, pages 27–35, 1996.

[13] I. Chao, U. Pinkall, P. Sanan, and P. Schröder. A simple geometric model for elastic deformations. *ACM Trans. on Graph.*, 29(4):38, 2010.

[14] W.F. Chen and A.F. Salee. *Constitutive equations for engineering materials: elasticity and modelling*. John Wiley & Sons Inc, 1982.

[15] I. Coony, D. Hutchins, K. Lee, M. Harris, T. Molinder, and K. Suong. Prep and landing Christmas in July: the effects snow process. In *ACM SIGGRAPH 2010 talks*, 2010.

[16] S. Cresseri, F. Genna, and C. Jommi. Numerical integration of an elastic–viscoplastic constitutive model for dry metamorphosed snow. *Intl. J. for Num. and Anal. Meth. in geomechanics*, 34(12):1271–1296, 2010.

[17] S. Cresseri and C. Jommi. Snow as an elastic viscoplastic bonded continuum: a modelling approach. *Italian Geotechnical J.*, 4:43–58, 2005.

[18] G. Debunne, M. Desbrun, M. Cani, and A. Barr. Dynamic real-time deformations using space and time adaptive sampling. In *Proc. SIGGRAPH*, pages 31–36, 2001.

[19] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39:1–38, 1977.

[20] D.C. Drucker and W. Prager. Soil mechanics and plastic analysis or limit design. *Quaterly of App. Math.*, 10:157–165, 1952.

[21] D. Dutykh, C. Acary-Robert, and D. Bresch. Mathematical modeling of powder-snow avalanche flows. *Studies in Appl. Math.*, 127(1):38–66, 2011.

[22] M. Egesdal, C. Fathauer, K. Louie, and J. Neuman. Statistical modeling of gang violence in los angeles. *SUIRO*, 2010.

[23] O. Etzmuss, M. Keckeisen, and W. Strasser. A fast finite element solution for cloth modeling. In *Proc. Pac. Graph.*, pages 244–251, 2003.

[24] F. Faure, B. Gilles, G. Bousquet, and D. Pai. Sparse meshless models of complex deformable solids. *ACM Trans. Graph.*, 30, 2011.

[25] P. Fearing. Computer modelling of fallen snow. In *Proc. of the 27th annual conf. on Comp. Graph. and interactive techniques*, pages 37–46, 2000.

[26] P. Fearing. *The computer modelling of fallen snow*. PhD thesis, University of British Columbia, 2000.

[27] B.E. Feldman and J.F. O'Brien. Modeling the accumulation of wind-driven snow. In *ACM SIGGRAPH 2002 conf. abstracts and applications*, pages 218–218, 2002.

[28] T.G. Goktekin, A.W. Bargteil, and J.F. O'Brien. A method for animating viscoelastic fluids. In *ACM Trans. on Graph.*, volume 23, pages 463–468, 2004.

[29] O. Gonzalez and A. Stuart. *A first course in continuum mechanics*. Cambridge Texts in Applied Mathematics, 2008.

[30] D. Gray and D. Male. *Handbook of snow: principles, processes, management & use*. Pergamon Press, 1981.

[31] E. Grinspun, P. Krysl, and Peter P. Schröder. CHARMS: a simple framework for adaptive simulation. *ACM Trans. Graph.*, 21:281–290, 2002.

[32] A. G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58:83–90, 1971.

[33] A. G. Hawkes and D. Oakes. A cluster process representation of a self-exciting process. *Journal of Applied Probability*, 11:493–503, 1974.

[34] T. Hinks and K. Museth. Wind-driven snow buildup using a level set approach. In *Eurographics Ireland Workshop Series*, volume 9, pages 19–26, 2009.

[35] P. D. Hoff. Multiplicative latent factor models for description and prediction of social networks. *Computational and Mathematical Organization Theory*, 15:261–272, 2009.

[36] P. D. Hoff, A. E. Raftery, and M. S. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97:1090–1098, 2002.

[37] M. Huisman. Imputation of missing network data: some simple procedures. *Journal of Social Structure*, 10, 2009.

[38] M. Huisman and C. E. G. Steglich. Treatment of non-response in longitudinal network studies. *Social Networks*, 30:297–308, 2008.

[39] M. Ihmsen, A. Wahl, and M. Teschner. High-resolution simulation of granular material with SPH. In *Workshop on Virtual Reality Interaction and Phys. Sim.*, pages 53–60, 2012.

[40] G. Irving, J. Teran, and R. Fedkiw. Invertible finite elements for robust simulation of large deformation. In *Proc. Symp. Comp. Anim.*, pages 131–140, 2004.

[41] B. A. Jacobs and R. Wright. Street justice: retaliation in the criminal underworld. 2006.

[42] R. Keiser, B. Adams, D. Gasser, P. Bazzi, P. Dutré, and M. Gross. A unified lagrangian approach to solid-fluid animation. In *Point-Based Graph., 2005. Eurographics/IEEE VGTC Symp. Proc.*, pages 125–148, 2005.

[43] L. Kharevych, W. Yang, Y. Tong, E. Kanso, J. Marsden, and P. Schröder. Geometric, variational integrators for computer animation. In *Proc. Symp. Comp. Anim.*, pages 43–51, 2006.

[44] T. Kim, D. Adalsteinsson, and M. Lin. Modeling ice dynamics as a thin-film Stefan problem. In *Proc. 2006 ACM SIGGRAPH/Eurographics Symp. Comput. Anim.*, pages 167–176, 2006.

[45] T. Kim and D. James. Skipping steps in deformable simulation with online model reduction. *ACM Trans. Graph.*, 28, 2009.

[46] T. Kim and M. Lin. Visual simulation of ice crystal growth. In *Proc 2003 ACM SIGGRAPH/Eurographics Symp. Comput. Anim.*, pages 86–97, 2003.

[47] Tae-Yong Kim and Lucio Flores. Snow avalanche effects for Mummy 3. In *ACM SIGGRAPH 2008 talks*, 2008.

[48] K. Klohn, M. O'Brien, T. Speltz, and T. Wichitsripornkul. Building the snow footprint pipeline on Brave. In *ACM SIGGRAPH 2012 talks*, 2012.

[49] G. Kossinets. Effects of missing data in social networks. *Social Networks*, 28:247–268, 2006.

[50] T. Lenaerts and P. Dutré. Mixing fluids and granular materials. In *Comp. Graph. Forum*, volume 28, pages 213–218, 2009.

[51] D.I.W. Levin, J. Litven, G.L. Jones, S. Sueda, and D.K. Pai. Eulerian solid simulation with contact. *ACM Trans. on Graph.*, 30(4):36, 2011.

[52] E. Lewis, G. Mohler, P. J. Brantingham, and A. Bertozzi. Self-exciting point process models of insurgency in iraq. *UCLA CAM reports*, pages 10–38, 2010.

[53] A. Luciani, A. Habibi, and E. Manzotti. A multi-scale physical model of granular materials. In *Graphics Interface*, pages 136–136, 1995.

[54] Short M., D'Orsogna M., Pasour V., Tita G., Brantingham P., Bertozzi A., and Chayes L. A statistical model of criminal behavior. *Mathematical Models and Methods in Applied Sciences*, 18:1249–1267, 2008.

[55] N. Marechal, E. Guerin, E. Galin, S. Merillou, and N. Merillou. Heat transfer simulation for modeling realistic winter sceneries. *Comp. Graph. Forum*, 29(2):449–458, 2010.

[56] A. McAdams, Y. Zhu, A. Selle, M. Empey, R. Tamstorf, J. Teran, and E. Sifakis. Efficient elasticity for character skinning with contact and collisions. *ACM Trans. Graph.*, 30:37:1–37:12, 2011.

[57] G. Meschke, C. Liu, and H.A. Mang. Large strain finite-element analysis of snow. *J. of Engng. Mech.*, 122(7):591–602, 1996.

[58] V.J. Milenkovic. Position-based physics: simulating the motion of many highly interacting spheres and polyhedra. In *Proc. of the 23rd annual conf. on Comp. Graph. and interactive techniques*, pages 129–136, 1996.

[59] G. Miller and A. Pearce. Globular dynamics: a connected particle system for animating viscous fluids. *Comp. & Graph.*, 13(3):305–309, 1989.

[60] G. O. Mohler, M. B. Short, P. J. Brantingham, F. P. Schoenberg, and G. E. Tita. Self-exciting point process modeling of crime. *Journal of the American Statistical Association*, 106(493):100–108, 2011.

[61] M. Müller, J. Dorsey, L. McMillan, R. Jagnow, and B Cutler. Stable real-time deformations. In *Proc. Symp. Comp. Anim.*, pages 49–54, 2002.

[62] M. Müller and M. Gross. Interactive virtual materials. In *Proc. Graph. Int.*, pages 239–246, 2004.

[63] R. Narain, A. Golas, and M.C. Lin. Free-flowing granular materials with two-way solid coupling. In *ACM Trans. on Graph.*, volume 29, page 173, 2010.

[64] F. Nicot. Constitutive modelling of snow as a cohesive-granular material. *Granular Matter*, 6(1):47–60, 2004.

[65] T. Nishita, H. Iwasaki, Y. Dobashi, and E. Nakamae. A modeling and rendering method for snow by using metaballs. In *Comp. Graph. Forum*, volume 16, pages C357–C364, 1997.

[66] J. O'Brien, A. Bargteil, and J. Hodgins. Graphical modeling and animation of ductile fracute. *ACM Trans. on Graph.*, 21(3):291–294, 2002.

[67] Y. Ogata. The asymptotic behaviour of maximum likelihood estimators for stationary point processes. *Annals of the Institute of Statistical Mathematics*, 30:243–261, 1978.

[68] Y. Ogata. Space-time point process models for earthquake occurrences. *Ann. Inst. Statist. Math.*, 50:379–402, 1988.

[69] Y. Ogata and M. Tanemura. Likelihood analysis of spatial point patterns. *Journal of the Royal Statistical Society, Series B*, 46:496–518, 1984.

[70] W. Ogden. *Nonlinear elastic deformations*. Dover, 1998.

[71] M. Pauly, R. Keiser, B. Adams, P. Dutré, M. Gross, and L.J. Guibas. Meshless animation of fracturing solids. In *ACM Trans. on Graph.*, volume 24, pages 957–964, 2005.

[72] M. Pla-Castells, I. García-Fernández, and R. Martínez. Interactive terrain simulation and force distribution models in sand piles. *Cellular Automata*, pages 392–401, 2006.

[73] R. A. Rutenbar. Simulated annealing algorithms: an overview. *IEEE Circuits and Devices Magazine*, 5:19–26, 1989.

[74] S. Sane. *Rate dependent constitutive modeling of engineering materials: model development and numerical implementation*. Lightning Source Incorporated, 2009.

[75] J. L. Schafer and J. W. Graham. Missing data: our view of the state of the art. *Psychological Methods*, 7:147–177, 2002.

[76] R. Schmedding and M. Teschner. Inversion handling for stable deformable modeling. *Vis. Comp.*, 24:625–633, 2008.

[77] A. Selle, M. Lentine, and R. Fedkiw. A mass spring model for hair simulation. In *ACM Trans. on Graph.*, volume 27, pages 64.1–64.11, 2008.

[78] A. Shamir, V. Pascucci, and C. Bajaj. Multi-resolution dynamic meshes with arbitrary deformations. In *Proc. IEEE Vis.*, pages 423–430, 2000.

[79] M. Short, M. D'Orsogna, P. Brantingham, and G. Tita. Measuring and modeling repeat and near-repeat burglary effects. *Journal of Quantitative Criminology*, 25:325–339, 2009.

[80] M. Short, G. Mohler, P. J. Brantingham, and G. Tita. Gang rivalry dynamics via coupled point process networks. *Discrete and Continuous Dynamical Systems*, 2013.

[81] F. Sin, Y. Zhu, Y. Li, D. Schroeder, and J. Barbic. Invertible isotropic hyperelasticity using svd gradients. In *Poster: Symp. Comp. Anim.*, pages 1–2, 2011.

[82] W. St Lawrence and C.C. Bradley. The deformation of snow in terms of structural mechanism. In *Snow Mech. Symp.*, page 155, 1975.

[83] M. Steffen, R. Kirby, and M. Berzins. Analysis and reduction of quadrature errors in the material point method (MPM). *Int. J. Numer. Meth. Engng*, 76(6):922–948, 2008.

[84] D. Sulsky, S.-J. Zhou, and H.L. Schreyer. Application of particle-in-cell method to solid mechanics. *Comp. Phys. Comm.*, 87:236–252, 1995.

[85] R.W. Sumner, J.F. O'Brien, and J.K. Hodgins. Animating sand, mud, and snow. In *Comp. Graph. Forum*, volume 18, pages 17–26, 1999.

[86] J. Teran, E. Sifakis, G. Irving, and R. Fedkiw. Robust quasistatic finite elements and flesh simulation. In *Proc. Symp. Comp. Anim.*, pages 181–190, 2005.

[87] D. Terzopoulos and W. Fleischer. Modeling inelastic deformation: viscoelasticity, plasticity, fracture. *Proc. ACM SIGGRAPH 1988*, 22(4):269–278, 1988.

[88] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *ACM SIGGRAPH*, 21:205–214, 1987.

[89] M. Teschner, B. Heidelberger, M. Müller, and M. Gross. A versatile and robust model for geometrically complex deformable solids. In *Proc. Comp. Graph. Int.*, pages 312–319, 2004.

[90] G. Tita. *Reducing gun violence: results from an intervention in East Los Angeles.* RAND Corporation, Santa Monica, CA, 2003.

[91] G. Tita and G. Ridgeway. The impact of gang formation on local patterns of crime. *Journal of Research in Crime and Delinquency*, 44:208–237, 2007.

[92] C. Truesdell and W. Noll. *The non-linear field theories of mechanics.* Springer, 2004.

[93] M. Wicke, D. Ritchie, B. Klingner, S. Burke, J. Shewchuk, and J. O'Brien. Dynamic local remeshing for elastoplastic simulation. *ACM Trans. Graph.*, 29:49:1–49:11, 2010.

[94] W.J. Wiscombe and S.G. Warren. A model for the spectral albedo of snow. i: pure snow. *J. of the Atmospheric Sciences*, 37(12):2712–2733, 1980.

[95] C. Wojtan, N. Thürey, M. Gross, and G. Turk. Deforming meshes that split and merge. *ACM Trans. Graph.*, 28:76:1–76:10, 2009.

[96] C. Wojtan and G. Turk. Fast viscoelastic behavior with thin features. In *ACM Trans. on Graph.*, volume 27, page 47, 2008.

[97] B. Zhu and X. Yang. Animating sand as a surface flow. *Eurographics 2010, Short Papers*, 2010.

[98] Y. Zhu and R. Bridson. Animating sand as a fluid. *ACM Trans. on Graph.*, 24(3):965–972, 2005.

[99] Y. Zhu, E. Sifakis, J. Teran, and A. Brandt. An efficient and parallelizable multigrid framework for the simulation of elastic solids. *ACM Trans. Graph.*, 29:16:1–16:18, 2010.

[100] J. Zhuang, Y. Ogata, and D. Vere-Jones. Stochastic declustering of space-time earthquake occurences. *JASA*, 97:369–380, 2002.