2009

Instabilities of elastic bodies in motion

Brian Douglas Gillispie

University of Iowa

Copyright 2009 Brian Douglas Gillispie

This dissertation is available at Iowa Research Online: http://ir.uiowa.edu/etd/238

Recommended Citation


Follow this and additional works at: http://ir.uiowa.edu/etd

Part of the Mathematics Commons
INSTABILITIES OF ELASTIC OBJECTS IN MOTION

by

Brian Douglas Gillispie

An Abstract

Of a thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Mathematics in the Graduate College of The University of Iowa

May 2009

Thesis Supervisor: Professor David E. Stewart
This thesis consists of two parts. In the first part, a simulation of contact between a two dimensional elastic object in motion in contact with a rigid obstacle under Coulumb friction is developed. Once this simulation is developed, the simulation attempts to find out what initial numerical conditions cause the object to go unstable. All results found in this first part are numerical. Based on the numerical results of this thesis it appears that the friction coefficient is the main source of instabilities occuring, but the initial velocity does impact whether or not an instability occurs to some degree as well. The second part of this thesis investigates whether or not energy is conserved during the simulations conducted in the first part of the thesis. In the first part we discover that whether or not energy is conserved has little impact on whether an instability occurs, but it is still desireable for energy to be conserved in a mathematical model. We show that with a slight modification to the friction condition that energy will be conserved. This is desireable as energy dissipation is usually needed to prove convergence of solutions. Unfortunately as of yet convergence of solutions cannot be proven as existance of solutions has yet to be established.
Abstract Approved: ______________________________
Thesis Supervisor

______________________________
Title and Department

______________________________
Date
INSTABILITIES OF ELASTIC OBJECTS IN MOTION

by

Brian Douglas Gillispie

A thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Mathematics in the Graduate College of The University of Iowa

May 2009

Thesis Supervisor: Professor David E. Stewart
This is to certify that the Ph.D. thesis of

Brian Douglas Gillispie

has been approved by the Examining Committee for the thesis requirement for the Doctor of Philosophy degree in Mathematics at the May 2009 graduation.

Thesis Committee:  

David E. Stewart, Thesis Supervisor

Weiman Han

Daniel Anderson

Palle Jorgenson

Chris Wyman
“Penalty method? Bad function, go sit in the corner” -Overheard in a class on contact mechanics, Spring 2005
ACKNOWLEDGMENTS

I’d like to start by thanking my advisor, David Stewart for his guidance and support and assistance while writing this thesis. I’d also like to thank Theodore Wendt for his collaboration on numerous projects throughout my time in graduate school. In addition, I’d like to thank Daniel Anderson, Palle Jorgenson, Chris Wyman, and Weimin Han for being on my thesis committee and their feedback on this paper. I would also like to thank my parents for their endless support while in graduate school, as well as my close friends who kept me going even when things got difficult.
TABLE OF CONTENTS

LIST OF TABLES ........................................................................................................... viii

LIST OF FIGURES ......................................................................................................... ix

CHAPTER

1 INTRODUCTION ........................................................................................................... 1

1.1 Introduction ............................................................................................................. 1
1.2 Bodies in Motion ..................................................................................................... 1
  1.2.1 Rigid Bodies ....................................................................................................... 1
  1.2.2 Elastic Bodies ..................................................................................................... 2
  1.2.3 Equations of Motion .......................................................................................... 3
1.3 Boundary Conditions .............................................................................................. 4
  1.3.1 Dirichlet Boundary Conditions .......................................................................... 4
  1.3.2 Neumann Boundary Conditions ......................................................................... 5
  1.3.3 The Linear Complementarity Problem .............................................................. 6
1.4 Theoretical Difficulties ............................................................................................ 7
  1.4.1 Existence and Uniqueness of Solutions .............................................................. 7
  1.4.2 Energy Conservation ......................................................................................... 9
  1.4.3 Instabilities ....................................................................................................... 10
1.5 Outline .................................................................................................................... 13

2 PRELIMINARIES ........................................................................................................ 14

2.1 Definitions .............................................................................................................. 14
2.2 Theorems ............................................................................................................... 17

3 SOLVING THE EQUATIONS OF MOTION ............................................................ 18

3.1 Modeling the Problem ............................................................................................ 18
  3.1.1 Equations of Motion and Boundary Conditions ................................................ 18
3.2 Discretizing the Problem ....................................................................................... 21
  3.2.1 Basic Setup ....................................................................................................... 21
  3.2.2 Discretization in Time ....................................................................................... 23
  3.2.3 Discretization in Space ..................................................................................... 25
  3.2.4 Problems in solving the equations for N and F .................................................. 28
3.3 The Jacobi Method Setup ....................................................................................... 29
  3.3.1 Solving for N ..................................................................................................... 29
  3.3.2 Solving for F ..................................................................................................... 34
3.4 Implementation Decisions in this Model Clarified ................................................ 36
3.4.1 Why \( \mu = 6.33? \) .................................................. 37
3.4.2 Why \( E = 210? \) .................................................. 37
3.4.3 Why Results Can Vary Across Computers .................. 38

4 RESULTS .................................................................. 39

4.1 Results from the Jacobi Method ................................. 39
4.1.1 Parameters for the Experiments ............................... 39
4.1.2 Frictionless Experiments ....................................... 41
4.1.3 No Initial Velocity Experiments .............................. 41
4.1.4 Variance of Initial Boundary Conditions .................. 43
4.1.5 Variance of Friction Coefficient ............................. 47
4.1.6 Variance of Timesteps and Friction Coefficient .......... 49
4.1.7 Cases with Different Number of Nodes .................... 52
4.1.8 Variance of Node Spacing ................................... 53
4.1.9 Removal of Stress and Strain Terms ....................... 55

4.2 Interpretation of the Results of the Jacobi Method ........ 56
4.2.1 Significance of Smaller Timesteps ........................... 56
4.2.2 Significance of Starting Boundary Conditions .......... 57
4.2.3 Significance of Number of Nodes ........................... 58
4.2.4 Significance of the No Grad Div Experiments .......... 59
4.2.5 Significance of Energy Conservation ...................... 59
4.2.6 Summary of Results ........................................... 60

4.3 What Occurs During an Unstable Experiment ............... 61
4.3.1 Special Notes ................................................... 61
4.3.2 Leading up to the Instability ................................. 62
4.3.3 At and After the Instability .................................. 66
4.3.4 Changes in the Normal Contact Force and Friction .... 68
4.3.5 Comparison with Stable Experiments ..................... 72
4.3.6 Implications .................................................... 73

4.4 Tables And Figures ................................................. 75

5 THEORETICAL RESULTS ........................................... 90

5.1 Is The Current Method Energy Conserving? ................ 90
5.2 Uniqueness of Solutions to the LCP ......................... 94
5.2.1 The Jacobi Model ............................................. 94

6 CONCLUSION AND FUTURE WORK ........................... 102

6.1 Future Research ..................................................... 102
6.1.1 A Changed Model ............................................. 102
6.1.2 Further Investigation Into the Current Model .......... 103
6.1.3 Change to the Finite Element Method ..................... 103
## LIST OF TABLES

Table

4.1 Frictionless Experiments, 25 nodes ........................................... 41
4.2 Variance of Friction Coefficient, 25 Nodes ................................. 76
4.3 Variance of Friction Coefficient, 10 Nodes .................................. 77
4.4 Variance of Friction Coefficient, 5 Nodes ................................... 78
4.5 No Grad Div Term Experiments .................................................... 79
LIST OF FIGURES

Figure

4.1 Energy Gain Per Timestep ........................................ 42
4.2 Variance of Starting Displacements ............................ 43
4.3 Variance of Starting Velocities, h = .00005 .................... 44
4.4 First Energy Increase with Velocity Varied .................... 45
4.5 Variance of Starting Velocity, h = .0001 ....................... 46
4.6 Variance of Starting Velocity, h = .00001 ..................... 47
4.7 Variance of Friction Coefficient ............................... 48
4.8 First Energy Increase vs Friction Coefficient .................. 49
4.9 Variance of h on Instabilities, 25 nodes, \( \mu = .5 \) .................. 50
4.10 Zoomed In View of Figure 9 .................................. 51
4.11 Variance of Number of Nodes .................................. 53
4.12 Variance of Number of Nodes, Fixed Object Length .......... 54
4.13 Average Normal Contact Force ................................ 71
4.14 Average Normal Contact Force, 2nd Graph ................... 72
4.15 14 Timesteps ...................................................... 80
4.16 500 Timesteps ..................................................... 80
4.17 541 Timesteps ..................................................... 81
4.18 2500 Timesteps ................................................... 81
4.19 5000 Timesteps ................................................... 82
4.20 7500 timesteps .................................................... 82
CHAPTER 1
INTRODUCTION

1.1 Introduction

Sliding objects are everywhere in our life. From cars on the freeway to books sliding on the table as we pass them to a friend, we are surrounded by objects in motion that are sliding along a surface. Even our everyday walking can be regarded as an object in motion in contact with the ground. Mathematically modeling these sliding objects is of use in robotics [1, 36] collision detection [6, 7, 8, 9], and even video games [24, 11]. Due to this, there is a heavy interest in modeling the object in motion accurately. More accurate modeling of an object in motion will result in better collision modeling algorithms and better quality video games, just to name a couple of applications. However, in order to model these objects in motion, mathematical representation of the object in motion is needed first.

1.2 Bodies in Motion

We will begin with an overview of the differences between rigid and elastic objects.

1.2.1 Rigid Bodies

One way to represent a sliding object is to assume that the object remains rigid while it is in motion. By rigid, we mean here that the total length, width, height and shape of the object remains the same while the object is in motion. The object does not stretch or bend while moving. This results in internal forces being ignored while the object is moving.
along its path. Rigid body dynamics is a reasonably accurate way to represent objects like pool balls and concrete slabs, as they tend to not stretch or bend much when they are in motion. However, it would be a poor way to model a block of Jell-O in motion, as the stretching and even internal collapsing of the object will not be represented in the model. Still, it is a widely popular approach to modeling an object in motion as computing external forces on the object is usually pretty straightforward. For a full discussion of rigid body motion and how to solve these kind of problems, see [33, 6, 7, 8, 9, 21]

1.2.2 Elastic Bodies

Another way to represent an object in motion is to allow for internal and external forces on the object to cause the object to bend and stretch. This results in a much more accurate representation of the object in motion now, as forces like sheer, stretching, bending, and even shock waves within the object are now represented. However, this results in many more variables to keep track of, and adds to the computational complexity of the problem. In an elastic body, we have to now keep track of stress and strain, which we will now define. Also, to understand stress, we need to define what we mean by deformation, so that term will be defined as well.

**Definition 1.1.** The stress experienced by a point in an elastic body is the average force per unit area. In two dimensions, we represent the stress by a tensor as:

\[
\sigma = \begin{pmatrix}
\sigma_{11} & \sigma_{12} \\
\sigma_{21} & \sigma_{22}
\end{pmatrix}
\]

Where \( \sigma_{ij} v = t'_j \), and \( t_j = \lim_{\Delta s \to 0} \frac{\Delta f_j}{\Delta s} \), where \( f_j \) is the surface force in the \( j \)th
dimension. Note that \( t_j \) is the surface tension, and the notation \( t_j \) is used to emphasize the fact that the stress vector at a given point depends explicitly upon the particular surface element chosen, as represented by the unit normal \( \nu \). Also, note that the stress tensor is symmetric, so \( \sigma_{ij} = \sigma_{ji} \) for all \( i \) and \( j \) where the stress tensor is defined [26].

**Definition 1.2.** The *strain* represents the local stretching and compressing of an object about a point in a material. Note that the strain tensor must ignore the effects of rotations. For linearized elasticity, we use for strain the following definition:

\[
\varepsilon = \frac{1}{2}((\nabla u) + (\nabla u)^T)
\]

This is often referred to as the *Eulerian Infinitesimal Strain Tensor*.

1.2.3 Equations of Motion

With stress and strain now defined, the equations of motion for a dynamic object in motion can now be given. They are:

\[
\text{div}\sigma + f = \rho u_{tt} \text{ in } \Omega
\]

This equation is often referred to as the *equation of motion for an elastic body in dynamic (or constant) contact*. Sometimes \( u_{tt} \) is assumed to be 0, in this case we are treating inertial forces as negligible. We call the problem where \( u_{tt} = 0 \) is assumed as the *quasistatic contact problem*, and the corresponding problem

\[
\text{div}\sigma + f = 0
\]
is called the equation of equilibrium. For more on the quasistatic problem, I refer the reader to [17, 2, 3].

1.3 Boundary Conditions

In modeling an object in motion, it is a reasonable assumption to assume that the behavior of an internal point depends on the points around it. This means that the points on the boundary of the object will then be affected by what is going on around it. Due to this, boundary conditions need to be developed to make the model work properly. Most of the common boundary conditions have been developed by Neumann and Dirichlet. An explanation of each of the two boundary conditions follow, as well as an explanation on how they relate to the problem being discussed in this thesis.

1.3.1 Dirichlet Boundary Conditions

The first common boundary condition used on these kind of problem are referred to as the Dirichlet Boundary conditions, or the Dirichlet problem. In the Dirichlet problem, we assume that part (or all) of the boundary is held at a fixed point in space and is not allowed to move. The Dirichlet problem often looks like

\[ u = g \]

on \( \partial \Omega \), where \( u \) is the displacement of the object on \( \Omega \) and \( g \) is a function that tells us what the displacement should be on the boundary.

The Dirichlet problem has the effect of clamping the object in place along part of the boundary. As we are studying an object in motion in this thesis, Dirichlet boundary
conditions are not used in these experiments, as we want the object to be free to move. For more on the Dirichlet problem, I refer the reader to any book on differential equations or partial differential equations (for instance [10]).

1.3.2 Neumann Boundary Conditions

Sometimes we wish to describe not what the displacement should be on the boundary, but instead describe what the rate of change is on the boundary. These kind of problems are referred to as Neumann problems. The Neumann boundary condition often looks like

\[ \nabla u \cdot \nu = g \text{ on } \partial \Omega \]

where \( \nabla u \) is the gradient of the displacement, \( \nu \) is the outward unit normal on the boundary, \( \cdot \) means take the dot product of the two terms, and \( g \) is a function that tells us what the gradient is supposed to be on the boundary [10].

The Neumann problem has the effect of providing a force to the object (for \( g \neq 0 \)). Providing a Neumann boundary condition to an object in motion can in effect mimic a continuous force being applied to the object.

For the problem that is modeled in this thesis, we need a force provided to the bottom boundary of the object to keep it from sinking into the ground. However, we only need the force provided when the object is in contact, and no other times. The Neumann conditions are not sufficient for this as the Neumann conditions (as currently stated) are set up to always provide (or not provide) the force to the object. Therefore, another kind of boundary conditions are needed for the experiments in this thesis, something that is similar
to the Neumann problem.

1.3.3 The Linear Complementarity Problem

What is needed for these experiments is a type of boundary conditions that are in effect only when the object is in contact with the ground. When the object is in contact with the ground, a force needs to be exerted by the ground that will keep the object from penetrating the ground. If the object is not in contact with the ground, no such force is needed. We will call the force exerted by the ground the normal contact force, represented by $N$. So what is needed to occur is:

$$N = 0 \text{ if } u > 0$$

$$N > 0 \text{ if } u = 0$$

Another way to look at this is to say that the following needs to occur:

$$0 \leq N \perp u \geq 0$$

where $N \cdot u$ must be 0 (denoted by the $\perp$ symbol). In the event that $N$ and $u$ are vectors and not scalars, the problem is interpreted to mean the condition must be met component wise, or in other words:

$$0 \leq N_i \perp u_i \geq 0$$

with $N_i \cdot u_i = 0 \forall i$. Problems like these are referred to as linear complementarity problems,
or LCP’s for short. Extensive study has been conducted on LCP’s, on when the problems have solutions and when do they not have solutions, as well as finding different methods for solving LCP’s. For a good overview of the different methods of solving LCP’s and the theoretical difficulties that arise when solving them, see [15]. Note that both \( N_i \) and \( u_i \) can be 0 in an LCP, and that at all times one of the terms in the LCP must be zero.

1.4 Theoretical Difficulties

Despite extensive study in this field, there remain quite a few open questions and unresolved issues. An overview of those issues are given next.

1.4.1 Existence and Uniqueness of Solutions

One of the obvious questions that needs to be considered when looking at a mathematical problem is whether or not solutions exist, and are they unique. Solutions of contact problems are not guaranteed to be unique, Painlevé’s paradox [33] shows us that for a rod in contact with a surface and in motion that two different, valid solutions can exist to the same contact problem. For this reason, some researchers feel that the contact problem does not admit unique solutions, and are focusing on finding when and if solutions exist.

However, even showing existence of solutions for the mathematical formulation has been difficult. The problem with solving the contact problems is that the normal contact force (or \( N \)) is usually not smooth. In order for solutions to exist, some regularity of the normal contact forces is required. Gaetano Fichera addressed this problem for the static problem by reposing the contact problem in terms of the minimization of energy over space. This technique allows for \( N \) to be essentially removed from the equation, and therefore
existence of solutions can be shown. Using this approach, Necas, Jarusek and Haslinger [28], then Jarusek [20], and then Kato [22], followed more recently by Eck and Jarusek [16], were able to show that for a sufficiently small friction coefficient, that solutions will exist to the Coulomb frictional contact problem.

Other approaches besides reposing the problem in terms of minimization of energy have been used to show that solutions exist. M. Cocou showed that via a regularization of contact stress and using a constant friction coefficient that a weak solution exists to the dynamic friction problem. Kuttler and Shillor [23] extend on this work to allow for allow for slip rate dependent friction, as well as for a variance in the values of the friction coefficient. Kuttler and Shillor also show in their work that the friction coefficient can jump from one value to another discontinuously and a solution will still exist to the problem. This accounts for the jump from dynamic to static friction, though some dispute does exist in the mathematical community over whether that is truly a discontinuous jump like is usually assumed.

Other researchers have used a normal compliance condition to represent objects in motion. Instead of assuming that the object that we are in contact with is fixed, the object is represented as a series of springs. This now allows for some interpenetration to occur. This technique is often called the penalty method, and is often used in papers and research dealing with this subject. The penalty method is used by many researchers, including [25] by D. E. Stewart, where existence and uniqueness of solutions are established. The penalty method is often used to approximate the LCP one gets when dealing with objects in motion and in contact. By doing this, one can avoid the problem of having to deal with totally rigid
9

Little has been discovered about uniqueness of the contact problem, and in fact there are some researchers who believe that the solutions to the contact problem may not be unique. Hassani shows in his paper [19] that for sufficient conditions in the problem that the Coulomb friction problem has infinite valid solutions, and even shows a numerical example where two solutions occur to the same problem. However, Hassani’s paper only shows this occurrence for the quasistatic problem, not the dynamic contact problem we are working with in this paper. No known work to date (to the author’s knowledge) has shown that solutions will not be unique for the dynamical contact problem.

1.4.2 Energy Conservation

One of the basic laws of physics is that an object in motion will eventually slow down and stop due to the force of gravitation and friction acting on the object. In addition, in the absence of external forces, the object will, once at rest, remain at rest. In both of these cases, the object in motion eventually slows down and comes to a stop. One will notice that if they push an object across a smooth surface (easily tested by sliding a book across a table), the object does not leave the table. Gravity pushes down on the book, and in return, the table exerts a force on the book to keep it from penetrating the table. One would expect, therefore, that an object put in motion and acted on by no other external forces (except for gravity) will not build up or gain energy. Therefore, it seems reasonable to expect that the total energy of the object remains constant over time, or even reduce or dissipate over time.
Even though energy conservation is desirable, there have been significant difficulties in proving that in the mathematical model for an elastic object in motion under friction, energy remains conserved. Stewart, in his paper [33] proves that energy will decrease between each timestep for an elastic body on a frictionless surface. Theodore Wendt in his paper [36] proves that for a viscoelastic object in motion, and under friction and mixed boundary conditions, that energy will remain conserved. Also, it is well documented that rigid bodies will gain or lose energy when a collision occurs (easily tested at your local pool hall) [35]. However, as of this paper no one has proved that the energy of an elastic body in motion under friction will remain conserved.

1.4.3 Instabilities

While modeling an object in motion under Coulumb friction, unexpected behavior tends to occur. These behaviors are usually referred to as instabilities in the literature. Examples of documented instabilities are models of brakes on cars where the brakes start to squeal unexpectedly when they should not [5, 27], and squeals occurring during contact between windows and glass. Other papers that deal with instabilities in some form are [37, 31, 29, 18]. However, in few of these cases did the author define what they were calling an instability. The only definition of an instability was given by [31] where an instability is defined to be when the wave speed inside the object goes complex. For the most part though, the term instability seems to be used to explain a mathematical behavior that is occurring which does not match up with what one expects to occur, usually based on real world observation.
Due to the fact that no solid definition of instability has ever been agreed upon in the mathematical research to date, I will give my own definition of what is considered to be an instability. For an elastic object in motion on a flat surface, in the absence of an external force, the object should not ever leave the surface. Therefore, the following definition of instability will be used:

**Definition 1.3.** An *Instability* is defined as an perturbation whose size does not depend on the size of the initial perturbation, or the perturbation has grown without bound.

Unfortunately, we can’t measure infinity or without bound, so some measurable definition needs to be used so we can declare that an experiment has gone unstable. If one pushes an object along a flat surface and never provides it with an external force to push it upwards, the object should never leave the ground. Therefore, it makes sense to define an experiment as unstable when the following occurs:

**Definition 1.4.** An experiment is said to have gone *unstable*, or for an instability to exist in the simulation, if in the simulation of an object in motion along a surface and no external forces applied upwards to the object, if the object leaves the ground by a significant margin. For our sake, we will say that the simulation is unstable if any part of the bottom boundary is more than .0005 units off of the ground, or in mathematical notation,

\[ u_i(y) \geq .0005 \]

at some particle \( i \) on the boundary which began in contact with the flat surface.

This definition will be used in practice to identify what does not happen in a steady-
state solution. The tolerance of .0005 (in whatever units are being used in the simulation) before saying the experiment has gone unstable is to allow for some rounding errors to occur by the simulation. In short, under this definition the simulation of the object in motion will be called to be unstable once the mathematical model stops agreeing sufficiently with the real world model that it is supposed to represent. At that point in time, the experimental model results will have become worthless for simulating an object sliding in motion on the ground.

Note that the choice of .0005 units is an experimental decision. Earlier experiments were attempted with .01 as the tolerance, and 99% of the cases that were flagged as unstable with the .0005 tolerance were flagged as unstable with the .01 tolerance. The author does recall one case that was not flagged as unstable with the .01 tolerance that was with the .0005 tolerance, but that could be because the experiment was not ran long enough in the .01 case to be flagged as unstable yet.

There has been some work done to show that these instabilities occur for sliding objects, but so far no one knows for sure what causes them to occur. All that current research has done is to show that they can occur under certain conditions, but no attempt to find out specifically what condition causes them to occur has been made to date (to the author’s knowledge). Renardy [29] proposes that under certain conditions for an incompressible object that instabilities will occur, but so far little known work has been done to extend this idea.
1.5 Outline

The rest of this thesis is ordered as follows. In Chapter 2 I will define what most of my common terms are for the rest of the paper.

Chapter 3 is dedicated to setting up and solving the equations of motion so that results can be achieved. Section 3.1 will set up the equations of motion and define all the terms used in section. In Section 3.2 I will show how to discretize the object in space and time so that a simulation can actually be run. Section 3.3 is dedicated to setting up the Jacobi method that will be used for solving for the friction and normal contact forces (those terms are also defined in Chapter 3). Section 3.4 is where I will explain some of the implementation decisions made in the experiment.

In Chapter 4, I will show the results from running experiments of a sliding elastic object in motion under Coulomb friction and on a flat surface, under no external forces (besides gravity) other than the initial push to get the object started. Section 4.1 is dedicated to showing all the results. In Section 4.2 I will go over and explain what these results seem to imply. Then, in Section 4.3 I will show what happens before, during, and after an instability.

Chapter 5 is dedicated any theoretical results relating to these experiments.

And finally, in Chapter 6 I will summarize all of the results and discuss what all of this means, as well as go over my plans for future work on this subject. Some of the remaining open questions on this subject will also be discussed here.
CHAPTER 2
PRELIMINARIES

In all of this thesis, unless noted otherwise, $u$ and $v$ are considered to be vector valued functions over $\mathbb{R}^n$, and $u = (u_1, u_2, \ldots, u_n)^T$, and similarly, $v = (v_1, v_2, \ldots, v_n)^T$.

2.1 Definitions

Here I will list the relevant definitions needed in this thesis. This first definition is useful in understanding the theorems stated in the next section

**Definition 2.1.** A matrix $M$ is said to be a $P$ matrix if all its principal minors are positive.

This next definition will define an important space referred to in this thesis

**Definition 2.2.** $w \in C^\infty(\Omega)$ means that in the region $\Omega$, the function $w : \Omega \rightarrow \mathbb{R}$ is infinitely differentiable.

In Chapter 1 I defined $\sigma$ as the stress tensor. Now I will give a way to compute $\sigma$ for an object in motion. To do that we need a few other terms to be defined first. The first term to be defined is Young’s Modulus

**Definition 2.3.** *Young’s Modulus*, denoted $E$, is the measure of stiffness of isotropic (properties are the same in all orientations) materials. It is defined as the ratio of the tensile stress divided by the ratio of the tensile strain, or also known as:

$$E = \frac{FL_0}{A_0\Delta L}$$
where $F$ is the force exerted, $L_0$ is the original length of the object, $A_0$ is original cross sectional area through which the force is applied, and $\Delta L$ is the amount by which the object changes. Note that Young’s Modulus is usually found experimentally in laboratories, and as such the values of $E$ are already known for many materials [26].

Another definition (though informal) for Young’s Modulus is the amount of pressure one needs to pull an object with in order to stretch it to double its original length.

**Definition 2.4. Poisson’s Ratio**, denoted $\nu$, is defined as the ratio of the transverse strain divided by the longitudinal strain. Or, it can be considered as the ratio of the strain normal to the applied load divided by the strain in the direction of the applied load [26]

Note that for a stable material, $-1.0 \leq \nu \leq 0.5$. Incompressible material is considered to have a Poisson’s ratio of 0.5

With Young’s Modulus and Poisson’s Ratio defined, we will now define one of the Lame Parameters, $\lambda$

**Definition 2.5.** Let $E$ be defined as Young’s Modulus and $\nu$ be defined as Poisson’s Ratio. Then $\lambda$ is defined as follows: $\lambda = \frac{EV}{(1+\nu)(1-2\nu)}$ [4].

Now we will define how to calculate $\omega$, which is another Lame Parameter

**Definition 2.6.** Let $E$ be defined as Young’s Modulus, and $\nu$ be defined as Poisson’s Ratio. Then $\omega$ is defined as follows: $\omega = \frac{E}{1+\nu}$ [4].

As $u$ is vector valued, a definition for $\nabla u$ is needed. That will be done next.

**Definition 2.7.** Let $u$ be a vector valued function defined on $\mathbb{R}^n$. Then $\nabla u$ is defined as an $n \times n$ matrix, where $(\nabla u)_{i,j} = \frac{\partial u_i}{\partial x_j}$
With all of those terms defined, we are now prepared to define $\varepsilon$, which is used in computing $\sigma$

**Definition 2.8.** Let $\nabla u$ be defined as in the previous definition. Then $\varepsilon$ is defined as the *linearized strain tensor*, and can be defined as $\varepsilon(u) = \frac{1}{2}(\nabla u + (\nabla u)^T)$ in the region $\Omega$ [4].

With all of those terms defined, we are now ready to define $\sigma$.

**Definition 2.9.** Let $\lambda$ and $\omega$ be defined as in Definition 2.5 and Definition 2.6, and $\varepsilon$ be as defined in 2.8. Then, the stress tensor $\sigma$ is defined as $\sigma = \lambda \text{trace}\varepsilon I + 2\omega\varepsilon$, where $\text{trace}(\varepsilon) = \sum_{i=1}^{n} \varepsilon_{ii}$.

Using the previous definition, we will use the following notation often in the thesis

**Definition 2.10.** Let $\sigma$ be defined as in Definition 2.9. Then, we will define $\sigma(u)$ as the stress tensor of $u$, and it will be computed as follows: $\sigma(u) = \lambda \text{trace}\varepsilon(u) + 2\omega\varepsilon(u)$, with $\varepsilon(u)$ defined as above.

This way, the notation $\sigma(u)$ will now be clear when it is used in the thesis.

Next I will define some other terms that will be used in this thesis. The first term occurs along the boundary (denoted $\partial\Omega$) of an object in motion

**Definition 2.11.** Let $u$ be the displacement of an object in motion along $\Omega$. The *normal displacement* of $u$ on $\partial\Omega$ is denoted $u_\nu$, and $u_\nu$ is defined as $u_\nu = u \cdot \nu$, with $\nu$ representing the unit outward normal of $u$ along the boundary $\partial\Omega$ [4].

Another term will arise while working along the boundary of $\Omega$, which we will define here:
**Definition 2.12.** Let \( u \) be the displacement of an object in motion along \( \Omega \). The **tangential displacement** of \( u \) on \( \partial \Omega \) is denoted \( u_\tau \), and \( u_\tau \) is defined as \( u_\tau = u - \nu u_\nu \), with \( \nu \) representing the unit outward normal of \( u \) along the boundary \( \partial \Omega \) [4].

### 2.2 Theorems

Here I will list the theorems that are needed in this paper. This first theorem is a major result for checking when an LCP has an unique solution, and is proven in [15].

**Theorem 2.1.** For any matrix \( M \), \( M \) is a P matrix if and only if the LCP\((M, q)\) has a unique solution for all vectors \( q \in \mathbb{R}^N \).

This theorem does then imply that if a matrix is not a P matrix, the LCP\((M, q)\) cannot have an unique solution for all vectors \( q \).

The next theorem gives another condition for showing a matrix is (or isn’t) a P matrix. This theorem is proven in [15].

**Theorem 2.2.** A matrix \( M \) is a P matrix if and only if all real eigenvalues of \( M \) and its principal submatrices are positive.
CHAPTER 3
SOLVING THE EQUATIONS OF MOTION

3.1 Modeling the Problem

3.1.1 Equations of Motion and Boundary Conditions

The standard equations of motion (with no boundary conditions) are given by:

\[ u_{tt} = \text{div}(\sigma) + f \] (3.1)

where \( \sigma = \lambda \text{trace}(\varepsilon)I + 2\omega \varepsilon \), and \( \varepsilon = \frac{1}{2}(\nabla u + (\nabla u)^T) \), with \( \lambda \) and \( \omega \) the Lame parameters as defined in Chapter 2, which vary depending on the material of the object being modeled. Here we use \( u \) to represent the displacement of the object in motion from the initial starting location. \( u_{tt} \) represents the 2\textsuperscript{nd} partial derivative of \( u \) with respect to time, and \( f \) represents a force acting on the object. Here, \( \text{div}(\sigma) \) and \( \text{trace}(\varepsilon) \) have the standard meaning as defined in [34]. Note also that \( \sigma \) is often referred to as the stress tensor, as defined in Chapter 1.

As we will be simulating the object in contact with a surface, we will need boundary conditions to represent this contact. The boundary conditions we will use are the standard Signorini contact conditions, as defined in [23]. The standard Signorini contact conditions are given by:

\[ 0 \leq N(x,t) \perp u(x,t) - g(x) \geq 0 \] (3.2)

for all \( t \) and all \( x \in d\Omega \)

These equations are what is commonly referred to as a linear complementary prob-
lems (LCP’s). Here $N(x,t)$ is the normal contact force, which is a measure of the force provided by the surface we are in contact with. $N(x,t)$ is positive only if the object is in contact, or in other words, if $u(x,t) - g(x)$ is 0, else $N(x,t)$ is 0 as the object is not in contact with the surface (note that both can be 0). Here we use $g(x)$ to stand for the normalized gap function of the object we are in contact with. Physically speaking, $g(x)$ is used to model the irregularities in the surface, as no surface is perfectly flat. However, note that a perfectly flat surface can be simulated (or assumed) by letting $g(x) = 0$. Here we are assuming the surface the object is in contact with is not in motion, so $g$ does not depend on time. For more on LCP’s and notation used I refer the reader back to Chapter 1 of this thesis.

A condition still needs to be added to address the friction the object is going to experience while in motion. In this thesis, I am going to use a simplified Signorini condition for the friction force $F$. For a object in motion on a smooth surface, the friction force satisfies:

$$|F| \leq \mu N$$

where $N$ is the normal contact force from (3.2) above.

Putting all of this together, I have so far have for the equations of motion the following, with $\Omega$ representing the region of the object that we are simulating, and $\partial \Omega_c$ representing the portion of the boundary of $\Omega$ in contact with the ground. All terms below are as they were defined in Chapter 2.
\[ u_{tt} = \text{div}\sigma(u) + f \]  \hspace{1cm} (3.4)

\[ 0 \leq N(x,t) \perp u(x,t) - g(x) \geq 0 \]  \hspace{1cm} (3.5)

\[ \sigma_v v = N \]  \hspace{1cm} (3.6)

\[ \sigma_t = F \]  \hspace{1cm} (3.7)

\[ |F| \leq \mu N \]  \hspace{1cm} (3.8)

where (3.4) is defined on the region \( \Omega \) and equations (3.5), (3.6), (3.7), and (3.8) are all defined on the region \( \partial \Omega_c \).

As it turns out, the above equations are great...if we are modeling a plate or another object where all parts of the object are always in contact with the ground. However, it rarely (if ever) happens that an object is always in contact with the ground, we need some boundary conditions for the parts of the boundary not in contact with the ground. As we want our object in motion to be free (or unrestrained), we want no clamping or other forces (other than gravity) applied to the object. Periodic boundary conditions will be used along the sides, which will mean that the equations for the right and left hand sides of the object will cancel each other out, so those equations will not be stated for that reason. Along the
top, the object will be free to move, so as we will be applying no traction force to the top (denoted \( \partial \Omega_t \)), we will have the following boundary condition:

\[
\sigma_t = 0
\]  

(3.9)

Combining 3.9 with the previous equations, gives us all the boundary conditions for the problem.

### 3.2 Discretizing the Problem

#### 3.2.1 Basic Setup

In order to solve these equations, I am going to discretize the equations in time and space. The time discretization will be done via an implicit midpoint method, and the space discretization will be done via the finite element method. For the time discretization, let \( h \) represent the length of each timestep, and let \( i \) represent the current timestep being modeled. In other words, \( u^i \) will represent the displacement of the object at time \( i \).

To start the time and space discretization, I will first of all need to define what space I am working in. To do this, we are first of all going to let \( w \in C^\infty \), with \( C^\infty \) as defined in Chapter 2. Take \( w \) and multiply all sides of equation 3.4, then integrate both sides of the equation over the region \( \Omega \).

Doing this will give the following equations:

\[
\int_\Omega u_{tt} \cdot w = \int_\Omega \text{div}\sigma(u) \cdot w + \int_\Omega f \cdot w
\]  

(3.10)

Now, use the formula \( \int_\Omega \text{div}\sigma(u) \cdot w = \int_{\partial\Omega} \sigma(u) \nu \cdot w - \int_\Omega \sigma(u) : \nabla w \) (from [4]) to integrate one of the terms above. Here, \( \partial\Omega \) is the boundary of \( \Omega \), and \( \tau \) stands for the unit
outward normal of the surface $\Omega$. Using this formula on equation 3.10 gives:

$$
\int_{\Omega} u_{tt} \cdot w = \int_{\partial\Omega_c} (\sigma(u)v) \cdot w - \int_{\Omega} \sigma(u) : \nabla w + \int_{\Omega} f \cdot w
$$

(3.11)

which is valid as the terms along the boundary portion not in contact (will) give us a 0 term and cancel out that integral.

The next step is to decompose the stress tensor into the normal and tangential components. Doing this gives the following formula (also found in [4]):

$$
\sigma v = \sigma_n v + \sigma_t
$$

(3.12)

where $\sigma_n$ is referred to as the normal component of $\sigma$ and $\sigma_t$ is referred to as the tangential component of $\sigma$, all calculated as stated in Chapter 2. Substitute 3.12 into 3.11 to get:

$$
\int_{\Omega} (u_{tt} \cdot w) = \int_{\partial\Omega_c} (\sigma_n u \cdot w + \sigma_t (u) \cdot w) - \int_{\Omega} (\sigma(u) : \nabla w) + \int_{\Omega} (f \cdot w)
$$

(3.13)

Now, using the boundary conditions from equations 3.6 and 3.7 on equation 3.13 gives us the following:

$$
\int_{\Omega} (u_{tt} \cdot w) = \int_{\partial\Omega_c} (N \cdot w + F \cdot w) - \int_{\Omega} (\sigma(u) : \nabla w) + \int_{\Omega} (f \cdot w)
$$

(3.14)

Using the equation in 3.14, it is now possible to discretize the problem in time, which is what will be done next.
3.2.2 Discretization in Time

Let \( u^i \), \( v^i \), \( N^i \), and \( F^i \) all represent the values of the displacement \( u \), velocity \( v \), normal contact force \( N \) and frictional contact force \( F \) at time \( i \). Also, let \( h \) represent the interval of time between steps \( i \) and \( i - 1 \). For simplicity, these experiments will use equally spaced timesteps, so \( h = \Delta i \) then by definition.

As \( v^i \) represents the velocity of the object at time \( i \), and \( u^i \) represents the displacement of the object at time \( i \), then by definition, \( v^i = u^i \), or differentiating both sides with respect to time gives \( v_i^i = u_i^i \). Substituting these equations into equation 3.14 and discretizing over time gives the following equations:

\[
\int_{\Omega} (v_i^{i+1} \cdot w) = \int_{\partial \Omega_c} (N_i^{i+1} \cdot w + F_i^{i+1} \cdot w) - \int_{\Omega} (\sigma(u_i^{i+1}) : \nabla w) + \int_{\Omega} (f \cdot w) \tag{3.15}
\]

We will use the derivative definition to approximate \( v_t \) at a given time by:

\[
v_i^{i+1} = \frac{v_i^{i+1} - v_i^i}{h} \tag{3.16}
\]

In addition, an implicit midpoint rule for solving for \( u_i^{i+1} \) will be used, so that gives the following equation for finding \( u_i^{i+1} \):

\[
u_i^{i+1} = \frac{h}{2} (v_i^{i+1} + v_i^i) + u_i^i \tag{3.17}
\]

Now, as \( u_i^{i+1} = v_i^{i+1} \), that means that we can use the time derivative to approximate \( v_i^{i+1} \) as well by:
\[
\frac{u^{i+1} - u^i}{h} = v^{i+1}
\]  

(3.18)

Plug equations 3.16, 3.17 and 3.18 into 3.15 to get the following equation:

\[
\int_{\Omega} \left( \frac{v^{i+1} - v^i}{h} \cdot w \right) = \int_{\partial \Omega} (N^{i+1} \cdot w) + \int_{\Omega} (F^{i+1} \cdot w) - \int_{\Omega} (\sigma \left( \frac{h}{4} (v^{i+1} + v^i) 
\right.
\left. + \frac{1}{2} u^i + \frac{1}{2} u^i) : \nabla w \right) + \int_{\Omega} (f \cdot w)
\]

(3.19)

Using the fact that the \( \text{div}(\sigma) \) and integration terms are linear, one can break up 3.19 to get all the \( v^{i+1} \) terms on the left hand side (after simplifying) as follows:

\[
\int_{\Omega} \left( \frac{v^{i+1}}{h} \cdot w \right) + \int_{\Omega} (\sigma \left( \frac{h}{4} v^{i+1} \right) : \nabla w) = \int_{\partial \Omega} (N^{i+1} \cdot w) + \int_{\Omega} (F^{i+1} \cdot w)
\]

\[
+ \int \left( \frac{v^i}{h} w \right) - \int_{\Omega} (\sigma \left( \frac{h}{4} (v^i + u^i) : \nabla w \right) + \int_{\Omega} (f \cdot w)
\]

(3.20)

Or, with some simplifying, equation 3.20 becomes:

\[
\int_{\Omega} (v^{i+1} \cdot w) + \frac{h^2}{4} \int_{\Omega} (\sigma(v^{i+1}) : \nabla w) = h \int_{\partial \Omega} (N^{i+1} \cdot w) + h \int_{\partial \Omega} (F^{i+1} \cdot w)
\]

\[
+ \int_{\Omega} (v^i \cdot w) - \frac{h^2}{4} \int_{\Omega} (\sigma(v^i) : \nabla w)
\]

\[
+ \frac{h^2}{2} \int (\sigma(u^i) : \nabla w) + h \int (f \cdot w)
\]

(3.21)

Which is the final time discretized equation.
3.2.3 Discretization in Space

Now, to solve this equation for $v^{i+1}$, it is necessary to discretize the object in space by using the finite element method. For the purposes of this paper, I will let $\Omega$ be a rectangle of equal length, so the object will be of length $a \times a$, where $a$ is a length that will be determined during the experiments. The object will be discretized into $n$ equally spaced nodes (the spacing of each node will be called $k$), where $n$ is a parameter that can be varied during the experiments. For purpose of reference, let $v_{q,r}$ to mean the node that is $k \times q$ distance from the leftmost boundary of the object, and $k \times r$ distance from the top boundary of the object. With this notation, $v_{2,3}$ would now mean the node that is $2k$ from the left of the object and $3k$ from the top of the object. This notation will be used on the function $w$ as well, with the same meaning. With this definition, $N_q$ and $F_q$ will refer to the normal contact force and the frictional contact force at the $q$th node from the left.

With this notation, we have the following for $u$ (similarly for $v$):

$$u = \sum_{q=1}^{n} \sum_{r=1}^{n} u_{q,r} \phi_{q,r}^u$$

where $\phi_{q,r}^u$ are the basis function over the region $\Omega$ (the superscript $u$ means basis function for $u$, for $v$ we will use $v$ superscript respectively, unless the meaning is obvious from the usage). For this problem, we are going to use linear basis functionals, and the region will be subdivided into triangles. The triangles will be aligned such that $\phi_{q,r}^u$, $\phi_{q,r+1}^u$, and $\phi_{q-1,r}^u$ are the three nodes of the triangles (where $q - 1$ is defined). $\phi_{q,r}^u$ will be such that $\phi_{q,r}^u$ will be 1 at the node $(q, r)$ and 0 at all other nodes. Also, $\phi$ is to be zero at all of the boundaries of the triangles. For ease of reference I will let $g = r + (q \times n)$, so will represent
node $\phi_{q,r}^u$ as $\phi_{g}^u$ going forward, and will use the variables $q$ and $r$ to represent other things from this point onwards (similarly for $u$ and $v$, as well as all of the basis functions).

With this change, the finite element space is now linear combination of linear functionals, or equivalently:

$$V = span\{\phi_i \mid 1 \leq i \leq n \times n\}$$

where we have $u, v, w \in V$ occurring. As $w$ is a test function, note that the previous time discretized equations must hold for all $w$ that we can select.

With the functions now represented as piecewise linear functionals, we now have to solve those integrals. Taking a look at the first integral, we have the following:

$$\sum_{g=1}^{n \times n} \int_{\Omega} v_{g}^{i+1} \phi_{j} \phi_{g}$$

for $1 \leq j \leq n \times n$. Or equivalently:

$$\sum_{g=1}^{n \times n} v_{g}^{i+1} \int_{\Omega} \phi_{j} \phi_{g}$$

for $1 \leq j \leq n \times n$. Represent the matrix $M$ as follows then:

$$M_{g,j} = \int_{\Omega} \phi_{j} \phi_{g}$$

for $1 \leq j \leq n \times n$. $M$ here is typically referred to as the mass matrix of the problem. Similarly, we can do the same thing with the $\sigma$ term, and if we let $K_{g,j}$ be the following:
\[ K_{g,j} = \int_{\Omega} \sigma(\phi_j) : \nabla \phi_g \]

for \( 1 \leq j \leq n \times n \). \( K \) here is typically referred to as the stress matrix of the problem.

With all of these substitutions made, equation 3.21 becomes as follows:

\[
Mv^{i+1} + \frac{h^2}{4} K v^{i+1} = Mv^i - \frac{h^2}{4} K v^i + \frac{h^2}{2} K + S \tag{3.22}
\]

where \( S \) stands for the boundary terms that have not been properly space discretized yet (as that step is next).

To solve the remaining integrals, use piecewise constant functionals along the boundary. As the choice of the constant makes little difference, we will for simplicity use \( \phi_g = 1 \) along the interval from node \( g \) to \( g + 1 \), and 0 elsewhere. With this substitution, now let \( C_q N_{q}^{i+1} = \int_{\partial \Omega} N_{q}^{i+1} \phi_q \). Similarly, let \( D_q F_{q}^{i+1} = \int_{\partial \Omega} F_{q}^{i+1} \phi_q \). To solve the integral \( \int_{\Omega} f \cdot \phi_g \), use the approximation \( \int_{\Omega} f(g) \cdot \phi_g = f(g) \int \phi_g \), and will then let \( E_g f(g) = f(g) \int \phi_g \). With all of these substitutions made into 3.22 and some simplifying, one gets:

\[
(M + \frac{h^2}{4} K)v^{i+1} = (M - \frac{h^2}{4} K)v^i + hCN_{i+1}^i + hDF_{i+1}^i + \frac{h^2}{2} Ku^i + hE f \tag{3.23}
\]

With all of the space and time discretization complete, the equations to solve are now as follows:

\[
(M + \frac{h^2}{4} K)v^{i+1} = (M - \frac{h^2}{4} K)v^i + hCN_{i+1}^i + hDF_{i+1}^i + \frac{h^2}{2} Ku^i + hE f \tag{3.24}
\]
\[ u^{i+1} = \frac{h}{2}(v^{i+1} + v^i) + u^i \]  
\[ (3.25) \]

\[ 0 \leq C^T(u^{i+1} - g) \perp N^{i+1} \geq 0 \]  
\[ (3.26) \]

\[ |F^{i+1}| \leq \mu N^{i+1} \]  
\[ (3.27) \]

Notice that the LCP condition changed to include a multiplication by the boundary matrix \( C \). This is because without that term, energy tends to increase in the discretized case, which is undesirable (see [33] for more details).

### 3.2.4 Problems in solving the equations for \( N \) and \( F \)

In order to solve the above equations for \( v \) and \( u \) (note that for ease of notation, the node numbers are omitted in this section), one need to know what the values of \( N \) and \( F \) are. In order to do this, one will need to solve the LCP given by:

\[ 0 \leq C^T(u^{i+1} - g) \perp N^{i+1} \geq 0 \]  
\[ (3.28) \]

However, solving this LCP is far from straightforward. This is because at each node one has to compute the values of \( N, F, u \) and \( v \) at that node. But, there is equation (3.27) to consider, which means that the value of \( F \) at the node depends on the value of \( N \) at that node. And, by equation (3.25), the value of \( u \) depends on the value of \( N \) and \( F \) at that node, but per equation (3.28), the value of \( N \) depends on whether or not the solution for \( u \) has \( u \) penetrating the ground! The solution for \( N, F, u \) and \( v \) all depend on each other.
at that node. Due to this reason, getting the values for $N$, $F$, $u$ and $v$ is far from trivial. One approach would be to assume there is no ground first, solve for $u$ and $v$, and use those solutions for $N$ and $F$. This has the problem of getting values of $N$ that are too large, and can cause the object to jump off the ground in response. Another approach is to assume the value of $N$ and $F$ at each node does not change much over each timestep, and use those values to solve for $u$ and $v$, then get $N$ and $F$. This does require $h$ be really small though to be reasonably accurate though.

There are two different approaches to solving for $N$ and $F$ that will be used in this paper. In Section 3.3 I am going to use a Jacobi like approach to solve for $N$, and use that solution for $F$. I will, with some algebra, effectively eliminate the need for the current values of $u$ and $v$ from the equations for $N$ and $F$, and will instead only need the values of $u$ and $v$ at the previous timestep to solve. Once $N$ and $F$ are found, I will then solve for $u$ and $v$ for that timestep, and then with that data I will solve for $N$ and $F$ for the next timestep, continuing as long as I wish to simulate. This approach is far from perfect though, which is what most of the rest of this thesis will be devoted to discussing. In Chapter 5, a new method using a different equation than (3.27) on the boundary will be proposed, and I will then show that energy is conserved for the revised problem.

3.3 The Jacobi Method Setup

3.3.1 Solving for $N$

In an attempt to solve for $N$ and $F$, I am going to solve the LCP into a form that only involves $u$ and $v$ at the previous timesteps, as well as the current value of $N$. $N$ will
still have an $F$ term in it, which will be dealt with later. To begin solving for $N$, we need to solve the LCP given by:

$$0 \leq N^{i+1} \perp C^T (u^{i+1} - g) \geq 0$$  \hspace{1cm} (3.29)$$

To solve for this, consider only the RHS (Right Hand Side) given by:

$$C^T (u^{i+1} - g)$$  \hspace{1cm} (3.30)$$

To solve the RHS, one need to know the term $u^{i+1}$, which has not been solved for yet in our method. However, note that there is an equation for $u^{i+1}$, given by (3.25). Insert that into (3.30), we get the following:

$$C^T \left( h^2 (v^{i+1} + v^i) + u^i - g \right)$$  \hspace{1cm} (3.31)$$

As 3.31 has the values of $v$ and $u$ at the previous timesteps, the only unknown now is the $v^{i+1}$ term. However, by (3.24), an equation for $v^{i+1}$ can be found. Take that equation, and substitute into equation (3.31) to get:

$$C^T \left( \frac{h}{2} (v^{i+1} + v^i) + u^i - g \right)$$  \hspace{1cm} (3.32)$$

where $A^{-1} = (M + \frac{h^2}{4} K)^{-1}$. To solve this for $N^{i+1}$, look again at equation (3.29). If the object is not on the ground, then by definition $u^{i+1} - g$ will be positive, making the RHS positive, and the normal contact force will be 0 in that case. Therefore, the case that one
needs to be concerned with is when the object is on the ground, or when the RHS of the equation is 0. Using that fact, one now needs to solve the following equation:

\[
C^T \left( \frac{h}{2} \left( A^{-1} \left[ (M - \frac{h^2}{4}K) v^i + hCN^{i+1} + hDF^{i+1} + \frac{h^2}{2} Ku^i + hEf \right] + v^i \right) + u^i - g \right) = 0 \quad (3.33)
\]

Re-arranging all the terms so that only the \(N^{i+1}\) term appears on the left side gives:

\[
\frac{h^2}{2} C^T A^{-1} CN^{i+1} = -\frac{h}{2} C^T A^{-1} \left[ (M - \frac{h^2}{4}K) v^i + hDF^{i+1} + \frac{h^2}{2} Ku^i + hEf \right] - C^T v^i - C^T (u^i - g) \quad (3.34)
\]

While (3.34) does indeed give a form of \(N^{i+1}\) that no longer depends on anything from the current timestep except for \(F^{i+1}\), solving this equation is far from trivial. This is because computing \(A^{-1}\) is not normally practical, due to the size of the matrix. In a typical run with 25 nodes in both the \(x\) and \(y\) direction of the object, the matrix \(A\) is of size \(1250 \times 1250\) (\(2 \times 25 \times 25\) in each direction). If an experiment with 100 nodes in the \(x\) and \(y\) direction were to be conducted, the matrix \(A\) would now be of size \(2 \times 100 \times 100\) in each direction, or in other words, matrix \(A\) would be a \(20000 \times 20000\) matrix. Attempts at computing the matrix inverse of a \(20000 \times 20000\) matrix directly resulted in an out of memory error on a computer with 4 GB of memory using Windows Vista and Matlab version 2007a. Attempts at computing the matrix inverse directly of the \(1250 \times 1250\) matrix on a computer with 4 GB of memory using Windows Vista and Matlab version 2007a took on average 5 minutes. Thankfully, the matrix inverse can be computed once at the start of the simulation and saved, but that is still not practical for quick runs of the program.
To get around this problem, a matrix splitting will be employed on the matrix \( A^{-1} \).

To start the matrix splitting, re-write the matrix as follows:

\[
A^{-1} = D + \bar{D}
\]

Where \( D \) stands for the diagonal element of \( A^{-1} \), and \( \bar{D} \) stands for all the other (non-diagonal) terms in the matrix. With this substitution into (3.34), the equation becomes as follows:

\[
\frac{h^2}{2} C^T D C N^{i+1} = -\frac{h^2}{2} C^T \bar{D} C N^{i+1} - C^T v^i - C^T (u^i - g)
\]

\[
-\frac{h}{2} C^T A^{-1} [((M - \frac{h^2}{4} K)v^i + hDF^{i+1} + \frac{h^2}{2} K u^i + h f)]
\]

Now, \( \bar{D} \) can be re-written as follows:

\[
\bar{D} = A^{-1} - D
\]

Substituting (3.36) into (3.35), gives the following:

\[
\frac{h^2}{2} C^T D C N^{i+1} = -\frac{h^2}{2} C^T A^{-1} C N^{i+1} + \frac{h^2}{2} C^T D C N^{i+1} + R
\]

where the matrix \( R \) stands for all the terms without \( N^{i+1} \) in them.

Now that the matrix \( A^{-1} \) has been split into the diagonal and non-diagonal terms, a Jacobi method can be employed to solve for \( N \). For full details on how the Jacobi method works and its convergence, see [4].
To use the Jacobi method on (3.37), the diagonal element of $A^{-1}$ needs to be computed. This can be done by employing the Conjugate Gradient Method (CG Method) method [30] to solve the equation $Ax = Ce_{dim\times n\times n}$, where $e_{dim\times n\times n}$ is the elementary basis element $dim \times n \times n$, where $dim$ is the number of dimensions of the object in motion (in this paper $dim$ will be 2 unless noted otherwise). Once this is done, notice that we will have $x = A^{-1}C$ for one value (specifically, the $n$ by $n$th position in the vector $x$). Then, taking $C^T \cdot x$ will yield us the diagonal element of $C^T A^{-1} C$ at the point $(dim \times n \times n, dim \times n \times n)$. Tests against the real diagonal value of $C^T A^{-1} C$ on experiments with 5 and 25 nodes showed this approximation was accurate to 15 significant digits, so I am reasonably confident that this approximation for the diagonal element will work for solving the problem.

Using similar reasoning, one can compute any of the other terms where $A^{-1}$ exists, by taking the equation $Ax = S$, where $S$ is whatever we are multiplying by the term $A^{-1}$, provided $S$ is a $dim \times n \times n$ vector. However, the other terms where $A^{-1}$ needs to be multiplied by in this problem are of that size, so one can solve for $x$ using a Conjugate Gradient method and then multiplying the solution $x$ by the terms that were going to be multiplied to the $A^{-1}$ term. By doing this, one can avoid having to compute any terms of the matrix $A^{-1}$ directly.

Now that the $A^{-1}$ term has been addressed, the problem will split the equation into a Jacobi problem. Let $r$ stand for the $r$th iteration of the Jacobi method. With that, the problem that needs to be solved is as follows:
\[
\frac{h^2}{2} C^T DCN^{i+1, r} = -\frac{h^2}{2} C^T A^{-1} CN^{i+1, r-1} + \frac{h^2}{2} C^T DCN^{i+1, r-1} + R
\] (3.38)

As the Jacobi method is guaranteed to converge as long as the matrix \( C^T DC \) is diagonally dominant [4], the starting value of \( N \) used in the iterations are irrelevant. However, the starting value of \( N = 0 \) is usually used to see if the case of no normal contact force (the trivial case) holds first, then if the Jacobi method finds that this is not accurate, it will adjust \( N \) based on (3.38). The method will run until the iterations are within a certain tolerance of each other, as specified in the program (usually \( .00001 \)).

### 3.3.2 Solving for F

In Section 3.1 the LCP problem (3.29) was solved for \( N \) in terms of values of \( u \) and \( v \) from the previous timestep. However, in equation (3.38), the \( R \) term does have one equation that has the term \( F^{i+1} \) in it. As we also have the relation (3.27), it seems that as one adjusts the values of \( N^{i+1} \), values of \( F^{i+1} \) should be updated as well. However, this is not practically possible. The Jacobi method can converge to a point as long as that point is not moving. If the values of \( F^{i+1} \) are updated with each iteration of the Jacobi method to reflect the changes made to \( N^{i+1} \), the point of convergence of the method will change with each step! This can result in the method constantly looping back and forth between two different values. Experiments ran where the value of \( F^{i+1} \) were updated during each iteration of the Jacobi method resulted in the method either diverging off to infinity, or alternating between two (or more) values for \( N \) and \( F \), even if those values were pretty far apart from each other. Attempts to fix alternation by taking the average of the alternating
values and continuing with them failed to produce any improvement, and still resulted in (more) alternation or divergence.

Because of this, one cannot update the values of $F^{i+1}$ during each step of the Jacobi method. However, if an object is in motion, it is safe to assume that the friction force being exerted on the object will not change significantly during a timestep, especially if the time interval used is small. Using that assumption, equation (3.34) becomes

$$\frac{h^2}{2}C^T A^{-1} C N^{i+1} = -\frac{h}{2} C^T A^{-1} [(M - \frac{h^2}{4} K) v^i + hD F^i + \frac{h^2}{2} K u^i + hE f] - C^T v^i - C^T (u^i - g)$$

(3.39)

All the other steps of the previous section follow as stated, with $F^{i+1}$ now replaced by $F^i$, or the friction value from the previous timestep. This results in the Jacobi method having a point of convergence that is not moving, and the method can converge now to the solution for $N^{i+1}$. Once $N^{i+1}$ has been found, then equation (3.27) can be used to find $F^{i+1}$. This method is far from perfect, as significant errors can result in the answer for $N^{i+1}$ if the timesteps are not small enough. However, this does not prove a significant problem for the simulations used in this thesis, but it still is something to keep in mind.

Now that $N^{i+1}$ no longer depends on the values of $F^{i+1}$, the only equation that determines the value of $F^{i+1}$ is given by (3.27). However, this means that $F^{i+1}$ can end up taking on any value between $-\mu N^{i+1}$ and $\mu N$ and be valid for that equation as written. A better way to write (3.27) is to break it into three cases:

$$F^{i+1} = -\mu N^{i+1}$$

(3.40)
\[ F^{i+1} = \mu N^{i+1} \quad (3.41) \]

\[ |F^{i+1}| < \mu N^{i+1} \quad (3.42) \]

Where (3.40) occurs when forward progress has been made, (3.41) occurs where backwards progress has been made (or in other words, the object is sliding backwards in our chosen coordinate system), and (3.42) occurs where the force exerted on the object is not enough to offset the current friction force. Note that the third case only occurs when the object is stationary, assuming no continuous external forces are exerted. For the purpose of this paper I am only going to look at objects that start in motion and are sliding forward with respect to our chosen coordinate system, so (3.40) is the only equation that will be considered for this paper. This will result in some inaccuracies on the final answer for how far the object has slid on the surface as the object will not stop as soon as it should have. However, as the purpose of this paper is to find out when the instabilities occur as well as show some things that cause them, this inaccuracy in the final location of the object is considered acceptable. For more detail on the problems and difficulties that can arise in solving the friction equations, see [32, 5, 33, 23], just to name a few.

### 3.4 Implementation Decisions in this Model Clarified

While implementing the Jacobi method, certain implementation decisions had to be made. This section is dedicated to explaining why those decisions were made.
3.4.1 Why \( \mu = 6.33 \)?

The reader will notice that in the results chapter (Chapter 4) that the case friction coefficient \( (\mu) = 6.33 \) is almost always run, despite the fact that almost no other cases of \( \mu \) between 1 and 6.33 are usually considered. This is because based on some private communication with Theodore Wendt, a certain range of friction coefficients were found to always have an instability occur for elastic bodies in motion under coulomb friction, \( \mu = 6.33 \) happens to be the midpoint of that range for the Poissions ratio that was chosen. Therefore, the decision to include the \( \mu = 6.33 \) case was made, as based on those preliminary results an instability should always occur in the 2 dimensional case for an object in motion as simulated by equations (3.1).

3.4.2 Why \( E = 210 \)?

The reader may also be wondering why the value for \( E \) was chosen as it was. \( E \) is defined as Young’s Modulus [26], and it is known as the amount of pressure you need to exert per square inch to double the length of the object. For Steel that number has been found to be 210GPa. Experiments were ran with \( E = 1 \) as that made the matrix \( A \) and \( B \) in the above work have a much better condition number (see reference for why that can be important), but it was found that with \( E = 1 \) the object had about the consistency of Jell-O when sliding along the surface. Nodes that started out well above the ground ended up touching the ground, and needing a normal contact force computed for them as well as the nodes that started out on the ground. This was fixed by changing the standard LCP condition of \( C'u - gap \) to \( C'(u - gap) \) (which the reader will notice is how the LCP in
3.29 was defined), as long as the gap function was defined for all nodes. Once $E = 210$ was implemented, none of the top nodes ever seemed to sink down and hit the ground, but the $C'(u - gap)$ term was kept just the same just in case it ever happened again, or if simulations modeling Jell-O (or other really soft objects) are ever desired.

### 3.4.3 Why Results Can Vary Across Computers

One other item of importance needs to be mentioned before covering the results. If these same experiments are re-ran on a different computer or with different versions of Matlab, you will get slightly different locations for when the first instability occurs. This is due to differences in how the computer stores information and also differences in how matlab stores information. This is a well documented issue with computers and data storage, and I invite the interested reader to read [12, 13] to learn more about this subject. Experiments were run on both version 2007a and version 14 of Matlab (both student versions), and a difference of as much as 1000 timesteps would occur on when the first instability occurred. For the reader who wishes to re-run my experiments listed in the next chapter, this difference will need to be kept in mind. No difference on whether an instability will occur seemed to exist though on any experiments that were ran on both computers.
CHAPTER 4
RESULTS

In this chapter, the results from the discretized equations are given. In all cases, the object was held to .25 meters wide and high, except where noted. All results were ran on either a Pentium D 2.8 GHZ Dual Core with 4 GB memory, or an AMD 64 x2 1.9 GHZ dual core with 2 GB memory. Most results were ran on Matlab Student version 2007a, though some of the older experiments were ran on Matlab Student version 14. However, each sequence of experiments were ran on the same computer and same version of matlab to avoid discrepancies in the results.

4.1 Results from the Jacobi Method

4.1.1 Parameters for the Experiments

All results are given in the appropriate section. In each series of results, I noted the number nodes in the $x$ and $y$ direction the object had (for simplicity a rectangular object was used), the $h$ value used, the $k$ value (grid spacing), what Poisson’s ratio was for the object, what the friction coefficient was, and what the starting values for $u$ and $v$ were as well. All results were done with the initial values of $u$ and $v$ being uniform (except when noted). The object was always assumed to be .25 units wide and high in all cases, so $k$ was adjusted based on the number of nodes desired (except when noted). Also due to how slow the algorithm was, most results were done with only 5 or 25 nodes used. On average, the 5 node case could run a little over 3.5 million timesteps per day, and the 25 node case could run about 750000 timesteps per day. A 100 node case took about a whole day to run 5000
timesteps. This unfortunately made running cases with high number of nodes unpractical due to the shear amount of time needed. Also, the time to run an experiment was not linear. A 5 node experiment where 7.5 million timesteps were desired would take 2 days, but an experiment where 15 million timesteps were desired took over 10 days. This is due to matlab having to reallocate the memory as the experiment progresses, which significantly slows down results after the experiment has been running for a long time.

In all cases, the object was pushed along in the $x$ direction until an instability was found. See Chapter 1 for a reminder of what definition of instability I am using here, as the definition varies by researcher. Also, as making sure that energy remains conserved is important, I also noted when and if energy ever increases as the object is sliding along the surface, and by how much. For details on what happens once an instability is found, and also what is occurring before and after one has been discovered, I refer the reader to later on in this chapter

On the enclosed CD is all the code as well as the Excel spreadsheet where all of the data was recorded. Note that in the upcoming tables and graphs some data is omitted, that was done in cases where some parameters remained the same on the entire table. For instance, if the initial velocity was $30m/sec$ for all experiments in the table, the initial velocity is not noted on the table then. This is to preserve space and to keep only the important data in the thesis. For the interested reader who wishes to see the data in more detail, I refer them to the enclosed CD. All the data is in an Excel spreadsheet titled Instability Data. Note that a printout of the sheet is well over 10 pages long, which is another reason only parts of the data is reported.
4.1.2 Frictionless Experiments

Table 4.1: Frictionless Experiments, 25 nodes

<table>
<thead>
<tr>
<th>h</th>
<th># of timesteps</th>
<th>Final x value</th>
<th>Final y value</th>
<th>Total time simulated</th>
</tr>
</thead>
<tbody>
<tr>
<td>.0005</td>
<td>150000</td>
<td>2.28 \times 10^3</td>
<td>0</td>
<td>75 seconds</td>
</tr>
<tr>
<td>.0001</td>
<td>750000</td>
<td>2.28 \times 10^3</td>
<td>0</td>
<td>75 seconds</td>
</tr>
<tr>
<td>.00005</td>
<td>1500000</td>
<td>2.28 \times 10^3</td>
<td>0</td>
<td>75 seconds</td>
</tr>
<tr>
<td>.00003</td>
<td>3000000</td>
<td>2.73 \times 10^3</td>
<td>0</td>
<td>90 seconds</td>
</tr>
</tbody>
</table>

First of all, the object was tested by sliding it along a frictionless surface ($\mu = 0$). In all these cases the object keeps on sliding along and no instability ever developed, see Table 1 for more details of these experiments. What is interesting to note is that in all frictionless cases, the energy increases on the 1st and 2nd time step. Strangely enough, the object never went unstable despite the increase in energy. See Figure 1 to see the energy gain per timestep over an object sliding for 5000 timesteps with no friction and 25 nodes. A quick observation of the graph reveals that the energy is continually increasing and decreasing throughout the simulation. However, it seems the gain of energy is not significant, as no problems or instabilities ever arise due to these increases in energy.

4.1.3 No Initial Velocity Experiments

The next set of experiments tested the object by seeing if it remained motionless if no velocity was applied to it. In this case, a friction coefficient was provided for some
experiments, as it should prove irrelevant to the final location of the object. If the object went unstable in this case, we could conclude that the friction coefficient was the cause of instabilities, as that would be the only thing changed between experiments. The results of these experiments are listed in Table 2. As you can see the object never went unstable, but it did drift forward a little still. This is probably due to the fact that in the experiments we elected to ignore static friction, so the program is treating the object as if it is moving forward, and as a result it is moving forward some, just very slowly.
4.1.4 Variance of Initial Boundary Conditions

The next set of experiments attempted to vary the boundary conditions (starting displacement and starting velocity) to see how changing them affected the occurrence of instabilities. In all cases, the object was slid along a surface given a constant velocity of $v = 30$ in the $x$ direction, a starting displacement of $u = 30$ in the $x$-direction ($u$ and $v$ were $0$ in the $y$-direction), $h = 5 \times 10^{-5}$, $k = .01$, $nodes = 25$, Poisson’s ratio of .3, A Young’s Modulus of $E = 210$, and a friction coefficient of $\mu = .5$. What was found is that the object went unstable after 11273 timesteps. To see what influenced this, $v$ was first varied while $u$ was held constant, then another set of experiments were ran with $u$ varied while $v$ was held
constant. Figure 3 shows how the initial starting displacement affects the timestep of first instability, and Figure 4 shows how the initial starting velocity affects the timestep of first instability.

What was discovered in these experiments is that varying the starting displacement had little to no effect on how long the experiment would run before an instability occurred, but if little to no starting velocity was provided, the object never went unstable during the 100000 timesteps that the experiment was run. This was checked again with an initial velocity of 1 m/sec provided to the entire left side of the object and during the 500000 timesteps that the object was simulated on instability was found. Also it is worth noting
that for small initial velocities ($v < 10$) the object appears to go unstable pretty quickly, then at $v = 10$ it jumps up to needing 12195 timesteps, and appears to go down from that number until $v = 200$, where it starts to increase again. It does appear that the starting velocity has some effect on when an instability appears based on these results. Also, notice that the higher the starting velocity is, the longer it takes for the first increase in energy to occur. The graph of first increase of energy can be found in Figure 4.

Figure 4.4: First Energy Increase with Velocity Varied

More experiments were conducted to see if varying the starting velocity had any
effect on when instabilities occurred and when the first occurrence of energy increase occurred, but different timesteps were used. This was to see if the results would occur again under different values for the other terms in the experiments, and then the starting velocity was varied on those cases. Those results can be seen in Figures 5 and 6. As the reader can see, the results prove pretty consistent even if some of the other terms are changed as well.

Figure 4.5: Variance of Starting Velocity, h = .0001
4.1.5 Variance of Friction Coefficient

For the next series of experiments, \( v \) and \( u \) were held at a constant 30 in the \( x \)-direction, 0 in the \( y \)-direction, Poisson’s was held constant at .3, the number of nodes was held steady at 25, \( k = .01 \), \( h = 5 \times 10^{-5} \), but the friction coefficient was varied. What was discovered was that as the friction coefficient got closer to 0, the object took longer to go unstable, but as the friction coefficient was increased, the object went unstable sooner. This is shown in Figure 7. Also of note is that the total energy of the system increased between timesteps before an instability occurred in all cases. Based upon these results it seems that the higher the friction coefficient is, the sooner the object will start to go unstable.
Figure 8 shows how much time elapses before the first energy increase. What is interesting to note is that it is not a continually decreasing line like the graph showing how varying the friction coefficient affected timesteps was. Instead, it starts small, increases, then decreases. It seems that a really low friction coefficient ($\mu \leq .15$) energy will increase between timesteps as early as the 2nd timestep, yet it was in these cases that the experiment took the longest total number of timesteps to go unstable.
4.1.6 Variance of Timesteps and Friction Coefficient

After finishing those experiments, new experiments were ran where \( h \) was varied to see if smaller (or bigger) timesteps would change when an instability occurred. After picking a new value for \( h \), the friction coefficient (\( \mu \)) was varied again to see if the same pattern would occur as it did when \( h = .5 \times 10^{-5} \). Experiments were run with \( h = 3 \times 10^{-5} \), \( h = 1 \times 10^{-5} \), \( h = 1 \times 10^{-4} \), \( h = 5 \times 10^{-4} \), \( h = 5 \times 10^{-6} \). Experiments were also attempted with \( h = .001 \), but it was found that for that value of \( h \) the Jacobi iterations failed to converge. Also, experiments were planned for \( h = 5 \times 10^{-7} \), but after seeing how many timesteps were needed in the \( h = 5 \times 10^{-6} \) case as well as how many timesteps (on average) the computer
could run per day it was concluded that about 100 days would be needed to run that case thoroughly, so the experiments with \( h \leq 5 \times 10^{-7} \) were ran for the case of \( \mu = .5 \) only for comparison sake.

All the experiments with \( h \) varied are included in the spreadsheet on the enclosed CD. Figure 9 shows that for \( \mu = .5 \), the total time it takes for an instability to occur does improve as \( h \) decreases, though after a while little to no gain is experienced by decreasing \( h \), and in some cases decreasing \( h \) actually causes the instability to occur sooner. Figure 10
Figure 4.10: Zoomed In View of Figure 9

A casual glance at the data on the enclosed spreadsheet shows that this does not seem to occur for only $\mu = .5$. Table 2 in Section 4.4 shows all of the various friction coefficients and how long it takes in terms of total time elapsed before the experiment goes unstable, and in most of the cases there is some improvement then a slight decrease at various points in the chart. There were only a couple exceptions, and chances are if the experiments for $h = 1 \times 10^{-6}$ or smaller values had been run those cases would have started to decrease as well.
4.1.7 Cases with Different Number of Nodes

To check things out farther, new experiments were run with 5 nodes in each direc-
tion. For these new experiments, Poisson’s was kept constant (.3), \( v = 30 \) in the x direction, \( u = 30 \) in the y direction, \( k = .05 \) (so that the object was kept a constant .25 in each direction), and \( h \) was kept constant while the friction coefficient was varied. This was done for \( h = 5 \times 10^{-5}, 3 \times 10^{-5}, 1 \times 10^{-5}, 5 \times 10^{-6}, 3 \times 10^{-6}, \) and \( 1 \times 10^{-6} \). What was found was that for \( h = 1 \times 10^{-5} \), the object reached a steady state for a friction coefficient of \( \mu \leq .25 \), no instability occurred in those cases. As \( h \) got smaller, the cases where no instability occurred increased (see graph), such that when \( h = 1 \times 10^{-6} \), no instability occurred for \( \mu \leq 1 \). This is also shown in Table 1 in Section 4.4, where the total time elapsed before an instability was discovered is listed as well as the friction coefficient given and the timestep given. The cases where no known instability was discovered are marked with N/A in the table.

It is entirely possible an experiment marked N/A may eventually go unstable, just that no instability was discovered during my simulations. On average the simulations were ran for \( 50 - 75 \) simulated seconds (which took on average \( 5 - 10 \) days an experiment!) to check for an instability.

New experiments were run with 10 nodes and \( k = .025 \), \( v = 30 \) in the x-direction, \( u = 30 \) in the y-direction, Poisson’s = .3, with \( h \) and \( \mu \) varying, but nothing was found in those cases that was not already found in the 25 node cases. All of the 10 node cases can be seen in Table 2 in Section 4.4, where the total time simulated before an instability is plotted along with the friction coefficients and timesteps used in the experiments. In addition, for 50 nodes, \( k = .005 \), same \( v \), \( u \) and Poisson’s as in the previous cases, nothing new was
learned. Due to the sheer length of time needed to simulate these problems, cases with more than 50 nodes were not thoroughly examined. All of the results with 50 nodes can be seen in the enclosed CD on the excel file instability notes. They are not added as a table as nothing significantly new was learned in these experiments that was already not learned with the 25 node experiments, but the file is available for the interested reader to reference.

4.1.8 Variance of Node Spacing

Figure 4.11: Variance of Number of Nodes

A couple of last cases were considered that need to be mentioned. In one experi-
ment, $k$ was varied, while Poisson’s was held constant (.3), $v = 30$ in the $x$-direction, $u = 30$ in the $y$-direction, $h = 5 \times 10^{-5}$, and $\mu = .5$. As the object is to be kept at .25 in all directions, the number of nodes had to change as $k$ changed so that $k \times \text{number of nodes} = 25$. What was found is that as the number of nodes increases (due to $k$ decreasing) an instability occurs much sooner than if less nodes are used. In the case where number of nodes were 250 and $k = .001$, an instability occurred as soon as 442 timesteps, much sooner than it did for 25 nodes and $k = .01$ (11273 timesteps). Note this is not universal as some improvement did occur for 500 nodes, $k = .0005$, but even there the gain was really slight (627 timesteps vs 447 timesteps).

Figure 4.12: Variance of Number of Nodes, Fixed Object Length
Another case was considered where just the number of nodes increased, but \( k \) did not (so now the object is getting longer as the number of nodes increases). The rest of the data was kept the same as the previous experiments. What was found is that when the number of nodes increases (and hence the length of the object) it takes longer for an instability to occur. This makes sense though, as since the object is longer now, it is harder for a particular region of the object to lift off of the ground. Also, it should be harder to get enough force to lift the entire object off the ground too, which is exactly what these results seem to suggest.

4.1.9 Removal of Stress and Strain Terms

In the last set of experiments, the grad div term of the equations of motion was commented out. This has the effect of ignoring all stress and strain in the object while simulating it in motion. This does ignore certain forces at work though, so the results will not be physically accurate. Once the term was commented out of the code, some of the previous experiments were re-ran to see what happens if the stress and strain terms were ignored. All these results are in Table 5 in Section 4.4. What was discovered is that without the stress and strain terms, the object does not go unstable, but instead slides along to a steady state. This seems to occur no matter what the friction coefficient is.

The reader will notice that for each friction coefficient the experiment was ended at totally different timesteps. This is because once it was noticed that the object was not moving anymore in the x-direction, the experiment was ended. Based on the results of simulating an object given no initial velocity, it seems reasonable to assume that an object
will not go unstable once it is no longer in motion and has become stationary.

### 4.2 Interpretation of the Results of the Jacobi Method

The results of the Jacobi method seem to suggest a couple of significant findings. These are summarized below.

#### 4.2.1 Significance of Smaller Timesteps

First of all, I refer the reader back to Figure 9 here, where I mapped out total time simulated before instability vs timesteps, for a friction coefficient of 5 and 25 nodes. What is interesting is that decreasing $h$ to a smaller value does allow one to run the experiment longer, but only for a while. For $h = 1 \times 10^{-5}$, the experiment actually goes unstable sooner (in terms of total time, not timesteps) than it did for $3 \times 10^{-5}$. In addition, decreasing $h$ significantly increases the total time needed to run an experiment to simulate the same about of total time. To give an example, for one experiment with 25 nodes and $\mu = .5$, $h = 5 \times 10^{-6}$ was used as the value of the timesteps. In that experiment, the object went unstable after 181178 timesteps, and the total time simulated was less than 1 second. However, on a Pentium 2.8 GHZ with 4 GB of memory, that simulation took over 3 hours. However, a simulation with $h = 3 \times 10^{-5}$ went unstable after 31826 timesteps, with a total time simulated of a little under a second. This time the experiment took a little over an hour, which is much better, and the final result was not that much more different from the case with a smaller timestep.

This seems to not be exclusive for a friction coefficient of .5. Looking at the results for different friction coefficients shows that decreasing $h$ gives diminishing returns over
time. And, for values of $h < 1 \times 10^{-5}$, the gain to total time one can simulate before an instability seems to be insignificant, and in some cases decreases too. Based on these results I would say that running experiments with $h < 1 \times 10^{-5}$ is not worth it as the extra time to compute the extra timesteps does not seem worth it for what little (if any) gain one gets in how long one can simulate the experiment before an instability is detected, at least for rectangular objects with 10 or 25 nodes. Preliminary results seemed to suggest the same pattern occurs with 50 nodes as well, though further investigation is needed to be sure if this is indeed occurring in those cases.

4.2.2 Significance of Starting Boundary Conditions

Varying the starting velocity and displacement did seem to influence the results, just in different ways. The starting values of $u$ (displacement) in the $x$ direction seemed to have no effect on when instabilities occurred. This does have significant implications because this means if one stopped an experiment, and if the experiment has all the $x$-values the same when stopped, those same $x$-values can be provided to a new experiment, and one can avoid an instability this way (or at least delay its occurrence). This also means that in theory one can never have an instability occur by the correct use of starting and stopping the experiments. However, this does assume all the $x$-values are the same, which unfortunately is not true, and is something I will discuss more in Section 4.3.

Varying the velocity also influenced the occurrence of instabilities, just in another way. For really small starting velocities ($0$ and $1m/sec$) no instability was ever detected for $h = 1 \times 10^{-5}$. This does imply that once an object goes stationary, it will never go unstable.
Also, for high initial velocities (like 500 m/sec), the simulation seems to take longer to go unstable than in the other cases. What is interesting about this is that the first occurrence of energy increase also takes longer to occur in this case as well.

4.2.3 Significance of Number of Nodes

Most experimenters don’t use 5 nodes to simulate a moving object as that leaves little room for an elastic object to properly simulate the stress and strain. However, it was noticed that when simulating an elastic body in motion that for 5 nodes on the object, that when $h$ was decreased, the values of the friction coefficient that did not go unstable increased. For $h = 1 \times 10^{-5}$, only the case of $\mu = 0$ did not go unstable. However, for $h = 1 \times 10^{-5}$, the object did not go unstable for $\mu \leq .25$. This continued as $h$ was decreased, so that finally by $h = 1 \times 10^{-6}$, the object remained stable for $\mu \leq 1.0$. This seems to imply that as $h$ is decreased, the values of $\mu$ where the simulation will stay stable will increase. For 5 nodes this was seen as early as $h = 3 \times 10^{-5}$, however for 25 nodes this did not start to occur for any cases that were run.

Based on this, it would appear that the proper way to simulate an elastic body in motion under friction is to use less nodes. However, this has a problem. The problem is that the less nodes one uses, the less one factors in stress and strain into the equation. A large object with 5 nodes might as well be treated as a rigid body, as the effects of stress and strain on the object are hardly being simulated. Therefore, what this result seems to imply is that stress and strain are the cause of the instabilities. The more accurately it is represented, the more likely an instability is to occur.
This theory is supported by the fact that when the number of nodes were increased from 25 and the rest of the parameters held the same, the occurrence of the first instability increased. The fact that a decrease did occur going from 250 nodes to 500 nodes is not seen as significant as the total number of timesteps that elapsed before an instability occurred in the 250 node case was so small, there was not room for much more decrease to occur.

4.2.4 Significance of the No Grad Div Experiments

The fact that the cases where the grad div term was commented out (so stress and strain was not factored into the equation) never went unstable adds to the implication that the stress and strain terms are significant for instabilities to occur. If stress and strain was irrelevant to instabilities occurring, then in some cases where the terms were commented out, some kind of instability should still have occurred. However, this was not the case in any experiment run. Taking out the grad div term results in the equations of motion reducing to the following (with some algebra):

\[ u_{tt} = \lambda \Delta u \]  

(4.1)

Equation 4.1 are able to be decoupled now in each of the two separate directions, and each equation is separately stable. Therefore, because each equation is separately stable, no instability should occur when 4.1 is used in place of 3.4 in our simulations for this reason.

4.2.5 Significance of Energy Conservation

It seems, based on these results, it is irrelevant whether energy increased or decreased. In all experiments, the energy between timesteps increased at some point, regard-
less of whether the experiment went unstable or not. Therefore, I would have to conclude that these simulations show that whether or not energy increases between timesteps is not relevant for the experiments, at least in terms of whether an instability occurs.

Why could the energy be increasing then? I would say it is probably increasing due to the collision between the object and the ground. It is known that in rigid body collisions that energy can increase or decrease when two bodies collide [35]. Here, the elastic object in motion is always colliding with another object (the ground). It is possible that due to these collisions that the ground is providing some energy to the object. However, it does not seem that the energy provided is enough to cause an instability, else we would have had all experiments run in this thesis going unstable. It is also possible that two particles within the object are colliding, but that would not change the total energy of the system, as it would just transfer from one particle to another within the system.

4.2.6 Summary of Results

Based on all of these results, I would have to conclude that the cause of the instabilities seems to be the stress and strain terms. Reducing the total number of nodes makes instabilities less likely to occur, but also reduces the effect of stress and strain. Adding to that, removing the stress and strain terms from the equation caused no instability to occur seems to make this a logical assumption. The initial velocity does seem to have some effect on when instabilities occur too, but it is minor. And in all cases, whether or not energy increased between timesteps had no effect on whether an instability occurred or not.

In addition, it seems that the friction coefficient is a huge factor in when instabilities
occur. The smaller the friction coefficient, the longer it takes for an instability to occur. It seems that friction is indeed contributing to the instabilities, but it is not the major cause of them. If friction was the major cause of the instabilities, then the cases where the stress and strain terms were not considered should have gone unstable as well, and that was not the case. Therefore, it seems that friction coefficient, while a major factor in when instabilities occur, is not the main cause of the instabilities occurring.

4.3 What Occurs During an Unstable Experiment

In an effort to get a better understanding of what exactly is causing an instability, I will now examine some experiments in full detail. All unstable experiments mentioned in this section were ran with a Poisson’s coefficient of .3, initial velocity of 30 in the x-direction, 0 in the y-direction, initial displacement of 30 in the x-direction, 0 in the y-direction, \( h = .5 \times 10^{-5} \), and \( \mu = .5 \). The number of nodes will vary and will be noted in each case. For comparison, the experiments will be compared against a stable case of \( \mu = 0.0 \), all other data remains the same as in the unstable case.

4.3.1 Special Notes

To aid in the viewing of an instability, I wrote a program in C that will display the object in space. The camera is centered on the lower left hand corner, this is to prevent the object from sliding off out of the viewscreen. All triangles of the finite element method were drawn on screen as well. To use this program, the Matlab program printed out into a file all the \( x \), \( y \) locations of each node at each timestep, as well as the values of \( N \) and \( F \) on each node at each timestep. Other important data like total number of timesteps simulated,
friction coefficient and such were saved at the beginning of the time. The program is enclosed on the CD, as well as a 5 node case. Go into the respective folders to run each case. The program will fast forward to a location closer to the instability (you will see a blank screen during the loading), then will slide along. An 100 node case was ran as well, however, it is not enclosed on the CD, as the files were over 20 GB and would not fit on the CD. One file was over 10 GB by itself, so breaking up the program over several CD’s was not a practical option for that reason either. Also, the 25 Node case file is over 1.5 GB so again it is not included on the CD due to size, but select screen shots of the 25 node case are enclosed on the CD, and also are present in this thesis.

At this point I invite the reader to go to the folder for 5 node case and run the file instabilityviewer.exe. It will show you what happens in a 5 node experiment. Based on what I saw during that experiment, it seemed that the object started swaying some before it went unstable. Further investigation was conducted with a 25 node case to see better what is going on in these cases. The rest of this section is devoted to discussing what occurred in that experiment.

Also, as the figures were almost a page each, and making them much smaller would lose significant data, all the figures referred to in this section are located in Section 4.4. They are in the order they are referred to here in this section of the thesis.

4.3.2 Leading up to the Instability

At first glance it seems that in the unstable cases, the object just moves along fine, and then ends up leaving the ground unexpectedly. If that was indeed what was going on,
this would imply that the normal contact force becomes too large in calculation at one point, forcing the object to suddenly jump, and causing an instability to be detected. However, a look at the data at the end of an unstable run also showed that all the $x$ values for the nodes on the boundary were not uniform. But, the object was given an uniform force on the left hand side, and the only other force acting on it was gravity (also an uniform force equally applied to all nodes). So, something is causing the object to no longer show an uniform displacement even though only uniform forces were applied. Also, in all the stable cases, the object appeared to have uniform $x$ values all along the boundary, even for an experiment ran for 75000000 timesteps! So something was causing this change. It is unlikely that the normal contact force suddenly getting too large would cause that, as $N$ would most likely be applied uniformly to all the nodes (even if it was too large).

To better analyze this, I had Matlab display all of the 14 digits after the decimal point for all the timesteps. I enclosed all the data collected in a word document titled xdata - instability. As the data is over 30 pages for the specific timesteps I monitored (the paper with all the $x$ and $y$ values for the entire experiment checked in at 5400+ pages...before the computer crashed and stopped counting), I am not going to include all of it in the thesis, but will reference select data from this paper.

What was first noticed was that the object moved along uniformly for a little while, but on the 14th timestep, the following $x$ values were returned along the boundary:

\[ x = 30.02099693614953 \quad 30.02099693614954 \]

This was for two different nodes that were right beside each other. So, by the 14th
timestep, the $x$ values along the boundary disagreed by 1 digit, in the 14th decimal digit. Now, by itself, this is not a big deal, as rounding errors and errors in solving the Conjugant Gradient method can result in a discrepancy of this much between the nodes. And, as a look at Figure 1 in Section 4.4 will show, the object seems to be sliding along just fine at this point. One cannot even notice a discrepancy of that little by looking at the figure, as an error of that size is too small to be seen by the human eye. In fact, it is unlikely one could notice it even if the object was zoomed in to maximum zoom.

However, that is unfortunately not the end of the story. By the 500th timestep, this discrepancy had become worse. Along the boundary the displacement varied from $30.74712623862621$ to $30.74712623862631$. Now the discrepancy was in the last two decimal digits. Still, it was unnoticeable in the experiment and in the image representing the object, as shown in Section 4.4.

As the object moved along, eventually the discrepancy started to show up in the $y$-values. The object had uniform displacement for the $y$-values of the object all the way to the 540th timestep. But, on the 541st timestep, the following values were noticed for two adjacent nodes along the boundary:

$$y = 0.00000000000001 - 0.00000000000001$$

Now the $y$ values were starting to disagree in the 14th digit. The difference in the $x$-values had propagated to the $y$-values now. Still, it was minor at this point and hardly noticeable, as the figure in Section 4.4 shows.

The experiment was continued, and the $x$ values and $y$ values were checked at the
2500th, 5000th, 7500th and 10000th timestep to see what the $x$ values were. In all cases, the boundary values were getting farther and farther from uniform, even though the only forces applied to the object were uniform in nature. In the 2500th timestep, the discrepancy was in the last 7 digits in both the $x$ and $y$ values. For the 5000th timestep, the discrepancy was in the last 8 digits (of the 14 shown) in the $x$ values, and in the last 7 digits for the $y$ values. However, for the 5000th timestep, the fact that the object is no longer moving along uniformly is starting to be visible now. A close look at the appropriate figure in Section 4.4 will show that those are not straight lines anymore going through the nodes, but they are starting to slope some.

After the experiment had run for 7500 timesteps, the $x$ values along the boundary now disagreed on the last 10 digits (of the 14 shown). The $y$ values were in disagreement on the last 8 digits displayed. Also, the object is starting to sway some and is obviously no longer moving along uniformly. The reader may also notice that in the figure showing the object after 7500 timesteps that is displayed in Section 4.4, that after 7500 timesteps it appears that the object has gone through the floor. That is actually an artifact of the display algorithm, as looking at the actual $y$ values at that point in time shows that the leftmost node is actually supposed to be at the $y$ value of $-0.00000002$. That should be too small to show up in display, but due to the way OpenGl is handling the difference between the surface we are in contact with being at $y = 0.0$ and the node being at $y = -0.00000002$, it is displaying the node as being under the floor. Unfortunately there is no way to fix that short of significantly magnifying the view window, and that would result in no longer being able to see the entire object on one screen.
After running the experiment for 10000 timesteps, the object still seems to be experiencing sway, but nothing too major yet. A look at the $x$ values revealed that the $x$ values were in agreement for only the first 3 digits after the decimal point, and the $y$ values were in agreement for only the first 6 digits after the decimal point.

The next screen shot was taken at the last timestep before an instability was detected. Looking at the picture located in Section 4.4, one can see that the bottom boundary of the object seems to be looking more like a parabola than a straight line. Also, some sway still seems to occur in the object. A look at the $x$ values show that the $x$ values along the boundary are only uniform up to 2 digits after the decimal point, and the $y$ values were uniform only up to 3 digits after the decimal point. That is as far as the $y$ values can disagree and not be declared an instability, as the $y$ values must be 0.000$x$ at the worst case scenario, anything but 0’s in the other values will either be flagged as significant interpenetration if negative, or instability if positive.

4.3.3 At and After the Instability

The $x$ and $y$ values were not compared to see how nonuniform they are after the instability was detected, as it appears obvious that after that point the $x$ and $y$ values disagree after the $3^{rd}$ decimal digit in the best case scenario. Nothing was found in the screen shots in this section to disprove that.

After an instability was detected, the object continues to sway back and forth. Watching the enclosed program for the 5 node case, the reader can see some of the shock waves starting to ripple up the object, and the object move and sway back and forth. Also,
the reader will notice the object nodes no longer appear to be uniformly spaced. The object started with all the nodes .01 meters apart, but now it seems that is no longer the case, as the nodes get closer and farther from each other. This is more apparent if Young’s Modulus ($E$) is changed to be 1, as the nodes on top will collapse down to lay on the surface the object is sliding along, making all the nodes an infinitesimally small distance apart in the $y$ direction. When that happens, it is possible that the nodes switch places even, with the bottom boundary now being a node that was originally on the top. None of the nodes seem to switch places in this simulation with $E = 210 \text{GPa}$ (which is the value for steel), but it is worth noting that it could occur, and will if Young’s Modulus is small enough.

The next figure in Section 4.4 shows the object after 17500 timesteps have been simulated. In this picture, the reader should notice that parts of the bottom boundary of the object are in contact, and other parts are not. Based on this, it is apparent that once an instability occurs, the object becomes even more and more nonuniform, even though the original forces given to the object were uniform.

The next three figures (all located in Section 4.4) show the object as the experiment continues along. The third figure shows what the object is doing after the 30000th timestep. Note that this is now about 19000 timesteps after the instability occurred. The object is now quite visibly not uniformly moving along. The left hand side has a couple obvious valleys that should not be there for a rectangular object that was given uniform forces.

The last figure, located in Section 4.4, shows the object after 35000 timesteps have been simulated. Here, the reader will notice that the entire object is off of the ground! So, by this point, the difference in the $x$ and $y$ values has propagated all the way to the entire
object going off into space. Obviously this is not a realistic simulation anymore, as if it were, every time you slid a book across the table to a friend your friend would have to watch out for the possibility the book flies up and hits them in the face. Further simulation of the object shows that the object will hover in the air for a while, then come back down, impact with the ground, then go back into the air. This skipping behavior appears to occur the rest of the simulation, or at least to 100000 timesteps. The object never appears to come back down and to go to a stationary rest state, but alternates between being in contact and not in contact. It is believed that if the experiment was ran long enough, eventually the friction forces would cause the initial velocity to dissipated, and the object would come to rest. However, the final x value returned would be horrendously off of the correct value, so no practical reason for running the experiments that long could be found.

4.3.4 Changes in the Normal Contact Force and Friction

Based on the work in the previous section, it seems that something is occurring to make the object start to no longer move uniformly along even when forces are applied uniformly to the object. To further investigate this, the normal contact force ($N$) and the frictional contact force ($F$) were analyzed in detail. As $F = -\mu N$ for our problem, only $N$ was analyzed in the end as changes (and perturbations) in $N$ will result in perturbations in $F$ as well.

The same experiments were ran as in the previous section, with 25 nodes and friction coefficient .5, starting velocity $30m/sec$ applied uniformly to the left hand side, starting displacement of 30 meters, Poisson’s ratio of .3 and $E = 210$. This time the normal contact
force was analyzed to see what is going on in the cases where the experiment goes unstable.

The normal contact force immediately started to show some interesting results. On the very first timestep, the normal contact force was not applied uniformly to the object! The normal contact forces were in the range of 0.07758819499222 to 0.07758819499265. Discrepancy occurred in the last 2 decimal digits of the normal contact force. An attempt to tighten the error bound on the Jacobi method to $1 \times 10^{-16}$ resulted in some improvement, with the discrepancy occurring in the last displayed decimal digit only. Attempts at using a smaller error bound on the Jacobi method resulted in the method falling to converge, but that is due to the fact that $1 \times 10^{-16}$ is actually smaller than unit round off on most modern computers (the fact that the experiment converged for even one timestep for an error bound of $1 \times 10^{-16}$ was actually surprising for that very reason).

What is surprising about this is that even with an error bound of $1 \times 10^{-15}$, the normal contact forces agree only up to a factor of $1 \times 10^{-13}$. So it appears that this discrepancy cannot be attributed totally to errors in calculating the normal contact forces, and therefore something else must be causing the forces to be applied non-uniformly to the object even when uniform initial forces were applied.

To check and see if the normal contact force discrepancies were the entire cause of the discrepancies in the $x$ and $y$ terms, the experiments was re-run with the normal contact force was averaged out across all the nodes on the boundary (so now this was essentially treating the object as a rigid body). What was found is that the discrepancies in the $x$ and $y$ terms still occurred, just later on. The first documented discrepancy in the $y$ term was found after the 23rd timestep, and the first documented discrepancy in the $x$ term was
found after the 24th timestep. So even with the $N$ term being averaged out (so forced to be applied uniformly), the $x$ and $y$ terms on the boundary still became nonuniform, it just took longer to occur than in the previous experiments. And, incidentally, the $y$ terms ceased to be uniform sooner than the $x$ terms in this select case. So it seems that it is not true that the $x$ terms will always go nonuniform first before the $y$ terms. Instead, it will vary by experiment.

The experiments were continued (with the normal contact force being applied normally for an elastic body now) to see what happened with the normal contact force over time. By the 500th timestep, the normal contact forces were uniform only up to the 5th digit after the decimal point. By the 2500th timestep the normal contact force was completely non-uniform, as some boundary nodes were being calculated as needing no normal contact force, and some were needing normal contact forces as large as $3.67922882773809$.

One interesting thing about this experiment though was seeing how the normal contact force oscillated throughout the experiment. Attached are two figures, showing what the average normal contact force looks like as the experiment proceeds. The first figure shows the normal contact force up to a little more than 12000 timesteps. As the reader can see, the normal contact force seems to go in a regular pattern, then the normal contact force suddenly starts to spike. The last figure shows that the normal contact force does start to oscillate between 0 and huge values, which makes sense seeing as the object is now skipping during the experiment and is therefore not always in contact.

In an attempt to see if errors in the Jacobi method were the entire reason for an instability to occur, the Jacobi tolerance was tightened up to $1 \times 10^{-13}$ (smaller values than
this caused the Jacobi method to fail to converge in reasonable time at various points in the experiment). This ended up changing absolutely nothing, except the experiment took hours longer to run than before. The $x$ and $y$ values ceased to be uniform at almost the exact same points in time, and the instability actually occurred sooner with the tightened tolerance. So apparently the errors in the normal contact force are not the main cause of the instability occurring.
4.3.5 Comparison with Stable Experiments

To check and see if these results occurred only in unstable cases, a frictionless case was ran and checked to see if any discrepancy occurred on the $x$ and $y$ values. In the frictionless case, the object was given an uniform force in the $x$ direction on the left hand side, and only subject to the force of gravity in the $y$ direction. The object never went unstable in the frictionless case. However, the object did show some non-uniformity in the displacement of the object over time. After 10000 timesteps, the displacement agreed up to 11 digits after the decimal point. Nothing changed after 50000 timesteps, the object still had displacement agreeing up to 11 digits after the decimal point. So in the frictionless case,
the object did show some non-uniformity on the final displacement, but all the nodes were uniform up to 11 decimal places, which is pretty significant. This shows that in the absence of friction it seems that a uniform force applied to an object will result in an uniform final displacement, at least up to a factor of $1 \times 10^{-11}$.

Another stable experiment was ran for comparison, this time the object was under a friction coefficient of .5 and initial displacement of 30 meters, but had no initial velocity applied. This object should stay stationary and never move. Here is where the flaw in not allowing for static friction in the experiment shows, as the object did drift very slowly forward, with a displacement of 30.00007543581968 meters after 50000 timesteps ($h = 5 \times 10^{-5}$). Longer experiments did show the object drifting to a final displacement of 30.02 sometimes. This could be either due to the fact that static friction was not accounted for in the experiments, or errors in the Conjugant Gradient method in solving the equations of motion. Also, the object was only uniform in displacement up to the 4th digit after the decimal point, with some nodes showing a displacement of 30.00007 and some showing 30.00006. So it seems that even in the absence of an initial velocity but with friction present, the object shows non-uniform movement after the 4th digit after the decimal point after 50000 timesteps.

4.3.6 Implications

These results, combined with the results in Section 4.2, seem to imply a few things for anyone who wishes to model an elastic object in motion under Coulomb friction. First of all, if one removes the friction term, no instability occurs. But, with the friction term
present, the object starts to show non-uniformity in movement even when an uniform force was applied to the object to begin with. This non-uniformity seems to propagate in the friction case and get worse the longer the experiment runs.

The errors in the solution of the equations seem to be a contributing factor to instabilities occurring, but they are not the main factor. The errors seem to propagate throughout the equation and get worse over time in the unstable experiments. However, the fact that even the stable experiments showed some of these errors cannot be ignored. Still, reducing these errors would probably go a long way to reducing the occurrence of these instabilities.

One cause of the errors here has to do with the fact that this model is iterative. For documentation on what can go wrong with iterative models, see [12, 13], where the author demonstrates errors and how that for an iterative model for $x_{n+1} = x_n^2 - 2$, $x_0 = .37$, two different models of calculators or computers will give two totally different answers as soon as $n = 45$. This is due to the fact that computers and calculators can only store a finite amount of data, and therefore decisions have to be made on how many digits to keep.

In this model, this is seen by how two different computers under two different chip sets (one by AMD, one by Intel) give different answers for when the instability occurs in the experiment. It is not unfeasible for someone to later on to not get an instability to occur on an experiment that I found to be unstable due to this reason as well.

It is possible that some of this error, and even instabilities, are due to the model used, which is something that future research will explore. However, any model will incorporate error in solving the problem. There is no reasonable way to incorporate a model of an object in motion that will not have some error in the solution.
Therefore, based on these results, I would highly suggest that anyone wishing to model an elastic object in motion under Coulomb friction use the models for only short term modeling. If a way to model an elastic body in motion under Coulomb friction without using an iterative model can be found, that would go significantly far in reducing the error and increasing the accuracy of the solutions. It is my belief, based on these results that further research needs to focus on either reducing the error in the solutions (either by finding better ways to solve the LCP, or better ways to solve the system of equations), or on seeing if a way can be found to model the object without using an iterative model.

4.4 Tables And Figures

Here are enclosed all of the tables that were a full page long, which were referenced in Section 4.2, as well as all the figures referenced in Section 4.3
Table 4.2: Variance of Friction Coefficient, 25 Nodes

<table>
<thead>
<tr>
<th>$\mu$</th>
<th>$h = 1 \times 10^{-4}$</th>
<th>$h = 5 \times 10^{-5}$</th>
<th>$h = 3 \times 10^{-5}$</th>
<th>$h = 1 \times 10^{-5}$</th>
<th>$h = 5 \times 10^{-6}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>0.05</td>
<td>7.2201</td>
<td>11.45695</td>
<td>11.67552</td>
<td>11.52811</td>
<td>11.33299</td>
</tr>
<tr>
<td>0.1</td>
<td>3.1591</td>
<td>5.37545</td>
<td>5.43312</td>
<td>5.60428</td>
<td>5.734625</td>
</tr>
<tr>
<td>0.15</td>
<td>2.0018</td>
<td>3.03875</td>
<td>3.6543</td>
<td>3.84554</td>
<td>3.67902</td>
</tr>
<tr>
<td>0.2</td>
<td>1.2507</td>
<td>2.1241</td>
<td>2.51658</td>
<td>2.68294</td>
<td>2.5285</td>
</tr>
<tr>
<td>0.25</td>
<td>0.9799</td>
<td>1.32005</td>
<td>1.93827</td>
<td>1.83399</td>
<td>1.940055</td>
</tr>
<tr>
<td>0.33</td>
<td>0.7299</td>
<td>0.9014</td>
<td>1.42155</td>
<td>1.42447</td>
<td>1.36055</td>
</tr>
<tr>
<td>0.4</td>
<td>0.5547</td>
<td>0.73205</td>
<td>1.18512</td>
<td>1.13407</td>
<td>1.12904</td>
</tr>
<tr>
<td>0.5</td>
<td>0.4553</td>
<td>0.56365</td>
<td>0.95478</td>
<td>0.90055</td>
<td>0.90589</td>
</tr>
<tr>
<td>0.6</td>
<td>0.3315</td>
<td>0.4949</td>
<td>0.73323</td>
<td>0.76677</td>
<td>0.73801</td>
</tr>
<tr>
<td>0.67</td>
<td>0.3083</td>
<td>0.43615</td>
<td>0.62841</td>
<td>0.68869</td>
<td>0.70549</td>
</tr>
<tr>
<td>0.83</td>
<td>0.2488</td>
<td>0.33705</td>
<td>0.44613</td>
<td>0.54504</td>
<td>0.5339</td>
</tr>
<tr>
<td>1.0</td>
<td>0.2098</td>
<td>0.2654</td>
<td>0.36525</td>
<td>0.4897</td>
<td>0.48418</td>
</tr>
<tr>
<td>6.33</td>
<td>0.0082</td>
<td>0.0129</td>
<td>0.01722</td>
<td>0.02141</td>
<td>0.02224</td>
</tr>
</tbody>
</table>
**Table 4.3: Variance of Friction Coefficient, 10 Nodes**

<table>
<thead>
<tr>
<th>$\mu$</th>
<th>$h = 1 \times 10^{-5}$</th>
<th>$h = 5 \times 10^{-6}$</th>
<th>$h = 1 \times 10^{-6}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>0.05</td>
<td>12.3458</td>
<td>13.524845</td>
<td>10.128007</td>
</tr>
<tr>
<td>0.1</td>
<td>6.15701</td>
<td>6.218765</td>
<td>5.185112</td>
</tr>
<tr>
<td>0.15</td>
<td>3.99366</td>
<td>4.077405</td>
<td>3.701971</td>
</tr>
<tr>
<td>0.2</td>
<td>2.90278</td>
<td>3.10982</td>
<td>2.844308</td>
</tr>
<tr>
<td>0.25</td>
<td>2.20489</td>
<td>2.456385</td>
<td>2.311625</td>
</tr>
<tr>
<td>0.33</td>
<td>1.56236</td>
<td>1.915635</td>
<td>1.719434</td>
</tr>
<tr>
<td>0.4</td>
<td>1.3428</td>
<td>1.444785</td>
<td>1.451079</td>
</tr>
<tr>
<td>0.5</td>
<td>1.01728</td>
<td>1.11833</td>
<td>1.168225</td>
</tr>
<tr>
<td>0.6</td>
<td>0.85696</td>
<td>0.901745</td>
<td>0.950919</td>
</tr>
<tr>
<td>0.67</td>
<td>0.79362</td>
<td>0.80505</td>
<td>0.800873</td>
</tr>
<tr>
<td>0.83</td>
<td>0.63089</td>
<td>0.68337</td>
<td>0.689048</td>
</tr>
<tr>
<td>1.0</td>
<td>0.54303</td>
<td>0.57516</td>
<td>0.580333</td>
</tr>
<tr>
<td>6.33</td>
<td>0.05646</td>
<td>0.05749</td>
<td></td>
</tr>
</tbody>
</table>
Table 4.4: Variance of Friction Coefficient, 5 Nodes

<table>
<thead>
<tr>
<th>$\mu$</th>
<th>$h = 5 \times 10^{-5}$</th>
<th>$h = 3 \times 10^{-5}$</th>
<th>$h = 1 \times 10^{-5}$</th>
<th>$h = 5 \times 10^{-6}$</th>
<th>$h = 3 \times 10^{-6}$</th>
<th>$h = 1 \times 10^{-6}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>0.05</td>
<td>14.6725</td>
<td>22.97973</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>0.1</td>
<td>7.5258</td>
<td>8.28375</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>0.15</td>
<td>4.9373</td>
<td>5.20668</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>0.2</td>
<td>3.67295</td>
<td>3.62103</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>0.25</td>
<td>2.6731</td>
<td>2.95728</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>0.33</td>
<td>1.954</td>
<td>1.98033</td>
<td>8.03976</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>0.4</td>
<td>1.4704</td>
<td>1.51737</td>
<td>1.04126</td>
<td>3.309205</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>0.5</td>
<td>1.23135</td>
<td>1.24797</td>
<td>0.75292</td>
<td>1.79793</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>0.6</td>
<td>0.9436</td>
<td>0.98757</td>
<td>0.5706</td>
<td>1.034405</td>
<td>4.439727</td>
<td>N/A</td>
</tr>
<tr>
<td>0.67</td>
<td>0.8475</td>
<td>0.85047</td>
<td>0.52178</td>
<td>0.702755</td>
<td>3.962793</td>
<td>N/A</td>
</tr>
<tr>
<td>0.83</td>
<td>0.69845</td>
<td>0.7608</td>
<td>0.43006</td>
<td>0.508775</td>
<td>3.112983</td>
<td>N/A</td>
</tr>
<tr>
<td>1.0</td>
<td>0.5713</td>
<td>0.62331</td>
<td>0.41272</td>
<td>0.42379</td>
<td>2.531548</td>
<td>N/A</td>
</tr>
<tr>
<td>6.33</td>
<td>0.1097</td>
<td>0.11754</td>
<td>0.11603</td>
<td>0.08908</td>
<td>0.072621</td>
<td>0.075095</td>
</tr>
<tr>
<td>$\mu$</td>
<td>Total Timesteps Simulated</td>
<td>Final X Displacement</td>
<td>Final Y Displacement</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>---------------------------</td>
<td>----------------------</td>
<td>----------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td>2500000</td>
<td>948.3678</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>819614</td>
<td>489.183</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.15</td>
<td>573427</td>
<td>336.1225</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>1500000</td>
<td>259.588</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td>856883</td>
<td>213.64</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.33</td>
<td>843264</td>
<td>169.15</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>943346</td>
<td>144.78</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>750000</td>
<td>121.83</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>678814</td>
<td>106.53</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.67</td>
<td>31\311739</td>
<td>98.53</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.83</td>
<td>490183</td>
<td>85.53</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>750000</td>
<td>75.91</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td>750000</td>
<td>52.89</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.0</td>
<td>750000</td>
<td>45.3</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.33</td>
<td>750000</td>
<td>37.23</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 4.15: 14 Timesteps

Figure 4.16: 500 Timesteps
Figure 4.17: 541 Timesteps

Figure 4.18: 2500 Timesteps
Figure 4.19: 5000 Timesteps

Figure 4.20: 7500 Timesteps
Figure 4.21: 10000 Timesteps

Figure 4.22: Instability
Figure 4.23: 15000 Timesteps
Figure 4.24: 17500 Timesteps
Figure 4.25: 20000 Timesteps
Figure 4.26: 25000 Timesteps
Figure 4.27: 30000 Timesteps
Figure 4.28: 35000 Timesteps
CHAPTER 5
THEORETICAL RESULTS

After seeing the results of the experiments so far, an attempt to find an alternate method was done next. This chapter is devoted to explaining where the current method fails to be energy conservant, and then an energy conservant method is proposed for future implementation.

5.1 Is The Current Method Energy Conserving?

The Jacobi method is dependent on the condition that the value of the friction force only depends on the current solution for the velocity term. As in this paper, only the problem of forward progress is considered here, the condition

\[ |F_i| \leq \mu N_i \tag{5.1} \]

ends up becoming

\[ F_i = -\mu N_i \tag{5.2} \]

where \( i \) is the current timestep that the friction force is being solved for. The important thing though is this is dependent on the fact that the friction term depends only on the current velocity term. Another way to define the friction term is that the friction term \( F \) can be seen as the solution of the following problem:

\[ \min_i (F_i)^T \gamma \cdot v^i \tag{5.3} \]
such that

$$|F^i| \leq \mu N^i$$

holds. Here, $\gamma$ is the boundary matrix as computed in the finite element method in the tangential direction.

To see if the current method is energy conservant under this friction condition, start off by defining an energy functional. The total change in energy for (3.24) can be given by

$$E(u^i, v^i) = \frac{1}{2} \int |u|^2 + \frac{1}{2} \int |v|^2 - \int f \cdot u$$

(5.4)

where the first term is the kinetic energy, the second term is the elastic energy, and the third term is the external potential energy. We want that this term is always decreasing, or that

$$E(u^{i+1}, v^{i+1}) \leq E(u^i, v^i) \forall i \geq 0$$

Theorem 5.1. With the friction term reformulated as the problem of finding the min of

$$\gamma^T F^{i+1} \left( \frac{v^{i+1} + v^i}{2} \right)$$

where $$|F^{i+1}| \leq \mu N^{i+1}$$ holds as well, then energy is conserved for the problem of an elastic object in motion under Coulomb friction

Proof. For energy to be conserved, we need to show that $E(u^{i+1}, v^{i+1}) \leq E(u^i, v^i)$ for any $i \geq 0$. To check into this, we will first of all look at the equations of motion. With some suitable reformulation of the equations of motion, we will get:

$$M \cdot \left( \frac{v^{i+1} - v^i}{h} \right) = -K \cdot \left( \frac{u^{i+1} + u^i}{2} \right) + f + C^T N^{i+1} + \gamma^T F^{i+1}$$

(5.5)
\[ \frac{u^{i+1} - u^i}{h} = \frac{v^{i+1} + v^i}{2} \]  

(5.6)

where \( M \) is the mass matrix computed via the finite element method and \( K \) is the stress matrix computed via the finite element method.

Take equation (5.5), multiply both sides of the equation by \( \frac{u^{i+1} - u^i}{h} \), and the equations will become:

\[
M \cdot \left( \left| v^{i+1} \right|^2 - \left| v^i \right|^2 \right) = -K \cdot \left( \frac{\left| u^{i+1} \right|^2 - \left| u^i \right|^2}{2h} \right) + f \cdot \left( \frac{u^{i+1} - u^i}{h} \right) + C^T N^{i+1} \cdot \left( \frac{u^{i+1} - u^i}{h} \right) + \gamma^T F^{i+1} \cdot \left( \frac{u^{i+1} - u^i}{h} \right)
\]

(5.7)

With a suitable rearrangement, the equation (5.7) becomes

\[
\frac{1}{2h} (M \cdot \left| v^{i+1} \right|^2 + K \cdot \left| u^i \right|^2 - f \cdot u^{i+1}) = \frac{1}{2h} (M \cdot \left| v^i \right|^2 + K \cdot \left| u^i \right|^2 - f \cdot \left( \frac{u^{i+1} - u^i}{h} \right))
\]

\[
+ C^T N^{i+1} \cdot \left( \frac{u^{i+1} - u^i}{h} \right) + \gamma^T F^{i+1} \cdot \left( \frac{u^{i+1} - u^i}{h} \right)
\]

(5.8)

Now, a suitable integration of (5.4) gives us the following, after a substitution into (5.8) and multiplying both sides of the equation by \( h \):

\[
E(u^{i+1}, v^{i+1}) = E(u^i, v^i) + C^T N^{i+1} \cdot \left( \frac{u^{i+1} - u^i}{h} \right) + \gamma^T F^{i+1} \cdot \left( \frac{u^{i+1} - u^i}{h} \right)
\]

(5.9)

So, if the two remaining terms on the RHS side can be shown to be negative, energy will be conserved. Stewart, in his paper [33] shows that the term \( N^{i+1} \cdot \left( \frac{u^{i+1} - u^i}{h} \right) \) is \( \leq 0 \) before integration, and the integral of a negative function is always negative (and the matrix
\( C \) is the result of the integration of the normal contact force along the boundary. Therefore, we will get:

\[
E(u^{i+1}, v^{i+1}) \leq E(u^i, v^i) + \gamma^T \cdot F^{i+1} \cdot \left( \frac{u^{i+1} - u^i}{h} \right)
\]

Or, using (5.6), the equation now becomes:

\[
E(u^{i+1}, v^{i+1}) \leq E(u^i, v^i) + \gamma^T F^{i+1} \cdot \left( v^{i+1} + v^i \right)
\]

(5.10)

The problem is that this term is not always guaranteed to be negative. If the object is always moving forward, and no other force acts to make the object move in any other direction, we will have that \( v \) is always positive in the tangential direction, and therefore that term will be positive. However, if the friction term is redefined such that we are now trying to find:

\[
\min \gamma F^{i+1} \cdot \left( v^{i+1} + v^i \right)
\]

(5.11)

such that

\[
|F^{i+1}| \leq \mu N^{i+1}
\]

(5.12)

holds, then we will get that 0 is a solution of this problem, as \(|0| \leq \mu N^{i+1}\) holds for all values of \( \mu \) and \( N \). Therefore, the minimum of problem (5.11) has to be smaller than or equal to 0 as 0 is a solution (just maybe not the minimum). This gives us:
\[ \gamma^T F^{i+1} \cdot \left( \frac{v^{i+1} + v^i}{2} \right) \leq 0 \] (5.13)

Or therefore, via substitution of (5.13) into (5.10), we get

\[ E(u^{i+1}, v^{i+1}) \leq E(u^i, v^i) \]

as desired

The proof leads us to conclude that energy was not being conserved in the method implemented in Chapters 3 and 4. Indeed, this was experimentally shown and is mentioned in the results, however based on the work in the proof, we can now say that energy will not be conserved as long as:

\[ C^T N^{i+1} \left( \frac{u^{i+1} - u^i}{2} \right) + \gamma^T F^{i+1} \left( \frac{v^{i+1} + v^i}{2} \right) \geq 0 \]

occurs for any given timestep \( i \).

### 5.2 Uniqueness of Solutions to the LCP

This section is dedicated to seeing if the LCP that we are trying to solve yields a unique solution

#### 5.2.1 The Jacobi Model

To solve the problem in Chapter 3, we assumed that the friction force did not change much in between timesteps, and used the old value of friction to approximate the normal contact force, then we solved the LCP for the normal contact force via a Jacobi method, and
used that new term to get the new friction force. While the method works, some error will exist in the solution, mainly in the frictional force being too large for an object in forward motion. However, due to that method the issue of whether or not the LCP has a solution did not have to be addressed directly, as the Jacobi method always converges to a value.

To see if an LCP has an unique solution, we need to look at the problem:

\[ 0 \leq z \perp Mz + q \geq 0 \]

which can be solved by a variety of methods. Two really popular methods for solving LCP’s are Lemke’s method and the Simplex method. Details on how to implement these methods can be found in [15]. However, neither method is guaranteed to converge. It is possible for Lemke’s method to alternate between two different values, or for Lemke’s to even end up on an unbounded ray. Neither conditions are desirable for our problem.

For our problem, we have four LCP’s that need to be solved per node (if we do not restrict the problem any). For convenience of notation, the subscripts indicating which node we are checking the conditions for will be omitted. Let \( F^i = F^{i+} - F^{i-} \), then the four LCP’s are:

\[ 0 \leq F^{i+} \perp \lambda^i + \gamma^T(v^i) \geq 0 \] (5.14)

\[ 0 \leq F^{i-} \perp \lambda^i - \gamma^T(v^i) \geq 0 \] (5.15)

\[ 0 \leq N^i \perp u^i - g \geq 0 \] (5.16)
where (5.16) is the LCP used in the previous model in Chapters 3 and 4. Previously we assumed the object was always going forward, so (5.15) was irrelevant for our model as \( F^i \), by definition, is the friction acting when the object is moving backwards. (5.17) can be thought of as a Karsh-Kuhn Tucker condition on the problem, where \( \lambda^i \) kicks in only when the constraint (in this case the need for a normal contact force due to the object touching the ground) is valid. Otherwise, \( \lambda^i \) will be positive, which will force \( N^i \) and \( F^i \) to be 0 for that timestep \( i \) by the remaining 3 LCP’s.

In Chapter 3, we showed that equation (5.16) can be solved to yield:

\[
0 \leq \lambda^i \perp \mu N^i - F^i \geq 0 \tag{5.17}
\]

\[
0 \leq N^i \perp \frac{h^2}{2} C^T A^{-1} CN^i + \frac{h^2}{2} C^T A^{-1} [(M - \frac{h^2}{4} K)v^{i-1} + hDF^i + \frac{h^2}{2} Ku^{i-1} + hEf] + C^T v^{i-1} + C^T (u^{i-1} - g) \geq 0 \tag{5.18}
\]

which gives us our LCP’s to solve. By the assumptions we made back in Chapter 3, equation (5.14) is assumed to always apply as we are in continuous contact, and therefore (5.17) was also assumed to always apply. (5.15) was irrelevant in our model as the object was assumed to be always going forward, so the only equation in the LCP that had to be solved was (5.18).

To solve the LCP in (5.18) easier, we will rewrite the equation above as follows:

\[
0 \leq N^i \perp \frac{h^2}{2} C^T A^{-1} CN^i + R \geq 0 \tag{5.19}
\]
with $R$ standing for all the other terms after the $N^i$ term in the LCP in (5.18). This now
gives us the LCP in the standard form $0 \leq z \perp Mz + q \geq 0$, and we can apply standard
LCP theory to this problem. Most of the work on existence and uniqueness of solutions
was done by R. Cottle in [15]. In that book, it is shown that as long as the matrix $M$ is
positive definite, the LCP will always have a solution. Also, if the matrix $M$ is a $P$ matrix
(defined in Chapter 2), then a unique solution to the matrix will exist for all vectors $q$ that
are provided.

Therefore, for our LCP to be guaranteed to have an unique solution, we need that
the matrix $C^TA^{-1}C$ is a $P$ matrix, or has all positive real eigenvalues. As we don’t know
the exact values of $A^{-1}$, calculating the eigenvalues of $C^TA^{-1}C$ is rough to do precisely.
Thankfully, we have a theorem that tells us that if $\lambda$ is an eigenvalue of $A$ then $\frac{1}{\lambda}$ is an eigen-
value of $A^{-1}$. So if we can find the eigenvalues of $A$, we can then find the eigenvalues of
$A^{-1}$, which can help us find the eigenvalues of $C^TA^{-1}C$. Unfortunately standard approxi-
mation theory is useless here as we need that $\lambda$ is real and positive, and most approximation
theory gives us $|\lambda|$, which is useless as we need the sign of $\lambda$ as well as the value of $\lambda$.

Therefore, as $A$ is too huge to find the eigenvalues well by hand, and approximation
theory will only tell us $|\lambda|$ (usually), what I did instead was have matlab compute $A^{-1}$ ex-
actly for all the cases that the experiments were conducted for in this thesis, then figured out
what the matrix $C^TA^{-1}C$ was, then had matlab compute the biggest and smallest eigenvalue
of the matrix. If the smallest eigenvalue was positive, then we would know that $C^TA^{-1}C$
had all positive eigenvalues, and was therefore a $P$ matrix by the theorem in Chapter 2.
This would guarantee that the Jacobi method for implementation of the LCP has an unique
solution then, and we did not have to worry about the possibility that the Jacobi method was returning one of two (or more) solutions to the LCP problem.

To see if $C^T A^{-1} C$ had all positive eigenvalues, it was necessary to find out what variables influenced the values of this matrix. As the reader may recall, $A = M + \frac{h^2}{4} K$, where $M$ is the mass matrix of the discretized problem, and $K$ is the stress matrix of the discretized problem. The values of $M$ and $K$ however, only depended on the region being integrated, which varies depending on the value of $k$. Therefore, if $k$ changes, the eigenvalues of $A$ will change. Also, as $A$ does depend on $h$ as well, every time we change the value of $h$, the eigenvalues of $A$ will change as well. However, no other values that we vary in the experiment ($v_0$ or $u_0$ for instance) affect the values of $A$. Therefore, due to this, we only need to check if $C^T A^{-1} C$ is a $P$ matrix for each value of $h$ and $k$ that was used in the experiments conducted previously.

Figures below show the smallest eigenvalue for this matrix for the 5, 10, and 25 node cases. As $k$ depends on the nodes used (as we assumed the object was .25 units in all dimensions except in special cases), it was necessary to check the eigenvalues for all the different nodes that the experiment was conducted on. The figures plot the $h$ value used versus the smallest eigenvalue as found by matlab. As the reader can see, these figures show that for all the experiments conducted in this thesis, the matrix $C^T A^{-1} C$ is a $P$ matrix, and therefore the LCP problem as solved by the Jacobi method does have a unique solution for each timestep, for those values of $h$ and $k$ that were tested.

The reader will notice as they look at these figures that as the values for $h$ get smaller, the value of the smallest eigenvalue does not change much, and in all cases there
is a point where for all $h$ smaller than a certain value, the smallest eigenvalue of the matrix remains the same. So it appears that if future experiments were conducted with smaller $h$ values than those used here, the Jacobi method would still yield an unique solution as the matrix $C^TA^{-1}C$ will be a $P$ matrix.

Also, the reader will notice that as the number of nodes of the matrix increases, the smallest eigenvalue found increased too. Therefore, it seems reasonable to assume that the matrix $C^TA^{-1}C$ will be a $P$ matrix for the 100, 250 and even 500 node cases. Unfortunately it is impossible to calculate $A^{-1}$ directly for the object represented by 100 nodes, as the matrix $A^{-1}$ is now a $20000 \times 20000$ matrix for a 2 dimensional object. Attempts to compute
the matrix $A^{-1}$ directly with the object represented by 100 nodes in each direction resulted in an out of memory error with 4 GB of memory. Perhaps as the available memory of computers increase a direct calculation of $A^{-1}$ will be possible, but for now it is not. Still, these results make it reasonable to believe that the matrix will still be a $P$ matrix for these larger problems, due to the fact that the smallest eigenvalue is increasing as the number of nodes increase.

Note this does not prove that the entire problem has a unique solution, only that the LCP under the Jacobi method has an unique solution per timestep that the problem is solved. Whether all of those come together to give an unique solution to the discretized
problem or even the continuous problem is still an open question for further research. As of this thesis, no one has even proven that solutions exist to the discretized or continuous problem (to the author’s knowledge).
CHAPTER 6
CONCLUSION AND FUTURE WORK

This chapter will start off by suggesting some future research on this subject, then will wrap up the main points of the thesis

6.1 Future Research

6.1.1 A Changed Model

Based on the results in Chapter 5, future research in this field should work on seeing if it is possible to implement a model where the friction term is implemented as:

$$\min \gamma^T \begin{pmatrix} F_{i+1} & (v_{i+1} + v_i) \end{pmatrix}$$

such that

$$|F_{i+1}| \leq \mu N_{i+1}$$

holds as well. Attempts to implement the model using this have so far proven unfruitful, as so far attempts to model this condition are resulting in the matrix $M$ in the LCP being singular. Future research will see if a way to model an object in motion using these friction conditions exists. Particularly, we will need to find a model that will allow us to solve an LCP with a singular matrix used. To do this, an approximate matrix that is nonsingular will probably have to be employed so that an approximate solution can be found to the LCP.
6.1.2 Further Investigation Into the Current Model

In the 5 node case, the object eventually started to stay stable as the values of $h$ decreased. Future research into whether this occurs for 10 nodes and even 25 nodes, and for what values of the friction coefficient as well as $h$ is warranted. It is possible that maybe for $h = 1 \times 10^{-8}$ we may find that for 10 nodes the experiment never goes unstable. The only problem with research into this field is that faster computers are desired, else it may take 150 days to run the needed experiments for 10 nodes with $h = 1 \times 10^{-7}$ in order to see what values of the friction coefficient are needed to make the object stay stable.

Also, further research into whether lowering the initial velocity to see if that changes when instabilities occur are warranted as well. Based on the research done in this paper, it seems that the object does not go unstable if the initial velocity is lower than (or equal to) $1\text{ m/sec}$. Further research to see if using an initial velocity of $5\text{ m/sec}$ (or 10, or 15) and seeing when instabilities occur for various friction coefficients are also warranted.

6.1.3 Change to the Finite Element Method

The reader probably noticed that the finite element method and all the reference triangles were preset at the beginning of the experiment, based on the timestep and grid spacing set. However, the object starts to stretch and bend as the experiment continues, meaning these uniform reference triangles (and therefore the matrices gotten based on these reference triangles) are no longer accurate. A more accurate model should recompute the matrices with each timestep, using how the triangles are really positioned in the object now to compute the new finite element matrices. This would have a disadvantage of significantly
slowing down the model, and can even result in the model taking 2–5 minutes to run a single timestep. Such a model would not be practical for any real world simulation due to how slow the model would run, but it would be of theoretical interest, as such a model could show if the instabilities of an elastic object are due to the finite element matrices no longer accurately representing the model they are supposed to

6.1.4 Iterative Method Changes

In Section 4.3, it is suggested that some if not all of the errors of the model is due to the fact that the model is iterative. Iterative models tend to be highly inaccurate in the long term, which is why they are not often used for real world modeling or computation. Further research into whether or not it is possible to model the object in motion without using an iterative model are warranted. It might be possible to solve and move forward in two (or more) timestep increments instead of one. Due to the way the normal contact force and friction force must be solved for each timestep and usually decrease (in the absence of an external force) each timestep finding such a model seems unlikely initially, but if some relation can be found on how the normal contact force of one timestep is based on the previous such a model becomes feasible. Future research into whether there is any relation between the normal contact forces of each timestep would be conducted to see if it is possible to get away from an iterative model for modeling an elastic object in motion

6.1.5 Do Instabilities Always Occur for Some Cases?

Renardy [29] proposes that there are some cases for an object in motion where an instability will always occur. Some preliminary work done in conjunction with Ted Wendt
suggests that this is indeed the case for an elastic body in motion. The preliminary work suggests that for a given Poisson’s ratio, that there are certain ranges of friction coefficients where the object will always go unstable. The work is not final, but further time and research into this problem is warranted to see if there are some cases where the model will always go unstable. If that is indeed the case then a different model will be needed to model real world simulations.

6.1.6 What About Three Dimensions

All research in this paper was conducted in two dimensions. Further research into the problem should be conducted in three dimensions, to see if the same pattern occurs. Unfortunately moving the problem into three dimensions will significantly slow down the model even more, which will be a problem. Research into the three dimensional problem and instabilities there will probably take years due to the sheer computing power that will be needed to solve and debug the problem.

6.1.7 A Further Proof

In Section 5.2.1 it was shown that experimentally for the cases ran in these experiments in this thesis that the matrix in the LCP problem is a $P$ matrix. However, as we all know, experimental cases do not constitute a proof. Future research will attempt to see if there is a way to find out exactly what are the eigenvalues of the matrix $C^T A^{-1} C$ for the Jacobi problem, and for what conditions will the matrix $C^T A^{-1} C$ be a $P$ matrix, and therefore the LCP will have an unique solution.
6.2 Conclusion

In summary, it seems that the model for an elastic body under Coulomb friction has lots of work to be done still. Instabilities seem to occur in the model as long as the friction coefficient is high enough, and a high enough initial velocity is provided to the object in motion. Even under uniform initial forces, the object appears to move a non-uniform distance over time. Some of the non-uniform displacement seems to be due to errors in solving the iterative equations that need to be solved in the model. Yet some of the non-uniform displacement also seems to be due to the stress and strain terms amplifying the errors and causing them to propagate throughout the object. It appears that for small enough timesteps, the object stays stable for larger values of the friction coefficient, but preliminary results also suggest that for some values of the friction coefficient the object will always go unstable. Further research is needed in this field in order to see what is going on in these models better.
REFERENCES


