PHYSICS-BASED CONTACT USING THE COMPLEMENTARITY APPROACH FOR DISCRETE ELEMENT APPLICATIONS IN VEHICLE MOBILITY AND TERRAMECHANICS

By

Daniel J. Melanz

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy
(Mechanical Engineering)

at the
UNIVERSITY OF WISCONSIN–MADISON
2016

Date of final oral examination: 05/02/2016

The dissertation is approved by the following members of the Final Oral Committee:

Dan Negrut, Professor, Mechanical Engineering
Krishnan Suresh, Professor, Mechanical Engineering
Darryl Thelen, Professor, Mechanical Engineering
Eftychios Sifakis, Professor, Computer Science
William Likos, Professor, Geological Engineering

Version 2.0

To Hannah,

I love you more!

ACKNOWLEDGMENTS

I am extremely grateful for the guidance and encouragement from my advisor, Professor Dan Negrut. I am thankful for the friendship and assistance of my colleagues in the Simulation–Based Engineering Laboratory and at the U.S. Army TARDEC. Above all, I would like to thank my friends and family, especially my mom and dad – this work would not have been possible without your love and support.

PHYSICS-BASED CONTACT USING THE COMPLEMENTARITY APPROACH FOR DISCRETE ELEMENT APPLICATIONS IN VEHICLE MOBILITY AND TERRAMECHANICS

Page

Page

Page

APPENDICES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

ABSTRACT

In the context of soil dynamics, terramechanics models fall into three categories of increasing complexity: (i) empirically-based, (ii) continuum-based, or (iii) discrete-based approaches. Empirical methods for modeling wheel performance, like the one used by Wong and Reece, rely on the relationship between soil sinkage and resistance force to infer the normal stress under a wheel. Continuum methods assume matter to be homogeneous and continuous, making it difficult to model soil flow and separation. The discrete element method (DEM) represents soil as a collection of many discrete bodies, or elements, where each element is defined by its size, shape, position, velocity, and orientation. When elements collide, forces and torques are generated using explicit equations.

Despite this wide array of formulations, deformable terrain models that enhance the fidelity of present day vehicle and tire models have been lacking due to the complexity of soil behavior. Purely empirical terrain models are typically only used for go/no-go vehicle mobility assessment and have several drawbacks: the parameters can be sensitive to experimental testing procedures, they do not scale well to vehicles with small contact patches, they expose only a small number of model parameters, and they cannot capture 3D effects manifest at the interface between wheel/track and terrain. Continuum-based terramechanics models have been applied with only limited success for general purpose vehicle mobility simulations in off-road conditions since: (a) tire geometry is most often assumed to be simple and described in two-dimensions, which does not capture tread/lug geometry effects on tractive performance, and (b) soil flow effects are also generally ignored or dealt with in an ad-hoc manner. Despite its potential, DEM is currently considered prohibitively expensive due to the amount of data computation that it requires. Recent advances in computer hardware and numerical methods, however, make DEM a viable candidate for real world engineering problems. In particular, solving contact through complementarity requires a small number of model parameters, allows for integration at large step-sizes, and robustly handles the discontinuities in velocities.

This thesis details several enhancements to the complementarity method of contact for discrete element applications in terramechanics. This work is motivated by the degree of fidelity that the discrete element method lends to terramechanics modeling and the advantages that the complementarity formulation provides over alternate contact formulations. Specific enhancements focus on physical modeling and numerical methods, with analytical and experimental techniques used for validation. This basic research is ultimately used to solve a real-world, engineering application, specifically the study of military vehicles operating in off-road terrain conditions.

Chapter 1

Introduction

Military maneuvers, involving tactical formations and movements of wheeled and tracked vehicles across a landscape, provide the edge in combat for military units. Unfortunately, scientists and engineers, who most often design the vehicle-weapons systems capabilities used in combat operations, are generally not included in the tactical planning process and must design vehicles based on expected mobility challenges. It is difficult and expensive to evaluate a vehicle's performance during a majority of military maneuvers using physical experiments. Because of this, engineers and scientists are increasingly relying on modeling and simulation to perform these expected mobility maneuvers, such as operations in urban areas, mud, and slopes.

Understanding vehicle–terrain interaction, such as the interplay between the HMMWV and rocky ditch shown in Fig. 1.1, is of great importance to military applications. Significant soil deformation and transport, and the presence of properties such as soil packing and non-homogeneity present a significant challenge to offroad vehicles. Highly deformable soils offer little resistance to wheel sinkage and can only support light vehicles. There are several problems that can occur while traversing sand, such as the cage wheel-blocking problem, where the wheel becomes completely laden with soil [**?**]. On the other hand, weakened soil is the main factor which causes the wheel to slip, sink and remain standstill [**?**].

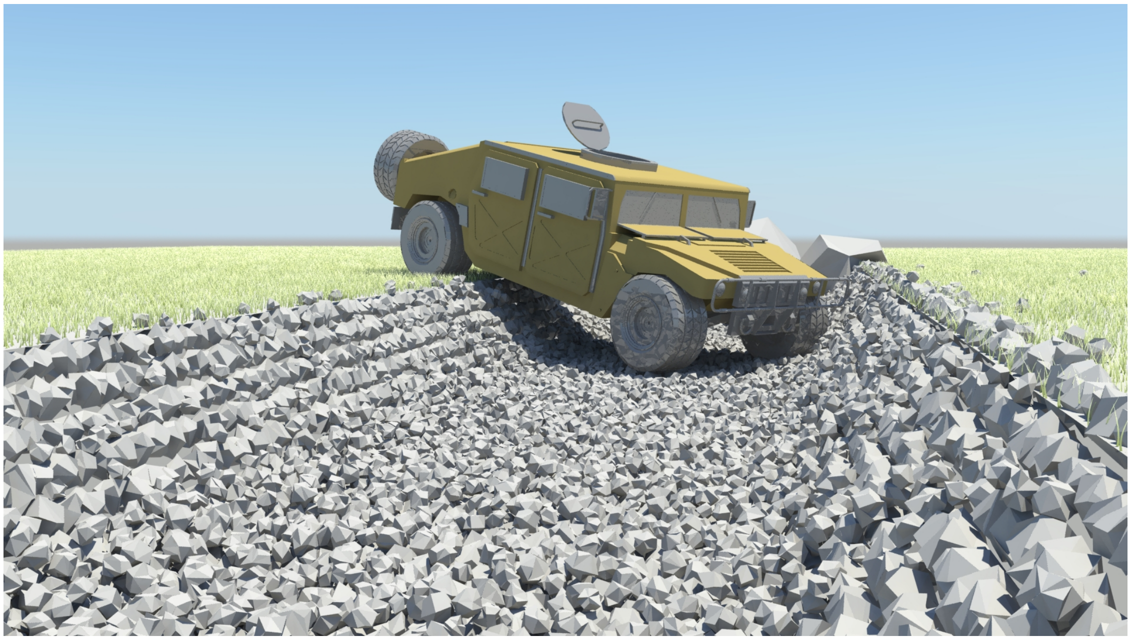


Figure 1.1: A HMMWV traversing a deformable terrain modeled using the discrete element method (DEM).

Lastly, maneuvering on non-flat terrain is an important capability of military vehicles [**?**]. Terrain with slopes exceeding 15° are considered steep slopes with vehicles typically being able to handle slopes up to 20° under favorable conditions. First studies of vehicle motion on slopes were published in the late 1960s [**?**]. Stability loss on rough slopes is more likely than on smooth slopes because the wheels, following the bumps and hollows of the rough ground, are bound to encounter even steeper slopes [**?**]. Some widely acknowledged causes of critical situations are: overturning due to dynamic effects, lack of inherent stability or external load, sliding resulting from excessive stress in the tire-ground contact area followed by a build-up of speed and eventual consequent overturn, and functional shortcomings of a vehicle (e.g. unsuitable engine, inadequate brakes, drivetrain) [**?**].

1.1 Motivation

Enhancing the complementarity method of frictional contact for discrete element applications in terramechanics is motivated by three factors: the high fidelity of the discrete element method for modeling terramechanics, described in §1.1.1, the advantages of the complementarity form of contact over alternate formulations, described in §1.1.2, and the shortcomings of current implementations of the complementarity form of contact, described in §1.1.3.

1.1.1 Fidelity of the discrete element method

Researchers do not currently have the ability to accurately (i) estimate vehicle operational parameters such as forces, torques, and sinkage that a wheel or track on a vehicle experiences, (ii) investigate the interaction that occurs between a wheel/track and soft soil, or (iii) simulate a ground vehicles ability to navigate complex off-road terrain. Deformable terrain models that complement the fidelity of present day vehicle and tire models have been lacking due to the complexity of soil behavior. Purely *empirical* terrain models, such as Bekker, Wong, and the NATO reference mobility model (NRMM) [**?**, **?**, **?**], typically used for go/no-go vehicle mobility assessment, have several drawbacks: the parameters can be sensitive to experimental testing procedures, they do not scale well to vehicles with small contact patches, they expose only a small number of model parameters, and they cannot capture the 3D effects at the interface between the wheel/track and terrain [**?**, **?**, **?**]. *Continuum*-based terramechanics models, such as [**?**], have been applied with limited success for general purpose vehicle mobility simulations in off-road conditions since: (a) tire geometry is most often assumed to be simple and described in two-dimensions, which does not capture tread/lug geometry effects on tractive performance, and (b) soil flow effects (e.g., bulldozing, slip sinkage, rutting) are also generally ignored or dealt with in an ad-hoc manner. The third approach, which is the focus of this thesis, draws on *discrete* methods which are currently prohibitively expensive due to the staggering amount of data computation that they require. At one end of the computer aided engineering (CAE) spectrum, there are commercial DEM codes that concentrate on dry-friction governed granular dynamics [**?**, **?**]. These codes can be used to investigate the dynamics of one to two million body systems. However, they cannot handle modeling and numerical solution challenges posed by highly nonlinear multibody systems such as robots, tractors, tracked vehicles, etc., whose dynamics are captured by systems of index-3 differential algebraic equations (DAEs) [**?**, **?**, **?**]. At the other end of the spectrum, these DAE problems are routinely solved by computational multibody dynamics commercial packages such as ADAMS, SimPack, or RecurDyn [?, ?, ?]. Yet these packages handle large frictional contact problems posed by discrete media dynamics poorly. For instance, a test was conducted first in 2007 [?] and repeated in 2013 [?] to gauge the scalability of a widely used commercial package. The conclusion was that a three second simulation of a dynamics problem with one million bodies would require tens of years of compute time in both the 2007 and 2013 versions of commercial solution. However, virtual prototyping in engineering requires analysis of complex vehicles operating with/on discrete systems with billions of bodies. After all, in one cubic meter of sand there are more than one billion elements. Driving over or moving around one cubic meter of sand cannot be simulated today due to the inability of the existing DEM and computational multibody dynamics solutions to handle the size and bridge the scales that manifest in the physics of interest. However, recent advances in computing technology provide a fresh opportunity to reconsider this approach. To carry out one million floating point operations in one second (1 MFLOP/s) in 1961 would have required the requisition of 17 million IBM-1620 computers. At $\$64$K apiece, when adjusted for inflation this would come out at half the 2013 US national debt. The price came down to around $\$1,000$ in 2000. Today, less than 10 cents of the value of a desktop would be used to yield 1 MFLOP/s. The packing of an increasing number of transistors per unit area, which is expected to continue according to Moore's law at least until 2021, demonstrates manifestly that the barrier to solving large-scale problems of relevance in science and engineering is lack of modeling techniques and solution methods that leverage advanced computing hardware. A back of the envelope calculation suggests that a problem with 1 billion bodies roughly requires memory on the order of 1 TB: 1 billion bodies times 8 bytes (double precision) times 100 variables per body (positions, velocities, accelerations, moments of inertia, bilateral constraints, etc.). In two years, 1 TB of memory and $5,000,000$ MFLOP/s; i.e., 5 trillion operations per second, is what an engineer will have access to on a high-end workstation tucked under the desk.

1.1.2 Advantages of contact through complementarity

The most common method of modeling contact is known as the penalty method. The penalty method of contact prevents bodies from passing through one another by the application of normal and tangential Hertzian style friction forces $\mathbf{F}_{n/t}$ [**?**, **?**, **?**, **?**] shown in Fig. 1.2. When elements collide, the reaction force is determined by the length and velocity of element-element overlap. Forces normal to the contact plane (along the unit vector between elements) are described by

$$F_n = k_n \delta n_{ij} - \gamma_n v_n \tag{1.1}$$

where δ is the overlap length of the two elements $\mathbf{n}_{ij}$ is the unit vector connecting the centers of the each element, and $\mathbf{v}_{n/t}$ is the normal component of the relative velocity of the two elements. The normal elastic spring coefficient k_n and viscoelastic damping coefficient γ_n are defined by

$$k_n = \frac{4}{3} Y_{\text{eff}} \sqrt{\frac{R_i R_j}{R_i + R_j} \delta} \tag{1.2a}$$

$$\gamma_n = -2\sqrt{\frac{5}{6}} \beta \sqrt{1.5 k_n \frac{M_i M_j}{M_i + M_j}} \tag{1.2b}$$

where R_i and M_i are the radius and mass of each element i. The effective Young's modulus Y_{eff} and the coefficient β are defined by

$$Y_{\text{eff}} = \frac{Y_i Y_j}{Y_i(1 - \nu_j^2) + Y_j(1 - \nu_i^2)} \tag{1.3a}$$

$$\beta = \frac{\ln e}{\sqrt{\ln^2 e + \pi^2}} \tag{1.3b}$$

where $Y_{i/j}$ is the Young's modulus of each element, $\nu_{i/j}$ is the Poisson ratio of each element and e is the coefficient of restitution.

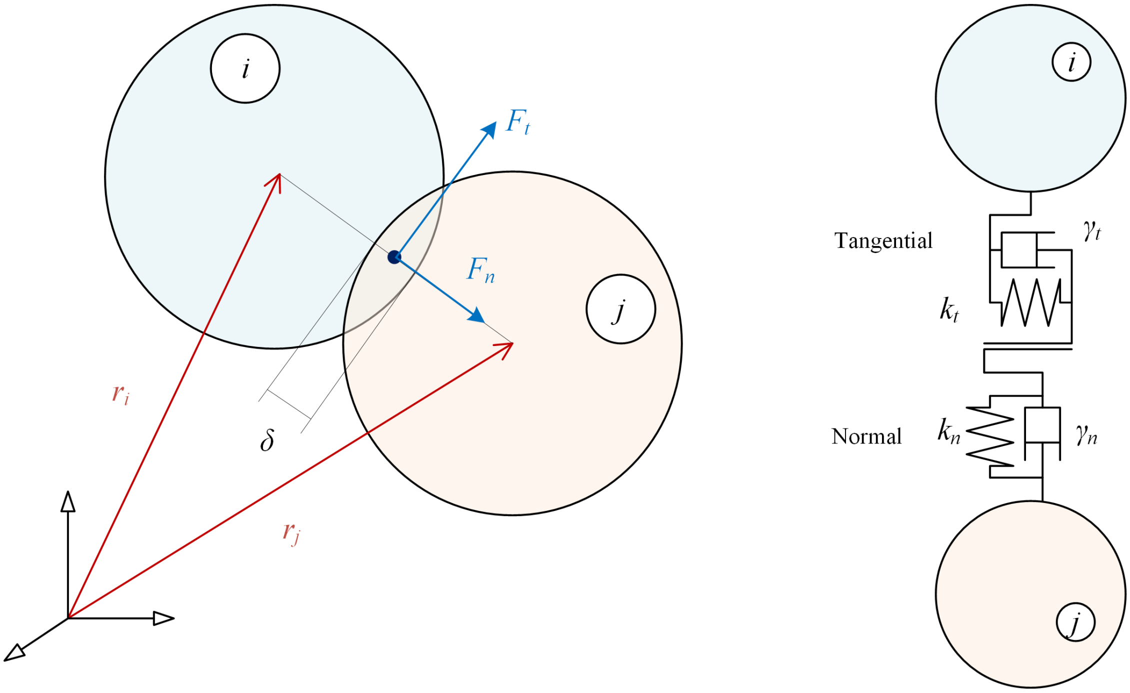


Figure 1.2: Penalty contact model for element-element force/torque interactions [**?**].

Similarly, forces tangential to the contact plane are described by

$$F_t = k_t \Delta s_t - \gamma_t v_t \tag{1.4}$$

where Δs_t is the tangential displacement vector between the two bodies for their entire contact duration and v_t is the tangential component of the relative velocity of the two elements. The tangential elastic spring coefficient k_t and viscoelastic damping coefficient γ_t are defined by

$$k_t = 8G_{\text{eff}}\sqrt{\frac{R_i R_j}{R_i + R_j}\delta} \tag{1.5a}$$

$$\gamma_t = -2\sqrt{\frac{5}{6}}\beta\sqrt{k_t\frac{M_i M_j}{M_i + M_j}} \tag{1.5b}$$

where G_{eff} is the effective shear modulus, defined by

$$G_{\text{eff}} = \frac{0.5Y_i Y_j}{Y_i(2+\nu_j)(1-\nu_j) + Y_j(2+\nu_i)(1-\nu_i)} \tag{1.6}$$

An upper limit exists for tangential forces through Coulomb criterion:

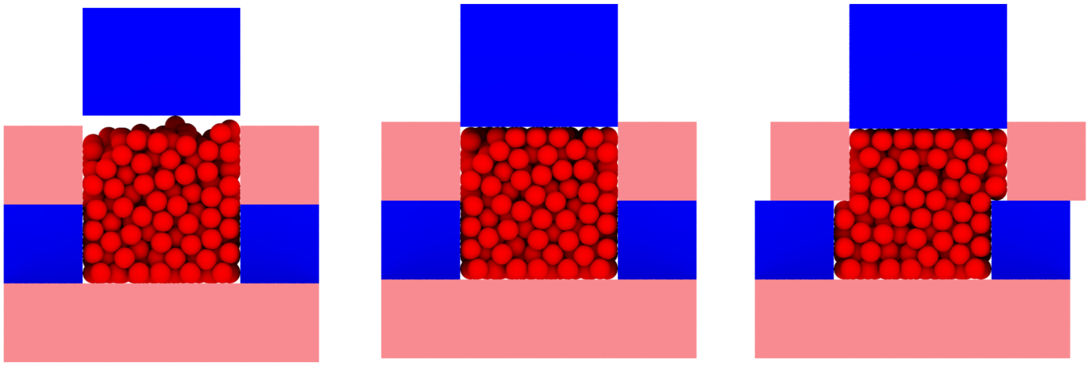
$$\text{if } |F_t| > \mu_c |F_n| \text{ then } |F_t| = \mu_c |F_n| \tag{1.7}$$

where μ_c is the static yield coefficient.

Unlike the penalty method, solving contact through complementarity requires a small number of model parameters, allows for integration at large step-sizes, and robustly handles the discontinuities in velocities. However, it calls for more involved mathematical instruments both in the equation formulation and equation solution stages. The concept of equations of motion are extended to include differential inclusions [**?**]. For frictional contact problems, the inclusion states that the frictional contact force lies somewhere inside the friction cone, with a value yet to be determined and controlled by the stick/slip state of the interaction between body and ground. In computational multibody dynamics the differential inclusion can be posed as a differential variational inequality (DVI) [**?**]. In its most general form, it assumes the form in [**?**, **?**, **?**], which has been proved to resolve Painleve's paradox [**?**].

1.1.3 Shortcomings of the current implementation

There are two shortcomings of the DVI implementation, namely (i) modeling of the physical situation of interest is not expressive enough and (ii) the numerical methods that are used to solve the underlying equations of motion are not efficient enough. These shortcomings present a major problem to the modeling of terramechanics and were recently highlighted by the simulation of the direct shear test. The goal of the direct shear investigation was to replicate an experimental direct shear test through a DEM simulation. The direct shear test, shown in Fig. 1.3, can be used to measure the shear strength properties of a soil, specifically the cohesion, angle of friction, and shear modulus. A sample of the soil is contained in between two rigid discs that are held in place by a shear box. The shear box is aligned under a load cell that applies a normal force to the soil. The combination of high normal and shearing pressures makes this test a simple but direct analog to the interaction of a wheel on granular terrain.



(1) Fill

(2) Compact

(3) Shear

Figure 1.3: The direct shear testing procedure.

The load cell is attached to a vertical translational joint that uses a linear variable differential transformer to measure displacement of the soil. The top of the shear box is clamped so that the lower half can be moved. The horizontal force required to displace the soil horizontally is measured by a dynamometer which is then transformed into the shear stress. The horizontal and vertical soil displacement is also output. The shear stress is plotted as a function of the horizontal, or shear, displacement to characterize the shearing behavior of the soil.

On the numerical side, the correct soil parameters are found by performing a direct shear simulation that is similar to the real experiment, using a nonlinear least-squares approach to determine the optimal, or best, soil parameters [**?**, **?**, **?**]. Soil parameter identification is made difficult by a combination of DEM's long computational time and the number of iterations that must be performed to determine the correct parameters. To verify that the complementarity and penalty contact models accurately model the micro-scale physics and emergent macro-scale properties of a simple granular material, Fig. 1.4 shows shear versus displacement curves obtained from both experimental [**?**] (left) and simulated (right) direct shear tests, performed under constant normal stresses of 3.1, 6.4, 12.5, and 24.2 kPa, on $5,000$ uniform glass beads. The shear box used in the experiment and simulations was $120 \times 120 \times 60$ mm (W × L × H) in size and had a shear rate of 1 mm/s. For the penalty method, the material properties of the spheres in the simulations were taken to be those corresponding to glass [**?**], for which the density is $2,550$ kg/m^3, the inter-particle friction coefficient is $\mu = 0.18$, Poisson's ratio is $\nu = 0.22$, and the elastic modulus is $E = 4 \times 10^{10}$ Pa, except that the elastic modulus was reduced by three orders of magnitude to $E = 4 \times 10^7$ Pa to ensure a stable simulation with a reasonable time integration step-size of $h = 10^{-5}$ seconds. For the complementarity method, a friction coefficient $\mu = 0.9$ was used with a step-size of $h = 10^{-3}$ seconds.

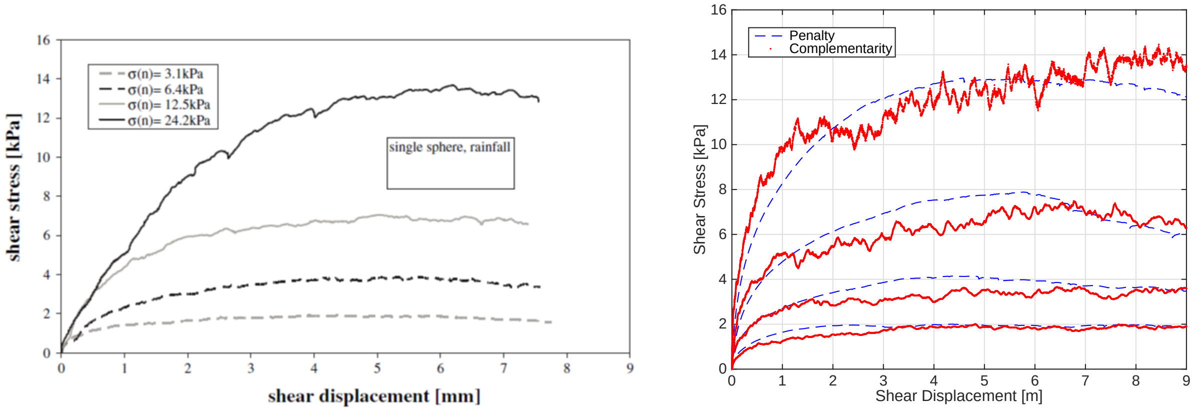


Figure 1.4: Direct shear test results for $5,000$ randomly packed uniform glass beads obtained by experiment [**?**] (left) and DEM simulations (right), under constant normal stresses of 3.1, 6.4, 12.5, and 24.2 kPa.

Figure 1.4 shows that the direct shear simulations on $5,000$ glass spheres do a fairly good job of matching the physical experiments for both the complementarity and penalty contact models. Although there is much less noise in the penalty case, the complementarity method required fewer parameters and could use a much larger time step than the penalty method. Using a larger time step generally results in shorter simulation times - in a similar experiment involving $1,000$ glass spheres, the complementary method was approximately $1.5\times$ faster than the penalty method.

1.2 Document overview

This thesis details several enhancements to the complementarity method of contact for discrete element applications in terramechanics. This work is motivated by the degree of fidelity that the discrete element method lends to terramechanics modeling and the advantages that the complementarity formulation provides over alternate contact formulations. Specific enhancements focus on physical modeling and numerical methods, with analytical and experimental techniques used for validation. This basic research is ultimately used to solve a real-world, engineering application, specifically the study of military vehicles over off-road terrain.

This document is organized as follows. Chapter 2 describes the study of soil dynamics, or terramechanics, and briefly covers the empirical, continuum, and discrete approaches for modeling terramechanics phenomena. Chapter 3 provides an overview of multibody dynamics, focusing on the complementarity form of handling friction and contact. Chapter 4 investigates several numerical methods used to determine the time evolution of large systems of rigid bodies that mutually interact through contact with friction. Chapter 5 focuses on an iterative refinement scheme, called anti-relaxation, to update the objective function of the underly complementarity model to yield a optimum that is equivalent to the solution of the original problem. Chapter 6 discusses how the complementarity formulation can be used to model frictional contact between flexible bodies formulated with the absolute nodal coordinate formulation (ANCF) to simulate large flexible multibody systems, such as a flexible tire. Chapter 7 reports on results obtained in the process of validating the solution methodology outlined in Chapter 3. Finally, Chapter 8 provides several cases of the complementarity form of contact being used to study full-scale vehicles driving over deformable terrain.

1.3 Specific contributions

The specific contributions of this work are summarized as follows:

- Applied contact through complementarity to discrete element simulations in large-scale mobility and terramechanics analyses:
    - Developed several case studies to highlight the potential of DEM with nonsmooth contact for large-scale mobility analysis [**?**]
    - Validated the complementarity approach for contact using standard terramechanics tests (direct shear, pressure-sinkage, and single wheel test) [**?**]

  - Proved the inaccuracy of traditional terramechanics techniques using uncertainty analysis [**?**, **?**, **?**]
  - Demonstrated the convergence of discrete element simulations for systems with over a million degrees of freedom using APGD and identified the inherent numerical challenges when solving a large-scale CCQO [**?**]
  - Characterized the effects of body shape and local friction coefficient on emergent behavior in terramechanics [**?**]

- Investigated new numerical methods for the differential variational inequality formulation which demonstrate improved convergence properties [**?**]:
  - Implemented PDIP to run in parallel with OpenMP or GPU programming
  - Extended and improved the SCIP method for frictional contact problems
  - Developed a consistent termination criteria for the CCQO and linked it to physical phenomena

- Improved the accuracy of the frictional contact solution in the differential variational inequality framework [**?**]:
  - Proved a mathematical artifact in the existing differential variational framework that resulted in non-physical behavior based on a case with an analytical solution
  - Demonstrated that the relaxation of the differential variational framework can be alleviated via an iterative refinement technique
  - Implemented the anti-relaxation via iterative refinement technique for large-scale discrete element simulations on the GPU

- Implemented an efficient model of the DVI-based frictional contact for flexible ANCF tires on deformable terrain [**?**]:
  - Developed a rigid-flexible multiphysics simulation engine on the GPU that relies on DVI-based contact for millions of degrees of freedom

- Validated rigid-flexible contact for simple scenarios with analytical solutions
- Demonstrated capabilities through simulation of a single flexible tire operating on granular terrain composed of over $100,000$ terrain bodies

Chapter 2

Terramechanics

In the context of soil dynamics, terramechanics models fall into three categories of increasing complexity: (i) empirically-based, (ii) continuum-based, or (iii) discrete-based approaches. Fig. 2.1 shows a hierarchy of these methods with examples of the modeling techniques.

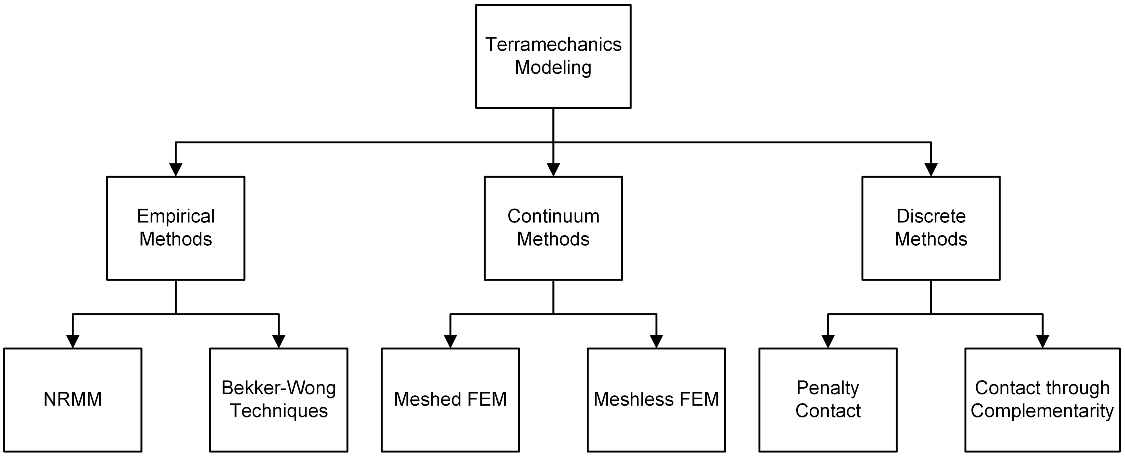


Figure 2.1: Examples of modeling techniques in terramechanics.

2.1 Empirical methods

Empirical methods for modeling wheel performance, like the ones used by Wong and Reece [**?**, **?**], rely on the relationship between soil sinkage and resistance force to infer the normal stress under a wheel. To predict the tractive force, the shearing strength of the soil is analyzed based on Coulomb's formula [**?**]. Methods of this class are ultimately based on experimentally determined soil parameters, whose inherent variability causes uncertainty in the determination of wheel performance. The rigid wheel free-body diagram, shown in Fig. 2.2, is used to model the interaction between the wheel and the soil. Using this model, the drawbar pull D, torque T, and sinkage z, can be estimated for a wheel of weight W, radius r, and wheel width b, travelling at a linear velocity v.

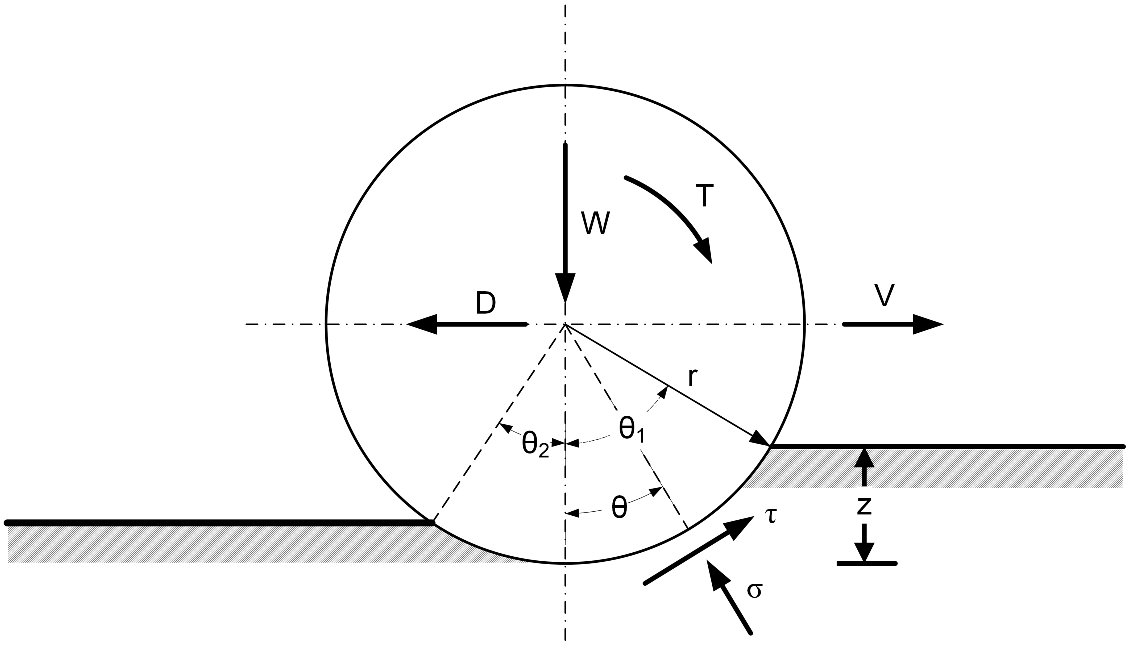


Figure 2.2: Forces, torques, and stresses experienced by a driven rigid wheel.

The sinkage of the wheel is typically converted into polar coordinates using the wheel hub as the origin. Once the limits of the contact patch, θ_1 and θ_2, between the wheel and the soil are determined, D and T can be calculated by integrating the radial and tangential stresses over the wheel.

2.2 Continuum methods

Continuum methods assume matter to be homogeneous and continuous, making it difficult to model soil flow and separation [**?**]. Some examples of terrains fitting the continuous assumption are mud, slurry, concrete and lava. The behavior of a continuum under the action of external forces is represented by a set of partial differential equations (PDE) with boundary conditions in the continuum field and a constitutive law capturing the continuum's material type. Producing an analytical solution of these partial differential equations is almost always impossible for practical continuum problems. Hence, numerical techniques are adopted to approximate the solution to these PDEs by discretizing the continuum domain using a mesh. A mesh can be defined as a way of connecting selected points in the space of the continuum in a predefined manner. In a finite difference method (FDM), the mesh is referred to as a grid; in the finite volume method (FVM), the meshes are called volumes or cells; and in finite element method (FEM), the meshes are called elements. Irrespective of the name used, these traditional methods use a predefined mesh in the problem domain. Alternatively, meshless approaches, such as the material point method (MPM) [**?**, **?**, **?**] have been shown to be quite promising for problems that require large deformations and plasticity. Unlike traditional FEM, MPM does not require re-meshing of the simulation domain or remapping of the state variables related to the constitutive model.

2.3 Discrete element methods

The discrete element method (DEM) represents soil as a collection of many three-dimensional bodies, called “elements”, where each element is defined by its size, shape, position, velocity, and orientation [**?**]. When elements collide forces and torques are generated using explicit equations. By modeling soil using individual bodies, DEM allows for significant soil deformation and transport, and the modification of properties such as soil packing structure and non-homogeneity [**?**]. There are multiple formulations of DEM, classified by the method in which contact and impact is handled when two bodies collide. The two main methods are: (i) the penalty approach, and (ii) the complementarity approach. There are advantages and disadvantages to both approaches, the penalty approach being more “tunable”, with many choices available for force models and parameter values. The penalty approach must always allow some amount of penetration between bodies, however, and may require very small time steps. On the other hand, the complementarity approach has few parameters that must be specified and enforces non-penetration, resulting in larger time steps. Unfortunately, although the complementarity approach is easily modeled, solving it numerically presents a challenge.

Chapter 3

Multibody dynamics with physics-based contact

This chapter introduces frictional multibody dynamics dynamics and describes the formulation for handling frictional contact in the numerical simulation. Nonsmooth rigid multibody dynamics (NRMD) consists of predicting the position and velocity evolution of a group of rigid particles that are subject to non-interpenetration, collision, adhesion, and dry friction constraints and to possibly global forces (such as electrostatic and gravitational forces). The dynamics of such a group of particles is nonsmooth because of the intermittent nature of non-interpenetration, collision, and adhesion constraints and because of the nonsmooth nature of the dry friction constraints at stick-slip transitions. Using Coulomb's Law of Friction, the dynamics of such a nonsmooth rigid multibody system can be resolved by simultaneously solving a linear complementarity problem (LCP) that links the normal contact impulses to the distances between bodies and a quadratic minimization problem that links the normal and tangential contact impulses via conic constraints. Solving this coupled system, or differential variational inequality (DVI), has proven to be quite difficult although polyhedral linearizations of the minimization problem have provided a reasonable approximation. To circumvent the difficulties posed by increasing complexity of classical LCP solvers and the increased size and inaccuracy introduced by polyhedral approximation, however, a novel solution method was developed based on a fixed-point iteration with projection on a convex set that can directly solve large cone complementarity problems with low computational overhead. This solution method works by adding a relaxation term to transform the original problem into a cone complementarity problem (CCP). By considering the Karush-Kuhn-Tucker (KKT) conditions, the CCP is further transformed into a cone-constrained quadratic optimization problem (CCQO) for which several iterative solution methods exist. In the DVI method, a CCQO must be solved at each time step of the simulation, where the unknowns are the normal and frictional contact forces between interacting bodies. The time-stepping scheme was proven to converge in a measure differential inclusion sense, to the solution of the original continuous-time DVI. In this chapter and the remainder of the document, scalars are written in italics, while matrix and vector terms are specified with bold symbols.

3.1 Posing the equations of motion

The time-evolution of a collection of n_b rigid bodies interacting through friction and contact is described herein using Cartesian coordinates associated with each body j, where $1 \leq j \leq n_b$. The array of generalized coordinates $\mathbf{q} = [\mathbf{r}_1^T, \epsilon_1^T, \dots, \mathbf{r}_{n_b}^T, \epsilon_{n_b}^T]^T \in \mathbf{R}^{7n_b}$, and its time derivative $\dot{\mathbf{q}} = [\dot{\mathbf{r}}_1^T, \dot{\epsilon}_1^T, \dots, \dot{\mathbf{r}}_{n_b}^T, \dot{\epsilon}_{n_b}^T]^T \in \mathbf{R}^{7n_b}$, are used to represent the state of the system, where for body j, $\mathbf{r}_j$ and ϵ_j are the absolute position of the center of mass and the body orientation Euler parameters, respectively. The derivative of the Euler parameters $\dot{\epsilon}$ can be replaced with a different set of unknowns; i.e., the angular velocity in local coordinates $\bar{\omega}$, formulating the generalized velocity $\mathbf{v} = [\dot{\mathbf{r}}_1^T, \bar{\omega}_1^T, \dots, \dot{\mathbf{r}}_{n_b}^T, \bar{\omega}_{n_b}^T]^T$, which can be mapped to $\dot{\mathbf{q}}$ via [**?**],

$$\dot{\mathbf{q}} = \mathbf{T}(\mathbf{q})\mathbf{v}. \tag{3.1}$$

The equation of motion is expressed in matrix form,

$$\mathbf{M}\dot{\mathbf{v}} = \mathbf{f}(t, \mathbf{q}, \mathbf{v}), \tag{3.2}$$

where $\mathbf{M} \in \mathbf{R}^{6n_b \times 6n_b}$ is the generalized mass matrix, and $\mathbf{f}(t, \mathbf{q}, \mathbf{v})$ is the generalized force applied to the system [**?**, **?**].

The presence of bilateral constraints such as spherical joints, revolute joints, translational joints, etc., restricts the relative motion of two rigid bodies. Mathematically, a set of holonomic algebraic equations are induced by the presence of each of these physical joints,

$$i \in \mathcal{B}: \quad \mathbf{g}_i(\mathbf{q}, t) = 0, \tag{3.3}$$

where $\mathcal{B}$ is the set of all bilateral constraints. The presence of the physical joints also induce kinematic constraints at the velocity level obtained by taking a time derivative of the position kinematic constraints of Eq. 3.3: $\dot{\mathbf{g}}_i(\mathbf{q}, t) = \vec{G}_i(\mathbf{q})\mathbf{v} + \frac{\partial \mathbf{g}_i}{\partial t} = 0$. Enforcing these constraints calls for the presence of a constraint reaction force $\vec{G}_i^T(\mathbf{q})\hat{\gamma}_{i,b}$, which is added to the right-hand side of the equations of motion in Eq. 3.2 and depends on an unknown Lagrange multiplier $\hat{\gamma}_{i,b}$ [**?**, **?**].

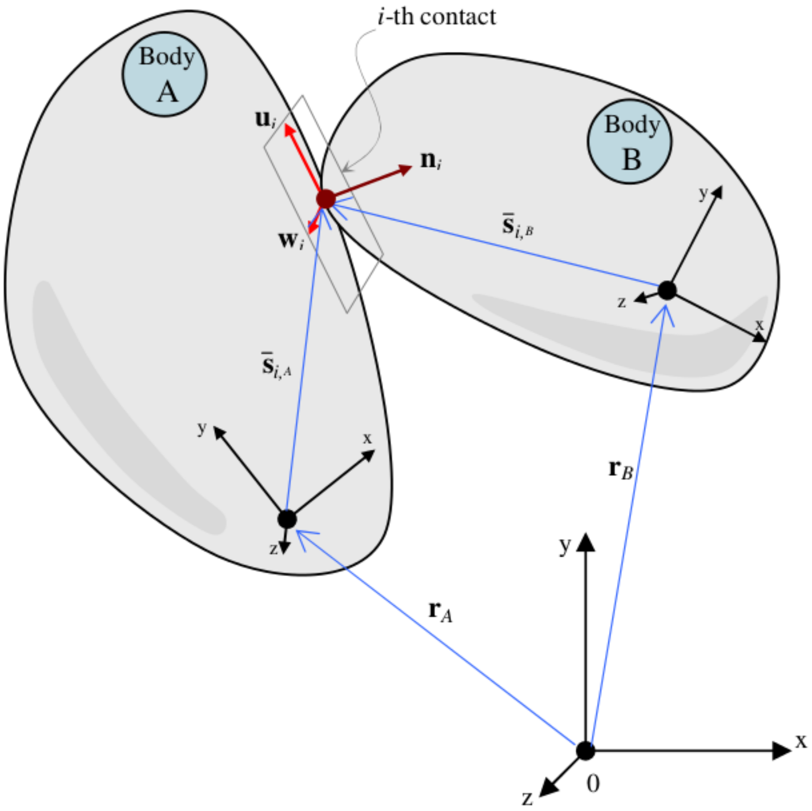


Figure 3.1: The i^{th} contact between two bodies A and B.

For two bodies A and B in contact, $1 \leq A < B \leq n_b$, see Figure 3.1, let $\bar{\mathbf{s}}_{i,A}$ and $\bar{\mathbf{s}}_{i,B}$ be the location of the contact point with respect to the reference frame of body A and B, respectively. Since we are concerned with rigid bodies, $A \neq B$. Assuming that the bodies in contact are defined by smooth boundaries, let $\mathbf{n}_i$ be the global unit vector denoting the normal direction at the contact points, and $\mathbf{u}_i$ and $\mathbf{w}_i$ be two unit vectors that span the contact plane at the point of contact. By convention, $\mathbf{n}_i$ points towards the interior of B, and $\{\mathbf{n}_i, \mathbf{u}_i, \mathbf{w}_i\}$ form a right-hand reference frame. The contact force, $\mathbf{F}_i$, associated with contact i can be decomposed into normal and tangential/frictional components, $\mathbf{F}_{i,N}$ and $\mathbf{F}_{i,T}$, respectively, where $\mathbf{F}_{i,N} = \hat{\gamma}_{i,n}\mathbf{n}_i$

and $\mathbf{F}_{i,T} = \hat{\gamma}_{i,u}\mathbf{u}_i + \hat{\gamma}_{i,w}\mathbf{w}_i$. Here, $\hat{\gamma}_{i,n}$, $\hat{\gamma}_{i,u}$ and $\hat{\gamma}_{i,w}$ are the magnitude of the force in each direction, and, for as far as body B is concerned, $\hat{\gamma}_{i,n} \geq 0$. That is, if body B comes in contact with any other body, a normal force will act on B to prevent penetration. This defines a unilateral kinematic constraint: body B can move away from body A, but it cannot move through it.This unilateral constraint is mathematically posed via the gap function $\phi(\mathbf{q})$ and involves the normal force. Specifically, when two bodies are not in contact, no normal contact force exists; when the bodies are in contact, the contact force is non-negative. This statement is captured in the following complementarity condition,

$$0 \leq \phi_i(\mathbf{q}) \perp \hat{\gamma}_{i,n} \geq 0 \quad \forall i \in \mathcal{A}(\mathbf{q}, \delta), \tag{3.4}$$

where $\mathcal{A}(\mathbf{q}, \delta)$ denotes the index set of all pairs of bodies which given a generalized position $\mathbf{q}$ are within a distance less than or equal to δ. We take $\delta > 0$ to also include bodies which might come in contact in the immediate future; i.e., during the next time step. The friction force is tied to the value of the normal force via the Coulomb friction model. Using a maximum dissipation principle [**?**], the Coulomb friction model is posed as the solution of an optimization problem

$$(\hat{\gamma}_{i,u}, \hat{\gamma}_{i,w}) = \underset{\sqrt{\tilde{\gamma}_{i,u}^2 + \tilde{\gamma}_{i,w}^2} \leq \mu_i \hat{\gamma}_{i,n}}{\arg\min} \mathbf{v}_{i,T}^T \left(\tilde{\gamma}_{i,u}\mathbf{u}_i + \tilde{\gamma}_{i,w}\mathbf{w}_i \right), \tag{3.5}$$

which provides for contact i the components $\hat{\gamma}_{i,u}$ and $\hat{\gamma}_{i,w}$ of the friction force given a friction coefficient μ_i, the relative tangential velocity $\mathbf{v}_{i,T}$ of the two bodies at the contact point, and the normal force at the contact point $\hat{\gamma}_{i,n}$.

A transformation matrix $\mathbf{A}_i = [\mathbf{n}_i, \mathbf{u}_i, \mathbf{w}_i] \in \mathbf{R}^{3\times3}$ is used for contact i to express the frictional contact force in the global frame. Additionally, a projection matrix $\mathbf{D}_i \in \mathbf{R}^{6nb\times3}$,

$$\mathbf{D}_i = [\mathbf{0}, \ldots, \mathbf{0}, -\mathbf{A}_i^T, -\mathbf{A}_i^T\mathbf{A}_A\tilde{\bar{\mathbf{s}}}_{i,A}, \mathbf{0}, \ldots, \mathbf{0}, \mathbf{A}_i^T, \mathbf{A}_i^T\mathbf{A}_B\tilde{\bar{\mathbf{s}}}_{i,B}, \mathbf{0}, \ldots, \mathbf{0}]^T, \tag{3.6}$$

is introduced to express the ensuing generalized frictional contact force. Here $\mathbf{A}_A$ and $\mathbf{A}_B$ are the body A and B rotational matrices, and a tilde over a vector denotes its skew-symmetric matrix [**?**]. The columns of $\mathbf{D}_i$ are denoted by $\mathbf{D}_{i,n}$, $\mathbf{D}_{i,u}$ and $\mathbf{D}_{i,w}$, therefore, $\mathbf{D}_i = [\mathbf{D}_{i,n}, \mathbf{D}_{i,u}, \mathbf{D}_{i,w}]$.

At this point, a DVI can be posed to capture the dynamics of the rigid body in the presence of friction and contact, see for instance [**?**, **?**]:

$$
\begin{aligned}
\dot{\mathbf{q}} &= \mathbf{T}(\mathbf{q})\mathbf{v} && \text{(3.7a)}\\
\mathbf{M}\dot{\mathbf{v}} &= \mathbf{f}(t,\mathbf{q},\mathbf{v}) + \sum_{i\in\mathcal{B}} \mathbf{G}_i^T \hat{\gamma}_{i,b} + \sum_{i\in\mathcal{A}} \mathbf{D}_i \hat{\gamma}_i && \text{(3.7b)}\\
i\in\mathcal{B} &: \mathbf{g}_i(\mathbf{q},t) = 0 && \text{(3.7c)}\\
i\in\mathcal{A} &: 0 \le \phi_i(\mathbf{q}) \perp \hat{\gamma}_{i,n} \ge 0 && \text{(3.7d)}\\
(\hat{\gamma}_{i,u}, \hat{\gamma}_{i,w}) &= \underset{\sqrt{\tilde{\gamma}_{i,u}^2+\tilde{\gamma}_{i,w}^2} \le \mu_i \hat{\gamma}_{i,n}}{\arg\min} \mathbf{v}_i^T \left(\tilde{\gamma}_{i,u}\mathbf{D}_{i,u} + \tilde{\gamma}_{i,w}\mathbf{D}_{i,w}\right) && \text{(3.7e)}
\end{aligned}
$$

The right-hand side of the momentum balance equation (Eq. 3.7b) embeds both the bilateral constraint, $\sum_{i\in\mathcal{B}} \mathbf{G}_i^T \hat{\gamma}_{i,b}$, and frictional contact forces, $\sum_{i\in\mathcal{A}} \mathbf{D}_i \hat{\gamma}_i$, where $\hat{\gamma}_i = [\hat{\gamma}_{i,n}, \hat{\gamma}_{i,u}, \hat{\gamma}_{i,w}]^T$. The differential attribute of the DVI problem stems from Eqs. 3.7a and 3.7b. Its variational attribute is associated with the minimization component in Eq. 3.7e. The inequality aspect goes back to Eq. 3.7d. The problem has also an algebraic attribute that goes back to the set of nonlinear algebraic equations that capture the bilateral constraints. This set of kinematic constraints, which turns an ordinary differential equations problem into an index-3 differential algebraic equations problem [**?**], is important in the economy of the numerical solution yet typically does not get represented in the name DVI associated with the problem in Eq. 3.7.

3.2 The time-stepping scheme

The discretization of the DVI in Eq. 3.7 is carried out at time $t^{(l)}$ using a step size h to yield [**?**]

$$\begin{aligned}
\mathbf{M}(\mathbf{v}^{(l+1)} - \mathbf{v}^{(l)}) &= h\mathbf{f}(t^{(l)}, \mathbf{q}^{(l)}, \mathbf{v}^{(l)}) + \sum_{i\in\mathcal{B}} \mathbf{G}_i^T \gamma_{i,b} + \sum_{i\in\mathcal{A}} \mathbf{D}_i \gamma_i && \text{(3.8a)}\\
i \in \mathcal{B} \quad &: \quad \frac{1}{h}\mathbf{g}_i(\mathbf{q}^{(l)}, t^{(l)}) + \mathbf{G}_i \mathbf{v}^{(l+1)} + \frac{\partial \mathbf{g}_i}{\partial t}|_{t^{(l)}} = 0 && \text{(3.8b)}\\
i \in \mathcal{A} \quad &: \quad 0 \le \gamma_{i,n} \perp \frac{1}{h}\phi_i(\mathbf{q}^{(l)}) + \Phi_i(\mathbf{q}^{(l)})\mathbf{v}^{(l+1)} \ge 0 && \text{(3.8c)}\\
(\gamma_{i,u}, \gamma_{i,w}) &= \underset{\sqrt{\bar{\gamma}_{i,u}^2 + \bar{\gamma}_{i,w}^2} \le \mu_i \gamma_{i,n}}{\arg\min} \quad \mathbf{v}_i^T \left(\bar{\gamma}_{i,u}\mathbf{D}_{i,u} + \bar{\gamma}_{i,w}\mathbf{D}_{i,w}\right) && \text{(3.8d)}\\
\mathbf{q}^{(l+1)} &= \mathbf{q}^{(l)} + h\mathbf{T}(\mathbf{q}^{(l)})\mathbf{v}^{(l+1)} \,. && \text{(3.8e)}
\end{aligned}$$

The contact force impulse, $\gamma_{i,\{n,u,w\}} \equiv h\hat{\gamma}_{i,\{n,u,w\}}$, and the contact force impulse triplets, $\gamma_i \equiv [\gamma_{i,n}, \gamma_{i,u}, \gamma_{i,w}]$, are evaluated at $t^{(l+1)}$; the quantities $\mathbf{D}$, $\mathbf{G}$ are evaluated in the configuration at $t^{(l)}$. Eq. (3.8c) relies on a linearization of the unilateral constraint $\Phi_i(\mathbf{q}^{(l+1)})$. Indeed, given $\mathbf{q}^{(l+1)} = \mathbf{q}^{(l)} + h\mathbf{v}^{(l+1)}$, $\phi_i(\mathbf{q}^{(l+1)}) \approx \phi_i(\mathbf{q}^{(l)}) + \Phi_i(\mathbf{q}^{(l)})h\mathbf{v}^{(l+1)} \ge 0$. This leaves the velocity $\mathbf{v}^{(l+1)}$ as the sole variable that needs to adjust at the right of the $\perp$ symbol to meet the complementarity condition.

Eqs. 3.8a through 3.8d combine to form a nonlinear optimization problem with equality and complementarity constraints, which can be solved by linearizing the friction cone and turning it into a multifaceted pyramid [**?**, **?**]. Herein, an alternative method is pursued whereby a *relaxation* of the complementarity constraints of Eq. 3.8c is considered [**?**]:

$$0 \le \gamma_{i,n} \perp \frac{1}{h}\phi_i(\mathbf{q}^{(l)}) + \Phi_i(\mathbf{q}^{(l)})\mathbf{v}^{(l+1)} - \mu_i\sqrt{(\mathbf{D}_{i,u}^T\mathbf{v}^{(l+1)})^2 + (\mathbf{D}_{i,w}^T\mathbf{v}^{(l+1)})^2} \ge 0. \qquad (3.9)$$

The solution of the relaxed problem approaches the solution of the original problem as $h \to 0$ [**?**]. The net effect of the relaxation is that the original highly nonlinear problem can be equivalently posed as a convex problem, more precisely a CCP, of the form [?]

$$\text{Find } \gamma_i^{(l+1)}, \text{ for } i = 1, \ldots, N_c \tag{3.10a}$$

$$\text{such that } \Upsilon_i \ni \gamma_i^{(l+1)} \perp - \left(\mathbf{N}\gamma^{(l+1)} + \mathbf{r}\right)_i \in \Upsilon_i^\circ \tag{3.10b}$$

$$\text{where } \Upsilon_i = \{[x, y, z]^T \in \mathbb{R}^3 | \sqrt{y^2 + z^2} \leq \mu_i x\} \tag{3.10c}$$

$$\text{and } \Upsilon_i^\circ = \{[x, y, z]^T \in \mathbb{R}^3 | x \leq -\mu_i \sqrt{y^2 + z^2}\}, \tag{3.10d}$$

where γ_i is the triplet of multipliers associated with contact i, Υ_i is the friction cone of contact i, and $\gamma \equiv \left[\gamma_1^T, \gamma_2^T, \ldots, \gamma_{N_c}^T\right]^T$. The positive semi-definite matrix $\mathbf{N}$ and vector $\mathbf{r}$ defined as

$$\mathbf{N} = \mathbf{D}^T \mathbf{M}^{-1} \mathbf{D}, \tag{3.11a}$$

$$\mathbf{r} = \mathbf{b} + \mathbf{D}^T \mathbf{M}^{-1} \mathbf{k}, \tag{3.11b}$$

where $\mathbf{b}^T = \left[\mathbf{b}_1^T, \ldots, \mathbf{b}_{N_c}^T\right]$ with $\mathbf{b}_i = \left[\frac{1}{h}\phi_i, 0, 0\right]^T \in \mathbb{R}^3$, and $\mathbf{k} = \mathbf{M}\mathbf{v}^{(l)} + h\mathbf{f}\left(t^{(l)}, \mathbf{q}^{(l)}, \mathbf{v}^{(l)}\right)$.

Solving the CCP of Eq. 3.10b is equivalent to solving a CCQO problem [?]. Given that the problems of interest have tens of bilateral constraints and thousands to millions of unilateral constraints, in what follows the bilateral constraints are ignored. In this case, the CCQO problem takes the form

$$\min \mathbf{q}(\gamma) = \frac{1}{2}\gamma^T \mathbf{N}\gamma + \mathbf{r}^T\gamma \tag{3.12a}$$

$$\text{subject to } \gamma_i \in \Upsilon_i \text{ for } i = 1, 2, \ldots, N_c, \tag{3.12b}$$

The relaxation of the complementarity condition in Eq. 3.9 has two consequences. On the upside, it leads in Eq. 3.12 to a problem whose solution can draw on a wider spectrum of methods. On the downside, the relaxation introduces artifacts when the step size h is large and/or the relative sliding velocity between the two bodies in contact is large and/or the friction coefficient at the interface is large. However, an anti-relaxation post-processing step can be invoked that at a slight increase in computational cost can eliminate the artifacts [?]. There are in the literature approaches that do not fall back on a relaxation, see for instance [?, ?, ?]. The approach embraced herein has one distinct advantage: a solution of the discretized problem exists, and it is unique in velocities and therefore positions [?]. This is important given that the convex optimization problem in Eq. 3.12, owing to the positive semidefinite attribute of $\mathbf{N}$, does not have a unique solution in the frictional contact impulses γ. The lack of uniqueness in the impulses is not specific to the solution embraced herein but rather a feature of the rigid body model when used in combination with the Coulomb dry friction model [**?**, **?**].

3.3 Overall solution methodology and software infrastructure

In the adopted contact approach, the solution is produced at a sequence of time steps $t^{(0)} < t^{(1)} < \ldots < t^{(N)} = T_{final}$. Currently, the integration step size is constant; i.e., $h = t_l - t_{l-1}$ is the same for any $1 \leq n \leq N$. Note that a set of initial conditions is provided at the beginning of the analysis; i.e., at time $t = t^{(0)}$. The solution is then advanced from $t^{(l)}$ to $t^{(l+1)}$ as follows. The optimization problem posed in Eq. 3.12 is solved to recover the set of Lagrange multipliers γ that are subsequently used in Eq. 3.8a to recover the new set of generalized velocities $\mathbf{v}^{(l+1)}$. Indeed, note that in the latter equation, provided γ is available, the velocities $\mathbf{v}^{(l+1)}$ are computed by multiplying from the left by $\boldsymbol{M}$. This matrix is constant and block diagonal. Once the velocity $\mathbf{v}^{(l+1)}$ is available, Eq. 3.8e is used to compute the new position and orientation of each part in the collection of bodies that make up the system of interest. It becomes manifest that the critical stage of the solution methodology is solving the optimization of Eq. 3.12, a process performed at each time step t_l. To that end, we draw on a first-order method called the accelerated projected gradient descent (APGD) [**?**], the algorithm of which can be found in §A.3. More than 80% of the entire simulation time is spent in setting up the optimization problem (Eq. 3.12) and solving it via APGD.

This numerical solution strategy is implemented in an open source analysis tool called `Chrono` . `Chrono`, a multibody dynamics engine released under a BSD3 license [**?**], produces the matrix $\boldsymbol{N}$ and vector $\mathbf{r}$ at each time step, an operation that employs a collision detection task. For instance, in a granular material analysis that monitors the dynamics of $10,000$ bodies, `Chrono` performs a collision detection at each time step that would find approximately $40,000$ collision events. Given that each contact event has a set of three Lagrange multipliers associated with it, `Chrono` assembles this information into the optimization problem in Eq. 3.12, which calls for finding the minimum of a cost function $\mathbf{q}\left(\gamma\right)$ that depends on approximately $120,000$ variables. Owing to the complementarity-based formulation adopted herein, the analysis proceeds at a relatively large step size; i.e., $h \approx 0.001$ s. This translates in performing $1,000$ collision detections and solving $1,000$ optimization problems for one second of dynamics analysis. To speed up the solution process, `Chrono` resorts to parallel computing in each stage of the analysis [**?**].

Chapter 4

Numerical solution methods for complementarity approaches

The purpose of this chapter is to investigate several numerical methods used to determine the time evolution of large systems of rigid bodies that mutually interact through contact with friction. The contact is modeled through a complementarity condition and the friction is posed as a variational problem. Upon discretization in time, the complementarity conditions enforced at the velocity level are relaxed to obtain a CCP. The solution of the CCP is found by minimizing an equivalent quadratic optimization problem with conic constraints. This contribution investigates five approaches – three first-order ones and two second-order ones, to solve this constrained optimization problem. The first-order methods, which only use gradient information, are: projected Jacobi (PJ), projected Gauss-Seidel (PGS), and accelerated projected gradient descent (APGD). The second-order methods are a symmetric cone interior point (SCIP) method and a primal-dual interior point (PDIP) method, both of which rely on a Newton step to identify the descent direction and a line search to compute the step size. All five methods draw on parallel computing on graphics processing unit (GPU) cards; the Newton step employs a sparse parallel GPU solver. Three types of numerical experiments, filling, drafting, and compacting, are carried out to evaluate the performance of the five solution strategies in terms of convergence rate, accuracy, and computational cost.

4.1 Optimization methods considered

The bottleneck of the solution embraced is the computation of the frictional contact impulses γ in Eq. 3.12. Three first-order methods are considered for finding γ: PJ, PGS, and APGD. The first two and variants thereof have been used extensively in the literature [**?**, **?**, **?**, **?**, **?**, **?**, **?**]. APGD has been introduced recently [**?**]. PJ, PGS, and APGD will be compared with two recently introduced second-order methods: PDIP [**?**] and SCIP [**?**].

4.1.1 First-order methods: Jacobi, Gauss-Seidel, and Nesterov

The DVI *modeling* approach leads to different *numerical* solutions based on the strategy adopted to discretize Eq. 3.7 [**?**, **?**]. Regardless of how Eq. 3.7 was discretized though, one common thread of numerous approaches is that subsequently they rely upon PJ and PGS [**?**, **?**, **?**, **?**, **?**, **?**]. The solution approach embraced herein is no exception – the solution of Eq. 3.12 originally drew on the PJ or PGS algorithms [**?**, **?**]. PGS was preferred owing to faster convergence. PJ was used in scenarios where parallel computing came into play. The execution flow for PJ and PGS is provided in Appendices A.1 and A.2 with additional details in [**?**]. A GPU implementation of PJ is discussed in [**?**, **?**].

The APGD approach to solve Eq. 3.12 is relatively recent [**?**], as is Nesterov's method, which was first introduced in 1983 [**?**]. The latter represented a step beyond traditional gradient descent methods. It used a "momentum" in the definition of the search direction as a quantitative way to account for the qualitative observation that the search direction should depend on past directions. Instead of taking the search direction to be opposite of the gradient, Nesterov's method used a weighted combination of the current and past gradient directions. The particular Nesterov method that serves as the cornerstone of the APGD algorithm used to solve the problem in Eq. 3.12 is described in detail in [**?**, **?**], along with aspects concerning its extension to the constrained case, the introduction of acceleration, a fall-back strategy, and the selection of the descent step. The APGD algorithm is outlined in Appendix A.3.

The projection operator $\Pi_{\mathcal{K}}$, which projects a vector onto the Cartesian product of the cones Υ_i, is used to maintain feasibility throughout the first-order iterative algorithms. In this case, a closed form expression can be written for the projection as follows:

$$\Pi_{\Upsilon_i}\left(\gamma_i\right) = \begin{cases} \gamma_i & \text{if } \gamma_i \in \Upsilon_i \\ \mathbf{0} & \text{if } \gamma_i \in \Upsilon_i^\circ \\ \left[\frac{\gamma_{i,n}+\mu_i\gamma_{i,t}}{\mu_i^2+1}, \gamma_{i,u}\frac{\mu_i\hat{\gamma}_{i,n}}{\sqrt{\gamma_{i,u}^2+\gamma_{i,w}^2}}, \gamma_{i,w}\frac{\mu_i\hat{\gamma}_{i,n}}{\sqrt{\gamma_{i,u}^2+\gamma_{i,w}^2}}\right]^T & \text{if } \gamma_i \notin \left(\Upsilon_i \cup \Upsilon_i^\circ\right) \end{cases} \tag{4.1}$$

4.1.2 Second-order methods

Interior point methods can solve convex optimization problems that include inequality constraints:

$$\min f_0\left(\mathbf{x}\right) \tag{4.2a}$$

$$\text{subject to } f_i\left(\mathbf{x}\right) \leq 0, \quad i = 1, \ldots, m\ . \tag{4.2b}$$

In Eq. 4.2, $f_0\left(\mathbf{x}\right), \ldots, f_m\left(\mathbf{x}\right) : \mathbb{R}^n \to \mathbb{R}$ are assumed convex and twice continuously differentiable. The KKT conditions are expressed as

$$f_i\left(\mathbf{x}^\star\right) \leq 0, \quad i = 1, \ldots, m \tag{4.3a}$$

$$\lambda_i^\star \geq 0, \quad i = 1, \ldots, m \tag{4.3b}$$

$$\nabla f_0\left(\mathbf{x}^\star\right) + \sum_{i=1}^{m} \lambda_i^\star \nabla f_i\left(\mathbf{x}^\star\right) = 0 \tag{4.3c}$$

$$\lambda_i^\star f_i\left(\mathbf{x}^\star\right) = 0, \quad i = 1, \ldots, m \tag{4.3d}$$

where $\mathbf{x}^\star \in \mathbb{R}^n$ and $\lambda^\star \in \mathbb{R}^m$ are the optimal primal and dual solutions, respectively [**?**]. The problem in Eq. 4.2 is equivalently posed as an unconstrained problem by penalizing the cost function via an indicator function $I : \mathbb{R} \to \mathbb{R}$ [**?**, **?**],

$$\min f_0\left(\mathbf{x}\right) + \sum_{i=1}^{m} I\left(f_i\left(\mathbf{x}\right)\right), \qquad \text{where} \quad I\left(z\right) \equiv \begin{cases} 0 & \text{if } z \leq 0 \\ \infty & \text{if } z > 0 \end{cases} . \tag{4.4}$$

Since the resulting objective function is not differentiable, the indicator function is replaced with a differentiable approximation, in this case the logarithmic barrier function defined for negative values of z as

$$B\left(z\right) = -\frac{1}{t} log\left(-z\right), \tag{4.5}$$

where $t > 0$ is a parameter controlling the strength of the barrier. Starting from a feasible point and staying in the interior of the feasible set; i.e., ensuring that the argument of B always stays negative, sets the stage for transforming the original problem into an unconstrained optimization problem

$$\min f_0\left(\mathbf{x}\right) + \sum_{i=1}^{m} B\left(f_i\left(\mathbf{x}\right)\right) \ . \tag{4.6}$$

Note that the approximation approaches the original problem as $t \to \infty$ [**?**, **?**]. Using this approximation, these methods solve Eq. 4.2 by applying Newton's method to a sequence of equality constrained problems, or to a sequence of modified versions of the KKT conditions. We concentrate on two IP algorithms, a barrier method (SCIP) and a primal-dual method (PDIP) [**?**].

4.1.2.1 Symmetric cone interior point method

The goal of the SCIP is to approximately formulate the inequality constrained problem as an equality constrained problem subsequently solved by Newton's method. SCIP exploits the fact that the Coulomb friction cone can be mapped bijectively to a symmetric cone. SCIP makes use of the algebraic structure presented in [**?**], specifically with the space $A = \mathbb{R}^3$ together with the Jordan product

$$\mathbf{x} \circ \mathbf{y} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{x}^T\mathbf{y} \\ x_{\mathbf{n}}\mathbf{y}_{\mathbf{t}} + y_{\mathbf{n}}\mathbf{x}_{\mathbf{t}} \end{bmatrix} \in \mathbb{R} \times \mathbb{R}^2 \tag{4.7}$$

for all

$$\mathbf{x} = \begin{bmatrix} x_{\mathbf{n}} \\ \mathbf{x}_{\mathbf{t}} \end{bmatrix}, \ \mathbf{y} = \begin{bmatrix} y_{\mathbf{n}} \\ \mathbf{y}_{\mathbf{t}} \end{bmatrix} \in \mathbb{R} \times \mathbb{R}^2 \tag{4.8}$$

and relies on the fact that the cone is symmetric, $\Upsilon_i = \Upsilon_i^\circ$. Unless $\mu_i = 1, \ \forall \, i = 1, ..., n$, the cones Υ_i and Υ_i° must first be transformed using

$$\mathbf{x} = T_x \cdot \gamma = \begin{bmatrix} T^x_{\mu_1} & & \\ & \ddots & \\ & & T^x_{\mu_n} \end{bmatrix} \cdot \gamma, \ \mathbf{y} = T_y \cdot \left(\boldsymbol{N}\gamma + \mathbf{r}\right) = \begin{bmatrix} T^y_{\mu_1} & & \\ & \ddots & \\ & & T^y_{\mu_n} \end{bmatrix} \cdot \left(\boldsymbol{N}\gamma + \mathbf{r}\right) \tag{4.9}$$

with

$$T^x_{\mu_i} = \begin{bmatrix} \mu_i & & \\ & 1 & \\ & & 1 \end{bmatrix}, \; T^y_{\mu_i} = \begin{bmatrix} 1 & & \\ & \mu_i & \\ & & \mu_i \end{bmatrix} \tag{4.10}$$

and $\bar{N} = T_y N T_x^{-1}$ and $\bar{\mathbf{r}} = T_y \mathbf{r} \in \mathbb{R}^{3n}$, so that the CCP can be rewritten as

$$\Upsilon \ni \mathbf{x} \perp \mathbf{y} = \bar{N}\mathbf{x} + \bar{\mathbf{r}} \in \Upsilon. \tag{4.11}$$

Barrier function and central path The SCIP method using a slightly modified version of the barrier function in Eq. 4.6 [**?**]. As in [**?**], it is convenient for the definition of the central path to minimize a function with a logarithmic barrier for the set

$$\Upsilon \cup (-\Upsilon) = \left\{ \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} \in \mathbb{R}^{3n} \;:\; \det(\mathbf{x}_i) \geq 0, \; \forall\, i = 1, ..., n \right\}, \tag{4.12}$$

where

$$\det(\mathbf{x}) := \frac{1}{2}(x_{\mathbf{n}}^2 - \|\mathbf{x_t}\|^2) \tag{4.13}$$

Rather than using a logarithmic barrier for Υ. Using $\Upsilon \cup (-\Upsilon)$ instead of Υ results in the following form of the barrier function

$$f(\mathbf{x}, \mathbf{y}) = (2n + \rho) \log \mathbf{x}^T \mathbf{y} - 2n \log n - \sum_{i=1}^{n} \log(2 \det(\mathbf{x}_i)) - \sum_{i=1}^{n} \log(2 \det(\mathbf{y}_i)), \tag{4.14}$$

where $\rho > 0$ is an arbitrary constant. The first two terms are a scaled version of the cost function. The last two terms act as a logarithmic potential that drives $\mathbf{x}_i$ and $\mathbf{y}_i$ away from the manifold defined by $\det(\mathbf{x}_i) = 0$, $\det(\mathbf{y}_i) = 0$, $i = 1, ..., n$; i.e., the boundary of the double cone $\Upsilon \cup (-\Upsilon)$. We split the potential function into two parts via

$$f(\mathbf{x}, \mathbf{y}) = \rho \log \mathbf{x}^T \mathbf{y} + f_{cen}(\mathbf{x}, \mathbf{y}), \tag{4.15}$$

$$f_{cen}(\mathbf{x}, \mathbf{y}) = 2n \log \frac{\mathbf{x}^T \mathbf{y}/n}{\prod_{i=1}^{n} \left[2\sqrt{\det(\mathbf{x}_i) \det(\mathbf{y}_i)} \right]^{1/n}}. \tag{4.16}$$

The logarithmic barrier f_{cen} penalizes values close to the boundary since the denominator in the logarithm in Eq. 4.16 tends towards zero as $(\mathbf{x}, \mathbf{y})$ approaches the boundary of the feasible set. Let $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$ be a sequence in the feasible set that approaches an optimum $(\mathbf{x}^*, \mathbf{y}^*)$ of Eq. 4.11. If $f_{cen}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) = 0$, the sequence approaches the boundary of the feasible set from the interior as fast as it decreases the cost function and therefore the sequence approaches the optimum strictly from within the feasible set, staying clear from the constraints. If $f_{cen}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) = 0$, the constraints are eliminated from the potential function. It is reduced to a scaled logarithm of the cost function. The central path is defined as

$$S_{cen} := \{(\mathbf{x}, \mathbf{y}) \in \Upsilon \cup (-\Upsilon) \ : \ f_{cen}(\mathbf{x}, \mathbf{y}) = 0\} \tag{4.17a}$$

$$= \{(\mathbf{x}, \mathbf{y}) \in \Upsilon \cup (-\Upsilon) \ : \ \mathbf{x} \circ \mathbf{y} = \alpha \mathbf{e}\} \tag{4.17b}$$

In practice it is not possible to rigorously enforce Eq. 4.17a. Instead, one tries to find a sequence in a small neighborhood of the central path, given by Eq. 4.17b. A Newton step applied to the function

$$\mathbf{g}(\mathbf{x}, \mathbf{y}) = \mathbf{x} \circ \mathbf{y} - \alpha \mathbf{e} = \mathbf{0} \tag{4.18}$$

is a step towards the central path at α. Given a feasible $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$, the search direction of the method is given by a solution to

$$\begin{bmatrix} \nabla_{\mathbf{x}} \mathbf{g}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) & \nabla_{\mathbf{y}} \mathbf{g}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \\ \bar{N} & -I \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}^{(k)} \\ \Delta \mathbf{y}^{(k)} \end{bmatrix} = \begin{bmatrix} \alpha \mathbf{e} - \mathbf{x}^{(k)} \circ \mathbf{y}^{(k)} \\ \mathbf{0} \end{bmatrix} \tag{4.19}$$

where

$$\nabla_{\mathbf{x}} \mathbf{g}(\mathbf{x}, \mathbf{y}) = \frac{1}{\sqrt{2}} \begin{bmatrix} x_{\mathbf{n}} & \mathbf{x}_{\mathbf{t}}^T \\ \mathbf{x}_{\mathbf{t}} & I_{2\times 2} x_{\mathbf{n}} \end{bmatrix}, \ \nabla_{\mathbf{y}} \mathbf{g}(\mathbf{x}, \mathbf{y}) = \frac{1}{\sqrt{2}} \begin{bmatrix} y_{\mathbf{n}} & \mathbf{y}_{\mathbf{t}}^T \\ \mathbf{y}_{\mathbf{t}} & I_{2\times 2} y_{\mathbf{n}} \end{bmatrix}. \tag{4.20}$$

Feasibility Barrier methods, like SCIP, require a strictly feasible starting point. When such a point is not known, the method is preceded by a preliminary stage, called *phase I*, in which a strictly feasible point is computed. The strictly feasible point found during phase I is then used as the starting point for the barrier method, which is called the *phase II* stage [**?**].In the context of the

SCIP method , an artificial variable is introduced to transform the original complementarity problem of size $3n$ into a complementarity problem of size $3n+1$ with an obvious starting point. We introduce the additional variable $s \in \mathbb{R}_+$ and a vector $\mathbf{d} \in \mathbb{R}^{3n}$ and consider the complementarity problem

$$\tilde{C} \ni \tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ s \end{bmatrix}, \; \tilde{C} \ni \tilde{\mathbf{y}} = \begin{bmatrix} \bar{\mathbf{y}} \\ s \end{bmatrix} = \tilde{N}\tilde{\mathbf{x}} + \tilde{\mathbf{r}} = \begin{bmatrix} \bar{N} & \mathbf{d} \\ \mathbf{0} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ s \end{bmatrix} + \begin{bmatrix} \mathbf{r} \\ 1 \end{bmatrix}, \; 0 = \tilde{\mathbf{x}}^T \tilde{\mathbf{y}}. \tag{4.21}$$

The central path of the new CCP is the zero set of

$$f_{cen}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) = 2(n+1) \log \frac{\mathbf{x}^T \mathbf{y}/(n+1)}{s^{\frac{1}{n+1}} \prod_{i=1}^{n} \left[2\sqrt{\det(\mathbf{x}_i)\det(\mathbf{y}_i)} \right]^{\frac{1}{n+1}}}. \tag{4.22}$$

Assume an initial feasible guess $\mathbf{x}^{(0)}$ for the original problem is given. For example, $\mathbf{x}_i^{(0)} = (1,0,0)^T$ obviously lies in int Υ for all $i = 1, ..., n$. We now choose $s^{(0)}$ and $\mathbf{d}$ such that

$$\bar{\mathbf{y}} = \tilde{\mathbf{y}}^{(0)}_{\{1,\ldots,3n\}} = \bar{N}\mathbf{x}^{(0)} + \mathbf{r} + s^{(0)}\mathbf{d} \in \text{int } \Upsilon, \tag{4.23}$$

Additionally, the parameters can be chosen in such a way that $(\tilde{\mathbf{x}}^{(0)}, \tilde{\mathbf{y}}^{(0)})$ lies on a point $\alpha < 0$ on the central path

$$\tilde{\mathbf{x}}^{(0)} \circ \tilde{\mathbf{y}}^{(0)} = \alpha \mathbf{e} \tag{4.24}$$

of the CCP. For all $i = 1, ..., n$ it must hold

$$\begin{bmatrix} \sqrt{2}\alpha \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} x_{in}^{(0)} \bar{y}_{in}^{(0)} + \mathbf{x}_{it}^{(0)T} \bar{\mathbf{y}}_{it}^{(0)} \\ x_{in}^{(0)} \bar{\mathbf{y}}_{it}^{(0)} + \bar{y}_{in}^{(0)} \mathbf{x}_{it}^{(0)} \end{bmatrix} \Rightarrow \bar{y}_{in}^{(0)} = \frac{2\alpha}{x_{in}^{(0)} - \frac{\left\| \mathbf{x}_{it}^{(0)} \right\|^2}{x_{in}^{(0)}}}, \; \bar{\mathbf{y}}_{it}^{(0)} = -\left(\frac{\bar{y}_{in}^{(0)}}{x_{in}^{(0)}} \right) \mathbf{x}_{it}^{(0)}. \tag{4.25}$$

For the last complementarity condition we have to make sure that

$$s^{(0)} \cdot 1 = s^{(0)} = \mathbf{x}_i^{(0)T} \bar{\mathbf{y}}_i^{(0)} = 2\alpha. \tag{4.26}$$

This means that

$$\mathbf{d} = \frac{1}{2\alpha} (\bar{\mathbf{y}}^{(0)} - \bar{N}\mathbf{x}^{(0)} - \bar{\mathbf{r}}). \tag{4.27}$$

A search direction for Equation 4.21 is given by

$$\begin{bmatrix} \nabla_{\mathbf{x}}\mathbf{g}(\mathbf{x}^{(k)},\bar{\mathbf{y}}^{(k)}) & \nabla_{\bar{\mathbf{y}}}\mathbf{g}(\mathbf{x}^{(k)},\bar{\mathbf{y}}^{(k)}) & \mathbf{0} \\ \bar{N} & -I & \mathbf{d} \\ \mathbf{0} & \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}^{(k)} \\ \Delta\bar{\mathbf{y}}^{(k)} \\ \Delta s^{(k)} \end{bmatrix} = \begin{bmatrix} \alpha\mathbf{e} - \nabla_{\bar{\mathbf{y}}}\mathbf{g}(\mathbf{x}^{(k)},\bar{\mathbf{y}}^{(k)})\bar{\mathbf{y}}^{(k)} \\ \mathbf{0} \\ 2\alpha - s^{(k)} \end{bmatrix} \tag{4.28}$$

or equivalently by

$$\Delta s^{(k)} = 2\alpha - s^{(k)}, \tag{4.29a}$$

$$\tilde{\mathbf{A}}\Delta\mathbf{x}^{(k)} = \mathbf{b}, \tag{4.29b}$$

$$\Delta\bar{\mathbf{y}}^{(k)} = \bar{N}\Delta\mathbf{x}^{(k)} + \Delta s^{(k)}\mathbf{d} \tag{4.29c}$$

where

$$\tilde{\mathbf{A}} = \left[\nabla_{\bar{\mathbf{y}}}\mathbf{g}(\mathbf{x}^{(k)},\bar{\mathbf{y}}^{(k)})^{-1}\nabla_{\mathbf{x}}\mathbf{g}(\mathbf{x}^{(k)},\bar{\mathbf{y}}^{(k)}) + \bar{N}\right] \ , \tag{4.30a}$$

$$\mathbf{b} = \alpha\left(\mathbf{x}^{(k)}\right)^{-1} - \bar{\mathbf{y}}^{(k)} - \Delta s^{(k)}\mathbf{d} \ , \tag{4.30b}$$

$$\mathbf{x}^{-1} = \frac{1}{\det(\mathbf{x})}\mathbf{J}\mathbf{x} \ , \quad \text{where} \quad \mathbf{J} \equiv \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_{2\times 2} \end{bmatrix} \in \mathbb{R}^{3\times 3}. \tag{4.30c}$$

Since an optimal solution $(\tilde{\mathbf{x}}^*, \tilde{\mathbf{y}}^*)$ implies that $s^* = 0$, it follows that $(\mathbf{x}^*, \bar{\mathbf{y}}^*) = (\mathbf{x}^*, \mathbf{y}^*)$ is optimal for the original CCP. During the iteration $s^{(k)}$ decreases and eventually $\mathbf{y}^{(k)} = \bar{\mathbf{y}}^{(k)} - s^{(k)}\mathbf{d}$ is feasible. Then we can use the current iterate $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$ as a starting point for a feasible interior point for the original CCP.

Maximum step length After determining the direction of the step $(\Delta\mathbf{x}, \Delta\bar{\mathbf{y}})$, the next task is to identify the supremum of step sizes $\theta > 0$ such that $\mathbf{x} + \theta\Delta\mathbf{x} \in \operatorname{int}\Upsilon$ and $\bar{\mathbf{y}} + \theta\Delta\bar{\mathbf{y}} \in \operatorname{int}\Upsilon$. As in [**?**], the maximum step size θ_{max} for $\Delta\mathbf{x}$ and $\Delta\bar{\mathbf{y}}$ is determined by

$$\theta_{\text{max}} = \begin{cases} 1, \text{ if } \Delta\mathbf{x} \in \Upsilon \\ \frac{\det\mathbf{x}}{\sqrt{\left[\frac{(\Delta\mathbf{x})^T\mathbf{J}\mathbf{x}}{2}\right]^2 - \det(\Delta\mathbf{x})\det(\mathbf{x})} - \frac{(\Delta\mathbf{x})^T\mathbf{J}\mathbf{x}}{2}}, \text{ else.} \end{cases} \tag{4.31}$$

Nesterov-Todd scaling Finally, the convergence of the method can be improved substantially by rescaling the space in which the cone Υ lives at the beginning of each iteration using the Nesterov-Todd scaling scheme [**?**]. This is done by replacing Eq. 4.29b with

$$\left[P(\mathbf{w}) + \bar{N}\right] \Delta\mathbf{x}^{(k)} = \mathbf{b}, \tag{4.32}$$

where

$$P(\mathbf{w}) = \mathbf{w}\mathbf{w}^T - \det(\mathbf{w})\mathbf{J}, \tag{4.33}$$

and the scaling point $\mathbf{w} \in \mathbb{R}^3$ is

$$\mathbf{w} = \frac{\mathbf{y} + \lambda\mathbf{J}\mathbf{x}}{\sqrt{\mathbf{x}^T\mathbf{y} + 2\sqrt{\det(\mathbf{x})\det(\mathbf{y})}}} \tag{4.34}$$

where

$$\lambda = \det\left(\mathbf{w}\right) = \sqrt{\frac{\det\left(\mathbf{y}\right)}{\det\left(\mathbf{x}\right)}} \tag{4.35}$$

4.1.2.2 Primal-dual interior point method

Primal-dual IP (PDIP) methods are very similar to the barrier method, with some differences [**?**]:

- The search directions in a PDIP method are obtained from Newton's method, applied to modified KKT equations (the optimality conditions for the logarithmic barrier centering problem). The primal-dual search directions are similar to, but not quite the same as, the search directions that arise in the barrier method.
- In a PDIP method, the primal and dual iterates are not necessarily feasible.

The starting point of the PDIP method [**?**] is the unconstrained problem

$$\begin{aligned} f_i\left(\mathbf{x}\right) &< 0, \quad i = 1, \ldots, m && (4.36\text{a}) \\ \nabla f_0\left(\mathbf{x}\right) + \sum_{i=1}^{m} -\frac{1}{t f_i\left(\mathbf{x}\right)} \nabla f_i\left(\mathbf{x}\right) &= 0\ . && (4.36\text{b}) \end{aligned}$$

Defining $\lambda_i = \frac{-1}{tf_i(\mathbf{x})}$, the KKT conditions assume the form

$$\begin{aligned} f_i(\mathbf{x}) &< 0, \quad i = 1, \ldots, m \qquad &(4.37\text{a}) \\ \lambda_i &> 0, \quad i = 1, \ldots, m \qquad &(4.37\text{b}) \\ \nabla f_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i \nabla f_i(\mathbf{x}) &= 0 \qquad &(4.37\text{c}) \\ -\lambda_i f_i(\mathbf{x}) &= \frac{1}{t}, \quad i = 1, \ldots, m\ . \qquad &(4.37\text{d}) \end{aligned}$$

Using the notation

$$\mathbf{r}_t(\mathbf{x}, \lambda) = \begin{bmatrix} \nabla f_0(\mathbf{x}) + \nabla \mathbf{f}(\mathbf{x})^T \lambda \\ -diag(\lambda)\, \mathbf{f}(\mathbf{x}) - \frac{1}{t}\mathbf{1} \end{bmatrix},\ \mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{bmatrix},\ \nabla \mathbf{f}(\mathbf{x}) = \begin{bmatrix} \nabla f_1(\mathbf{x})^T \\ \vdots \\ \nabla f_m(\mathbf{x})^T \end{bmatrix},\ \mathbf{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \tag{4.38}$$

the goal is to solve the nonlinear system $\mathbf{r}_t(\mathbf{x}, \lambda) = \mathbf{0}$. This is accomplished via a Newton method

$$\begin{bmatrix} \mathbf{x} + \Delta \mathbf{x} \\ \lambda + \Delta \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{x} \\ \lambda \end{bmatrix} - \nabla \mathbf{r}_t(\mathbf{x}, \lambda)^{-1} \mathbf{r}_t(\mathbf{x}, \lambda)\,. \tag{4.39}$$

The PDIP search directions are closely related to the search directions used in the SCIP method, but not quite the same. In practice, the search direction is obtained by solving the linear system

$$\begin{bmatrix} \nabla^2 f_0(\mathbf{x}) + \sum\limits_{i=1}^{m} \lambda_i \nabla^2 \mathbf{f}_i(\mathbf{x}) & \nabla \mathbf{f}(\mathbf{x})^T \\ -diag(\lambda)\, \nabla \mathbf{f}(\mathbf{x}) & -diag(\mathbf{f}(\mathbf{x})) \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \lambda \end{bmatrix} = -\mathbf{r}_t(\mathbf{x}, \lambda)\ . \tag{4.40}$$

The method can be described in pseudocode as follows [**?**]:

```
ALGORITHM PDIP($f_0$, $f_1$, ..., $f_m$, $\mu \geq 1$, $\epsilon$)
    while $||\mathbf{r}_t(\mathbf{x}, \lambda)||_2 > \epsilon$
        Set $t = \frac{\chi m}{-f^T \lambda}$
        Compute search direction $[\Delta \mathbf{x}^T \quad \Delta \lambda^T]^T$
        Compute step length $s > 0$ via line search
        Update: $\mathbf{x} = \mathbf{x} + s\Delta \mathbf{x}$, $\lambda = \lambda + s\Delta\lambda$
    endwhile
    return Solution $\mathbf{x}^\star = \mathbf{x}$, $\lambda^\star = \lambda$ .
```

The value of t is gradually increased as indicated in step (2). A line search is performed via backtracking to ensure that $f_i(\mathbf{x}) < 0$ and $\lambda_i > 0$ for all i. This is achieved by first computing the largest step length, s_{max}, which maintains $\lambda_i \geq 0$:

$$s_{max} = \min\{1, \min\{-\lambda_i/\Delta\lambda_i | \Delta\lambda_i < 0\}\} \tag{4.41}$$

In practice, to enforce $\lambda_i > 0$,we set $s = 0.99s_{max}$. Subsequently, s is repeatedly multiplied by a parameter β until $f_i(\mathbf{x}) < 0, \forall i$. Finally, s is scaled by α until the following condition is satisfied:

$$||\mathbf{r}_t(\mathbf{x} + \Delta\mathbf{x}, \lambda + \Delta\lambda)||_2 \leq (1 - \alpha s)||\mathbf{r}_t(\mathbf{x}, \lambda)||_2 \tag{4.42}$$

According to [**?**], it is common to take $\chi = 10$, $\alpha \in [0.01, 0.1]$, and $\beta \in [0.3, 0.8]$.

What is left to fully define the PDIP method is to provide the quantities required to carry out one Newton step according to Eq. 4.40 given the problem stated in Eq. 3.12. First, the objective function is

$$f_0(\gamma) = \frac{1}{2}\gamma^T\mathbf{N}\gamma + \mathbf{r}^T\gamma. \tag{4.43}$$

Define $\mathbf{f}(\gamma)$ as

$$f_i(\gamma) = \begin{cases} \frac{1}{2}\left(\gamma_{i,u}^2 + \gamma_{i,v}^2 - \mu_i^2\gamma_{i,n}^2\right) & : i \in [1, \ldots, N_c] \\ -\gamma_{(i-N_c),n} & : i \in [N_c + 1, \ldots, 2N_c]\ . \end{cases} \tag{4.44}$$

Therefore,

$$\nabla f_i(\gamma) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -\mu_i^2\gamma_{i,n} \\ \gamma_{i,u} \\ \gamma_{i,v} \\ 0 \\ \vdots \\ 0 \end{bmatrix},\ i = 1, \ldots, N_c, \text{ and } \nabla f_i(\gamma) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix},\ i = N_c + 1, \ldots, 2N_c\ , \tag{4.45}$$

and

$$
\begin{aligned}
\nabla^2 f_i\left(\gamma\right) &= \begin{bmatrix}
0 & \cdots & & & & & & \cdots & 0 \\
\vdots & \ddots & \vdots & \vdots & & & & & \vdots \\
 & \cdots & 0 & 0 & \cdots & & & & \\
 & \cdots & 0 & -\mu_i^2 & 0 & 0 & & & \\
 & & \vdots & 0 & 1 & 0 & \vdots & & \\
 & & & 0 & 0 & 1 & 0 & \cdots & \\
 & & & & \cdots & 0 & 0 & \cdots & \\
\vdots & & & & & \vdots & \vdots & \ddots & \vdots \\
0 & \cdots & & & & & & \cdots & 0
\end{bmatrix}, \ i = 1, \ldots, N_c \ , \qquad (4.46a) \\
\nabla^2 f_i\left(\gamma\right) &= \mathbf{0}_{(3N_c \times 3N_c)}, \ i = N_c + 1, \ldots, 2N_c \ . \qquad (4.46b)
\end{aligned}
$$

Let $\mathbf{A}$ be the Newton step matrix of Eq. 4.40. Using the notation $\mathbf{B} = \nabla \mathbf{f}\left(\gamma\right)^T \in \mathbb{R}^{3N_c \times 2N_c}$, $\mathbf{C} = -diag\left(\lambda\right) \nabla \mathbf{f}\left(\gamma\right) \in \mathbb{R}^{2N_c \times 3N_c}$, and $\mathbf{D} = -diag\left(\mathbf{f}\left(\gamma\right)\right) \in \mathbb{R}^{2N_c \times 2N_c}$, $\mathbf{r}_d = \nabla f_0\left(\mathbf{x}\right) + \nabla \mathbf{f}\left(\mathbf{x}\right)^T \lambda \in \mathbb{R}^{3N_c}$, and $\mathbf{r}_g = -diag\left(\lambda\right) \mathbf{f}\left(\mathbf{x}\right) - \frac{1}{t}\mathbf{1} \in \mathbb{R}^{2N_c}$,

$$
\mathbf{A} = \begin{bmatrix} \mathbf{N} + \hat{\mathbf{M}} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \qquad \text{and} \qquad \mathbf{r}_t = \begin{bmatrix} \mathbf{r}_d \\ \mathbf{r}_g \end{bmatrix} \ , \qquad (4.47)
$$

where

$$
\begin{aligned}
\hat{\mathbf{M}} &= \sum_{i=1}^{2N_c} \lambda_i \nabla^2 \mathbf{f}\left(\gamma\right) = diag\left(\hat{\mathbf{m}}\right) \in \mathbb{R}^{3N_c \times 3N_c} \qquad (4.48a) \\
\hat{\mathbf{m}} &= \left[-\mu_1^2 \lambda_1, \lambda_1, \lambda_1, -\mu_2^2 \lambda_2, \lambda_2, \lambda_2, ..., -\mu_{N_c}^2 \lambda_{N_c}, \lambda_{N_c}, \lambda_{N_c}\right]^T \in \mathbb{R}^{3N_c} \ . \qquad (4.48b)
\end{aligned}
$$

4.2 Comparison of first and second-order methods

We compare six methods to solve the problem in Eq. 3.12. Three of them; i.e., PJ, PGS, and APGD, are first-order solvers that rely on sparse matrix–vector multiplications. The next two, SCIP and PDIP, are second-order solvers that require the solution of sparse linear systems with coefficient matrices defined in Eqs. 4.47 and 4.30a, respectively. SCIP was implemented as discussed in [**?**], where the linear systems are solved using a Krylov-subspace iterative method.

The sixth method, P-SCIP, is identical to SCIP except that it uses preconditioning to accelerate the convergence at each Newton step. PDIP and P-SCIP use the same preconditioner and Krylov-subspace iterative method. The implementation of these six methods was done on a GPU card using CUDA [**?**]. The sparse matrix–vector operations relied on the CUSP library [**?**]. The sparse linear solver used was SaP::GPU [**?**, **?**, **?**]. All six methods were implemented in the same code and drew on the same collision detection implementation [**?**].

4.2.1 Termination criterion

Comparing the accuracy and/or efficiency of the six algorithms requires an "accuracy metric" to ensure that each algorithm yields a solution approximation that is sufficiently accurate. In general, we prescribe a solution tolerance and the algorithm stops when the accuracy of the approximate solution, as measured using the accuracy metric of choice, falls below the prescribed tolerance; i.e., $r < \tau$. The tolerance τ is user prescribed, and the accuracy metric is tied to

$$r = \max\left\{r_p, r_d, r_c\right\} , \tag{4.49}$$

where $r_p = ||\mathbf{f}_p||_\infty$ and $r_d = ||\mathbf{f}_d||_\infty$ are measures of the feasibility of the primal and dual vectors:

$$\begin{aligned}
\mathbf{f}_p &= \left[\ldots, -\min\left(0, \mu_i\gamma_{i,n} - \sqrt{\gamma_{i,u}^2 + \gamma_{i,v}^2}\right), \ldots\right]^T \in \mathbb{R}^{N_c} && \text{(4.50a)}\\
\mathbf{f}_d &= \left[\ldots, -\min\left(0, \frac{1}{\mu_i}\mathbf{y}_{i,n} - \sqrt{\mathbf{y}_{i,u}^2 + \mathbf{y}_{i,v}^2}\right), \ldots\right]^T \in \mathbb{R}^{N_c} , && \text{(4.50b)}
\end{aligned}$$

and $\mathbf{y} = \mathbf{N}\gamma + \mathbf{r}$. The residual r_c is a measure of the worst violation of the complementarity conditions stated in Eq. 3.10b.

In the PDIP method the iterates are not necessarily feasible, except in the limit as the algorithm converges. This means that we cannot easily evaluate a duality gap associated with step k of the algorithm, as we do in the other methods. Instead, for any γ that satisfies $f(\gamma) < 0$ and $\lambda \geq 0$, we define a surrogate duality gap [**?**]

$$\hat{\eta}(\gamma, \lambda) = -f(\gamma)^T\lambda \tag{4.51}$$

The surrogate gap $\hat{\eta}$ would be the duality gap, if γ were primal feasible and λ were dual feasible.

The suitability of the chosen accuracy metric is verified in conjunction with a drafting test in which a rectangular blade moves through a trench filled with granular material, see Fig. 4.1. In this 3D setup there are $21,638$ spherical bodies with a density $\rho = 2,500$ g/cm^3, a friction coefficient $\mu = 0.25$, and a randomly distributed radius between 0.008 and 0.016 m. The blade has a width of 0.2 m and moves with a constant horizontal speed of 0.2 m/s at an initial depth of 0.2 m. The simulation length was 3 s and a reference solution was generated with APGD with a integration step size of $h = 0.001$ s and a tolerance $\tau = 1 \times 10^4$ N. At $t = 3$ s, the force that the blade experienced was considered to have reached a steady value and the entire system state was exported to a file. That system state was subsequently imported into PJ, PGS, APGD, SCIP, and PDIP and each solver was used to take one single step using various values τ. The purpose of this exercise was to understand whether the solvers, starting from an identical configuration, find solutions that are increasingly closer to each other as the solver tolerance is tightened. The most challenging results to converge are forces and this is what the plot in Fig. 4.2 shows. Indeed, as τ becomes smaller, all solutions approach the same total force acting on the blade. This indicates that the draft force for tight tolerances is independent of the solver, which confirms that the heuristics adopted in defining the accuracy metric via Eq. 4.49 yield a good way to judge the accuracy of an approximate solution.

(a)

(b)

Figure 4.1: Simulation snapshots of a blade moving through granular material at the initial (left) and final (right) configurations.

The plot in Fig. 4.3 indicates the speed of convergence; i.e., amount of iterations to meet a user prescribed accuracy specified by the user via τ. Two remarks are in order: (i) the interior point solvers require much fewer iterations than the first-order solvers to reach values of τ that are of practical interest; (ii) the cost per iteration is quite different from method to method and simply requiring a smaller number of iterations does not translate into better overall efficiency, which is measured as the amount of compute time spent to meet a certain solution tolerance.

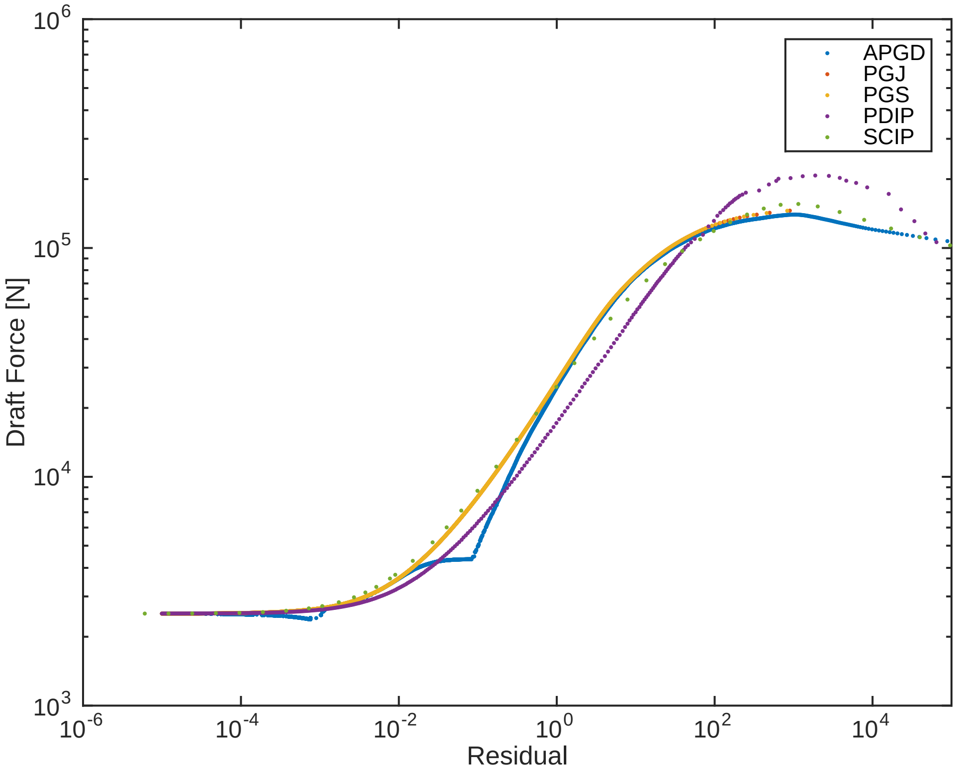


Figure 4.2: The draft force that the blade experiences as a function of the residual for a single time step during steady-state operation.

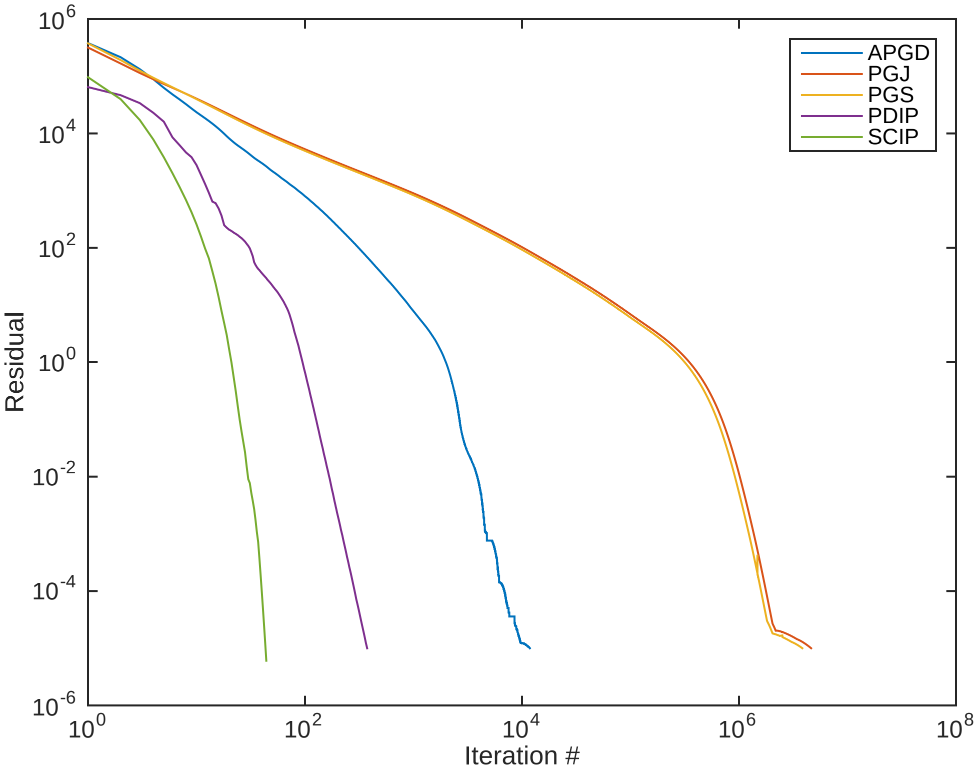


Figure 4.3: The draft force that the blade experiences as a function of the iteration number for a single time step during steady-state operation.

4.2.2 Solver accuracy investigation

The test is tied to a simulation of a container-filling process, see Fig. 4.4. The gravitational acceleration $g = -9.81$ m/s^2 acts in the vertical direction. The simulation was run for 11 s with a time step $h = 0.01$ s and at varying tolerances τ. The simulation attributes monitored for this test were computation time. Since the only external force acting on the bodies is due to gravity and there is no jamming, the resulting composite contact force should be equal to the weight of the $1,000$ spheres. The purpose of the test is to understand whether, as the tolerance is tightened, the solutions generated by the six methods converge towards a unique solution. This test has a “global convergence” demonstration purpose and in this respect is different from the one in §4.2.1, which assessed an accuracy metric by comparing solver behavior under the assumption that all solvers, staring from the same configuration, advanced the state of the system by one time step only.

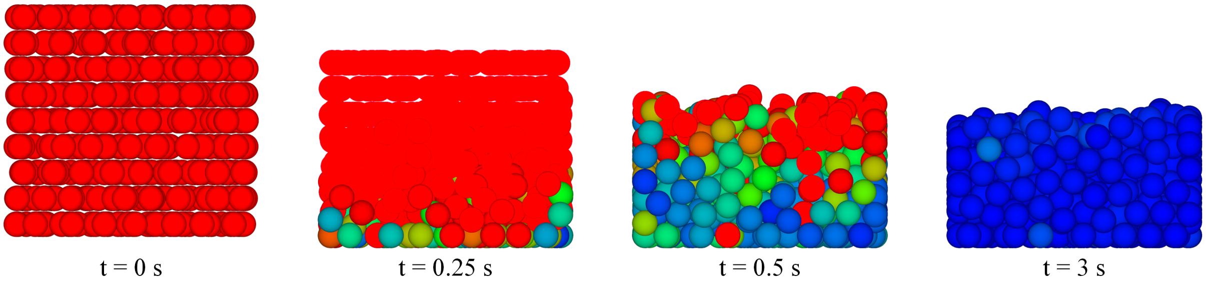


Figure 4.4: Simulation of $1,000$ spheres (radius $r = 1$ m, mass $m = 1$ kg) falling under gravity into a container. Color code: red–high speed, green–moderate speed, blue–zero speed. Friction coefficient $\mu = 0.25$ (sphere–sphere and sphere–container).

The results in Fig. 4.5 show the percent error in the composite contact force that the container experiences during the last second of an 11 s simulation. Three conclusions can be drawn from this study: (i) as the solver tolerance is tightened, the percent error goes towards zero; (ii) the percent error as a function of the tolerance is relatively independent of the solver considered; and, (iii) the solver tolerance is a good proxy for the average physical error, which comes in line with results obtained in the §4.2.1. As expected, there is virtually no difference between SCIP and P-SCIP.

Figure 4.6 summarizes the number of iterations for convergence required by each solver. The values reported are averages computed over the last second of simulation. Specifically, if there were 100 time steps taken during the last second and at each time step the number of iterations was $N_i^{(it)}$, $i = 1, \ldots 100$, the value used in Fig. 4.6 was $N_{average}^{(it)} = (\sum_{i=1}^{100} N_i^{(it)})/100$. In general, as the tolerance is tightened, the number of iterations required by all solvers increases. The PJ solver requires the largest amount of iterations, although the PGS solver comes in as a close second. At the other end of the spectrum, the SCIP and P-SCIP solvers require the least amount of iterations. The number of iterations is of secondary relevance insofar a first-order method vs. second-order method comparison is concerned. Indeed, the cost of one iteration is quite different between the two. The relevant informations pertain solver efficiency; i.e., how much time is spent to complete a simulation given a user prescribed accuracy via τ. This aspect is addressed in the next section.

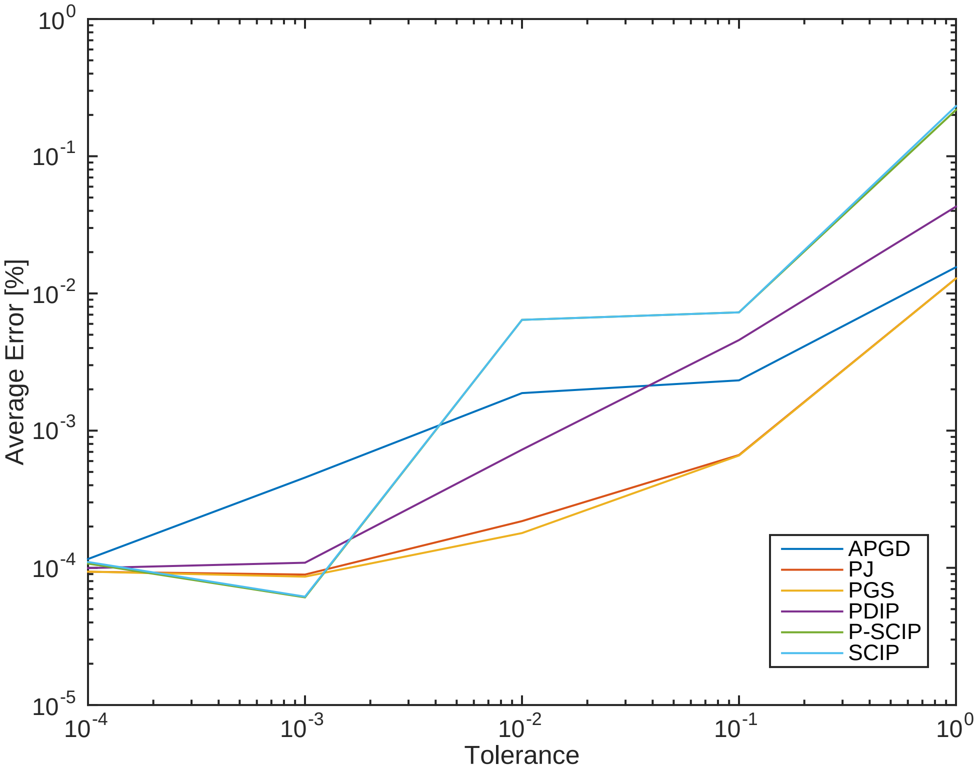


Figure 4.5: Percent error in composite contact force as a function of solver tolerance. Results report the average contact force that the container experienced for the last second of simulation in the filling test.

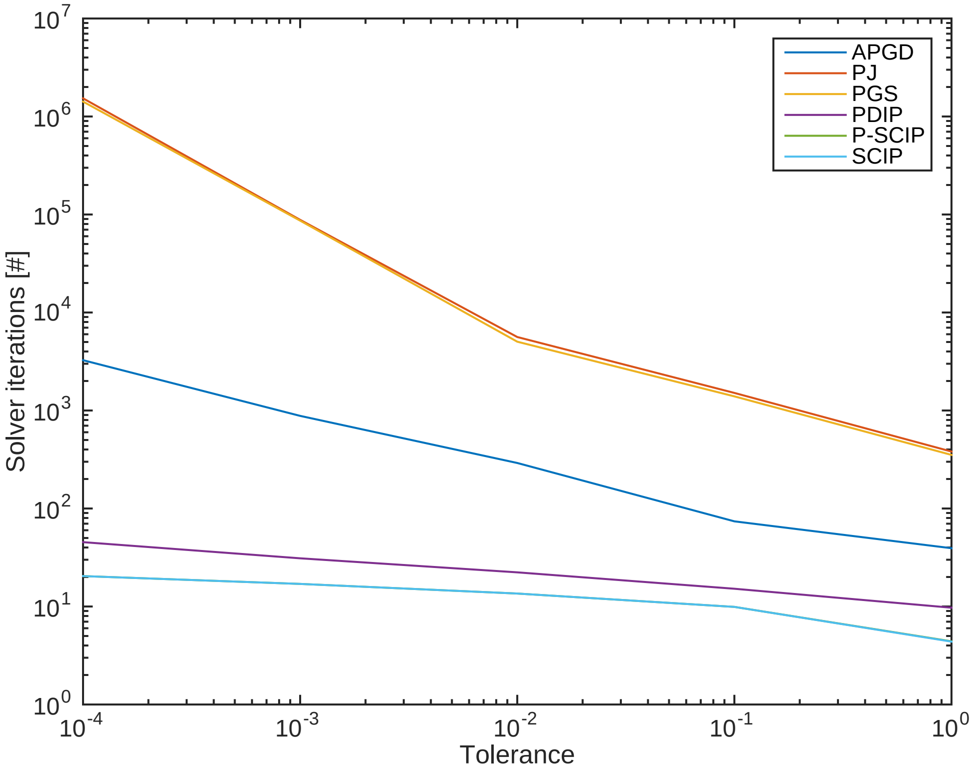


Figure 4.6: The average number of solver iterations required to reach a specified tolerance for the final second of an eleven second filling operation.

4.2.3 Efficiency and scaling analysis

Using the same setup as in the previous section, the interest is in performing a scaling analysis. To this end, the number of spheres dropped into the container was varied from 160 to $16,000$.The simulation was run for 11 s with a time step $h = 0.01$ s and a solver tolerance $\tau = 0.001$. The quantities of interest were the computation time taken by the last second of simulation and the average number of iterations performed.

Figure 4.7 shows the number of iterations required by each solver for $\tau = 0.001$ as a function of the number of bodies dropped in the container. Much like in §4.2.2, the second-order solvers require significantly fewer iterations. For the case of $16,000$ bodies, the SCIP solvers require approximately 50 times fewer iterations than the APGD solver and $60,000$ times fewer iterations than the PJ solver.

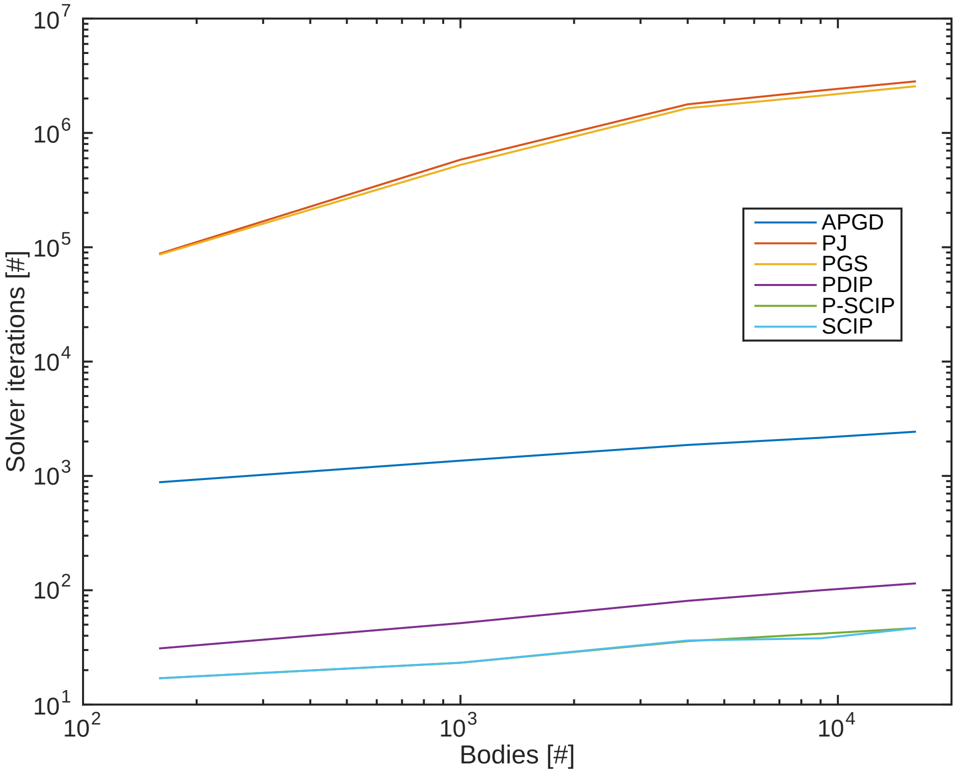


Figure 4.7: The average number of solver iterations to solve a time step during the final second of the filling simulation.

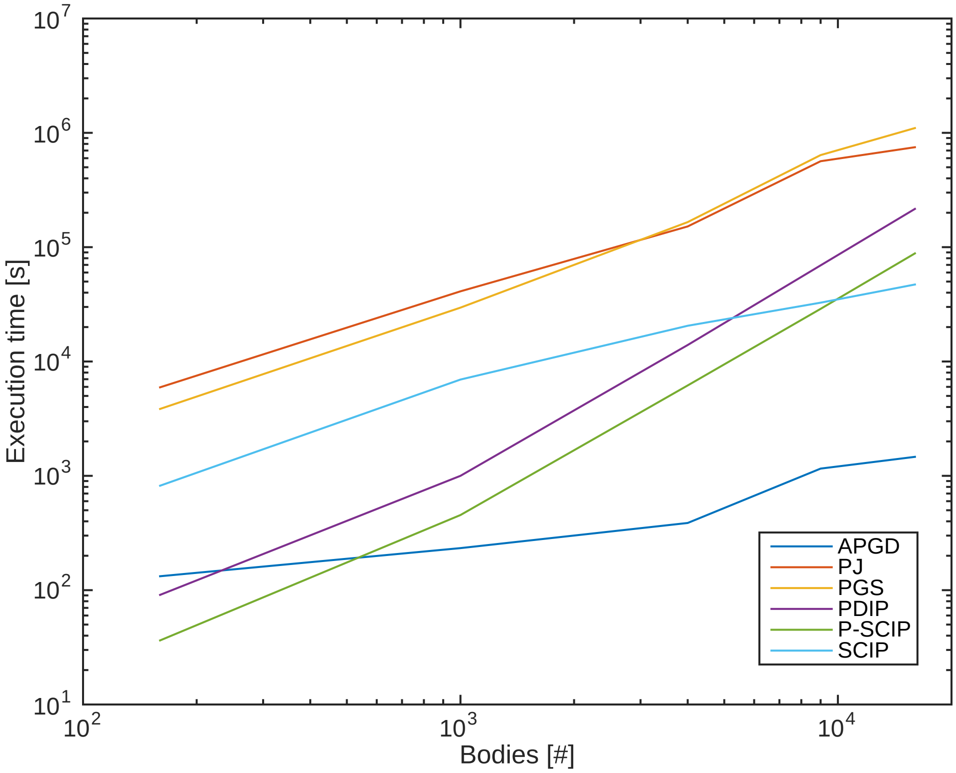


Figure 4.8: The total execution time to solve the final second of the filling simulation. The simulations were run on an Intel Nehalem Xeon E5520 2.26GHz processor with an NVIDIA K40c GPU.

The computational time to solve the final second of the container filling as a function of the number of bodies is shown in Fig. 4.8. For systems with less than 500 bodies, the second-order solvers are more efficient; i.e., they reach a desired level of accuracy in a shorter amount time. For large problem sizes, a first-order method; i.e., APGD, becomes more efficient. This test elucidates one issue; i.e., whether first-order or second-order methods are superior. When solving the problem in Eq. 3.12, PJ, PGS, and APGD rely only on the first derivative; i.e., the gradient, of the objective function, and call repeatedly for sparse matrix–vector multiplications. SCIP, P-SCIP and PDIP are better informed since they fall back on a Newton step, which provides second-order convergence speed. However, these latter methods require the solution of linear systems. In general, sparse matrix–vector multiplications scale better than the factorization of sparse matrices. Against this backdrop; i.e., more information in the solution but worse scaling, the second-order methods are faster for small problems. However, the improved convergence order does not compensate for the linear algebra costs incurred.

4.2.4 Compaction test

In this test, a slab is placed on top of $4,000$ spherical bodies pressing them down inside a container. The mass of the slab is used to dial up the amount of pressure that the spheres should collectively support. This test is motivated by a scenario such as a heavy vehicle operating on granular terrain. For a wheeled vehicle, one body; i.e., the wheel, imparts over a small area; i.e., the contact patch, a significant amount of pressure onto a large collection of small bodies. A similar benchmark test was used in [**?**, **?**] and shown to be strenuous owing to challenges posed by converging to a frictional contact force distribution that collectively combines to sustain the slab weight. Each sphere had a radius $r = 1$ m, mass $m = 1$ kg, and friction coefficient $\mu = 0.25$. The slab prevents spheres from escaping the container and has a mass $m = 4,000$ kg and friction coefficient $\mu = 0.25$. The simulation was run for 4 s with a time step $h = 0.001$s and a solver tolerance of $\tau = 1 \times 10^{-4}$. The time evolution of the system is shown in Fig. 4.9.

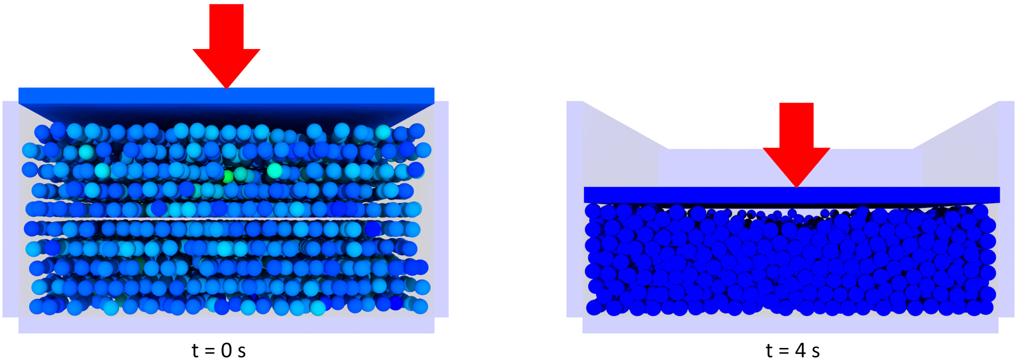


Figure 4.9: A simulation of $4,000$ spherical bodies in a container being compacted by a slab. The colors of the bodies represent the magnitude of the linear velocity (red = fast, blue = slow).

The statistics monitored for this test were the computation time, number of iterations for convergence, and frictional contact force impressed onto the container. Fig. 4.10 shows the total compaction force as a function of the solver residual for a single time step during the settled configuration; i.e., at the tail end of the analysis. Based on this plot, the compaction force experienced by the container for a given residual is similar for all of the solvers, demonstrating the independence of the residual criteria on the solver and reinforcing the conclusions in §4.2.1. As expected, the number of iterations as a function of residual, shown in Fig. 4.11, is solver dependent.

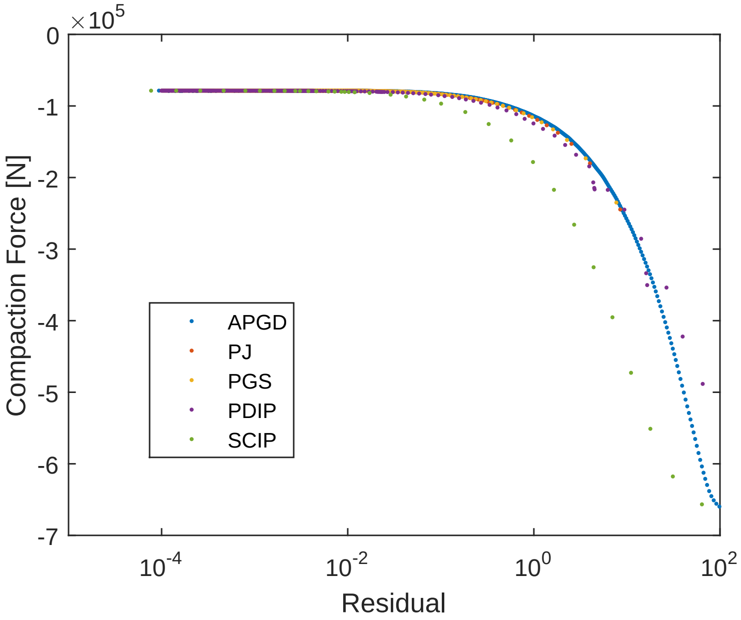


Figure 4.10: The compaction force that the container experiences as a function of the residual for a single time step during the settled configuration.

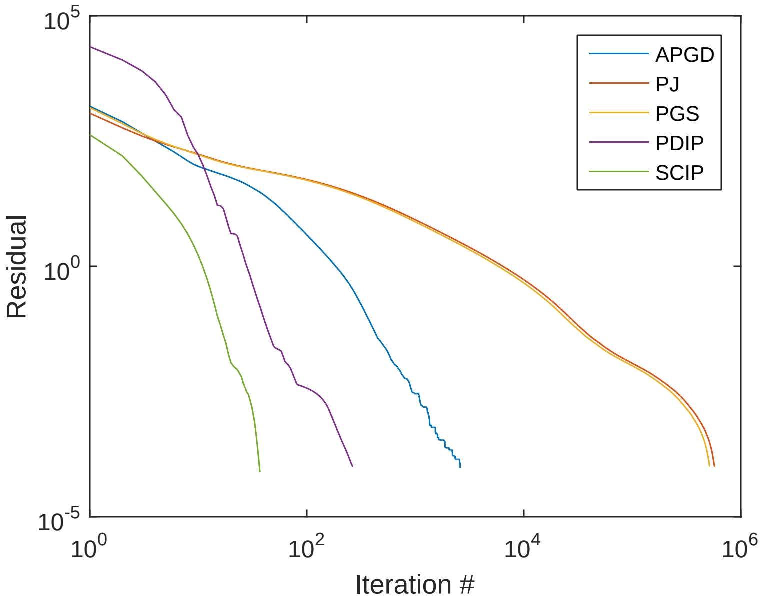


Figure 4.11: The compaction force that the container experiences as a function of the iteration number for a single time step during the settled configuration.

To further stress test the solvers, the slab mass was varied from $m = 10$ to $100,000$ kg. With the exception of the SCIP solver, the results shown in Fig. 4.12 indicate that increasing the mass of the slab requires additional solver iterations to reach a tolerance of $\tau = 1 \times 10^{-4}$. Note that the number of iterations was set to a maximum of $1,000,000$ to prevent excessive execution times.

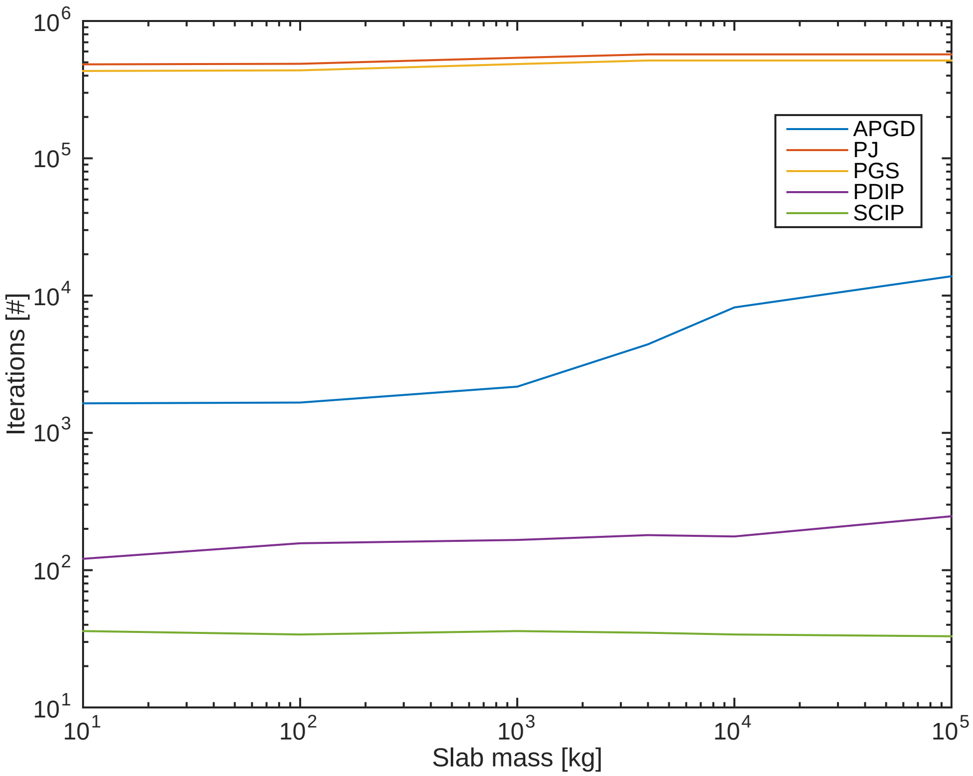


Figure 4.12: The total number of solver iterations required to solve a single time step of the compaction test as a function of the slab mass.

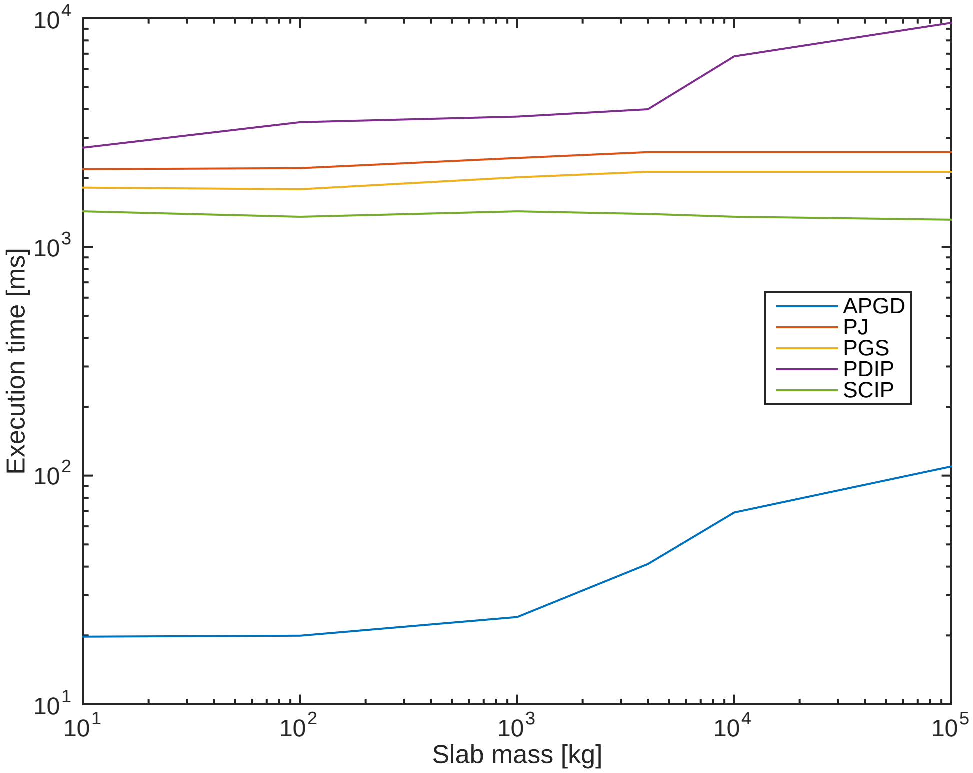


Figure 4.13: The total execution time to solve a single time step of the compaction test as a function of the slab mass. The simulations were run on an Intel Nehalem Xeon E5520 2.26GHz processor with an NVIDIA K40c GPU.

Chapter 5

Iterative refinement of the frictional contact solution

It should be noted that the relaxation introduced by Eq. 3.9 is non-physical. The purpose of this chapter is to describe an iterative refinement scheme, called anti-relaxation, to update the objective function of the CCQO to yield a optimum that is equivalent to the solution of the original NRMD problem. It is shown that the proposed anti-relaxation step incurs a relatively modest cost to improve the quality of a numerical solution strategy which poses the calculation of the frictional contact forces as a cone-complementarity problem. To accomplish this, the objective function of the CCQO can be modified by an anti-relaxation term, $\mathbf{a}^T = \left[\mathbf{a}_1^T, \ldots, \mathbf{a}_{N_c}^T\right]$ with $\mathbf{a}_i = \left[\alpha_i, 0, 0\right]^T \in \mathbb{R}^3$, to result in a solution that is equivalent to the optimum of Eq. 3.8. This modification results in the anti-relaxed CCQO,

$$\begin{aligned} \min \mathbf{q}\left(\gamma\right) &= \frac{1}{2}\gamma^T \boldsymbol{N} \gamma + \left(\mathbf{r} + \mathbf{a}\right)^T \gamma \\ &\text{subject to } \gamma_i \in \Upsilon_i \text{ for } i = 1, 2, \ldots, N_c, \end{aligned} \tag{5.1}$$

The value of $\mathbf{a}$ can be determined by considering the Lagrangian, $\mathcal{L}\left(\gamma, \lambda\right)$, for the constrained optimization problem in Eq. 5.1,

$$\mathcal{L}\left(\gamma, \lambda\right) = \frac{1}{2}\gamma^T \boldsymbol{N} \gamma + \left(\mathbf{r} + \mathbf{a}\right)^T \gamma - \sum_{i=1}^{N_c} \lambda_i \left(\mu_i \gamma_{i,n} - \sqrt{\gamma_{i,u}^2 + \gamma_{i,w}^2}\right). \tag{5.2}$$

Then, the KKT conditions for Eq. 5.1 can be stated as follows [**?**]. If γ^* is a local solution of Eq. 5.1, then there exists a vector λ^* of Lagrange multipliers, $\lambda_i^*, i = 1, \ldots, N_c$, such that the following conditions are satisfied,

$$\nabla\mathcal{L}\left(\gamma^*,\lambda^*\right) \;=\; 0, \tag{5.3}$$

$$c_i\left(\gamma^*\right) \;\geq\; 0, \quad \forall i=1,\ldots,N_c \tag{5.4}$$

$$\lambda_i^* \;\geq\; 0, \quad \forall i=1,\ldots,N_c \tag{5.5}$$

$$\lambda_i^* c_i\left(\gamma^*\right) \;=\; 0, \quad \forall i=1,\ldots,N_c \tag{5.6}$$

where $c_i\left(\gamma\right) = \mu_i\gamma_{i,n} - \sqrt{\gamma_{i,u}^2+\gamma_{i,w}^2}$. These equations can be stated in a more manageable form by defining the vector s as follows:

$$\mathbf{s} = \left[\ldots,\lambda_i\mu_i,\frac{-\lambda_i\gamma_{i,u}}{\sqrt{\gamma_{i,u}^2+\gamma_{i,w}^2}},\frac{-\lambda_i\gamma_{i,w}}{\sqrt{\gamma_{i,u}^2+\gamma_{i,w}^2}},\ldots\right]^T \in \mathbb{R}^{3N_c} \tag{5.7}$$

Then, the condition in Eq. 5.3 can be expressed as follows:

$$\boldsymbol{N}\gamma+\mathbf{r}+\mathbf{a} \;=\; \mathbf{s} \tag{5.8}$$

which, equivalently can be expressed as

$$\boldsymbol{D}^T\boldsymbol{M}^{-1}\boldsymbol{D}\gamma+\boldsymbol{D}^T\boldsymbol{M}^{-1}\mathbf{k}+\mathbf{b}+\mathbf{a} \;=\; \mathbf{s} \tag{5.9}$$

$$\boldsymbol{D}^T\boldsymbol{M}^{-1}\left(\boldsymbol{D}\gamma\right)+\mathbf{b}+\mathbf{a} \;=\; \mathbf{s} \tag{5.10}$$

$$\boldsymbol{D}^T\mathbf{v}+\mathbf{b}+\mathbf{a} \;=\; \mathbf{s} \tag{5.11}$$

Specifically, the rows of Eq. 5.8 associated with contact i are

$$\left(\boldsymbol{N}\gamma+\mathbf{r}+\mathbf{a}\right)_i = \boldsymbol{D}_i^T\mathbf{v}+\mathbf{b}_i+\mathbf{a}_i = \mathbf{s}_i \tag{5.12}$$

$$\begin{bmatrix} \boldsymbol{D}_{i,n}^T\mathbf{v}+\frac{1}{h}\Phi_i+\alpha_i \\ \boldsymbol{D}_{i,u}^T\mathbf{v} \\ \boldsymbol{D}_{i,w}^T\mathbf{v} \end{bmatrix} = \begin{bmatrix} \lambda_i\mu_i \\ \frac{-\lambda_i\gamma_{i,u}}{\sqrt{\gamma_{i,u}^2+\gamma_{i,w}^2}} \\ \frac{-\lambda_i\gamma_{i,w}}{\sqrt{\gamma_{i,u}^2+\gamma_{i,w}^2}} \end{bmatrix}. \tag{5.13}$$

For the case where $\gamma_{i,n} > 0$, the quantity in the first line of Eq. 5.13 must be equal to zero according to Eq. 3.8c and $\alpha_i = \lambda_i\mu_i$. By manipulating the second and third lines of Eq. 5.13, λ_i can be expressed as follows:

$$\lambda_i = \sqrt{\left(\boldsymbol{D}_{i,u}^T\mathbf{v}\right)^2+\left(\boldsymbol{D}_{i,w}^T\mathbf{v}\right)^2} \tag{5.14}$$

5.1 Anti-relaxation via iterative refinement

As the anti-relaxation term α_i relies on the value of $\mathbf{v}^{(l+1)}$, the approximation $\mathbf{v}_k = \boldsymbol{M}^{-1}\left(\mathbf{k} + \boldsymbol{D}\gamma_k\right)$, where the subscript k represents the k^{th} iterate in a process that aims to find $\mathbf{v}^{(l+1)}$, is used as γ_k approaches γ^* via an iterative scheme. Although this refinement can be incorporated in any iterative scheme, a gradient-descent method with an improved convergence rate of $\mathcal{O}\left(1/k^2\right)$ based on [**?**] was chosen for this work. The method in [**?**] was shown to be an 'optimal' first-order method for smooth problems [**?**] in terms of its performance among all first-order methods, up to a constant. Given an initial $\gamma_0 \in \mathbb{R}^n$ and letting $\mathbf{y}_0 = \gamma_0$ and $\theta_0 = 1$, the anti-relaxation via iterative refinement algorithm with the step size t_k can be summarized as

$$\begin{aligned}
\mathbf{v}_k &= \boldsymbol{M}^{-1}\left(\mathbf{k} + \boldsymbol{D}\gamma_k\right) && (5.15)\\
\gamma_{k+1} &= \Pi_{\mathcal{C}}\left(\mathbf{y}_k - t_k \nabla f\left(\mathbf{y}_k, \mathbf{v}_k\right)\right) && (5.16)\\
\theta_{k+1} &\text{ solves } \theta_{k+1}^2 = \left(1 - \theta_{k+1}\right)\theta_k^2 && (5.17)\\
\beta_{k+1} &= \frac{\theta_k\left(1 - \theta_k\right)}{\theta_k^2 + \theta_{k+1}} && (5.18)\\
\mathbf{y}_{k+1} &= \gamma_{k+1} + \beta_{k+1}\left(\gamma_{k+1} - \gamma_k\right) && (5.19)
\end{aligned}$$

where $\nabla f\left(\mathbf{y}_k, \mathbf{v}_k\right) = \boldsymbol{N}\gamma + (\mathbf{r} + \mathbf{a})$. When $f\left(\mathbf{x}\right)$ is convex and Lipschitz continuous with constant L, then the method described by Equations 5.15-5.19 converges for any $t_k \leq 1/L$. The complete algorithm obtained when applying the APGD method with anti-relaxation to the problem in Eq. 5.1 is shown below. As stated, the algorithm includes an adaptive step size which may both shrink and grow, an adaptive restart scheme based on the gradient, and a fall-back strategy to allow early termination [**?**].

```
ALGORITHM APGD WITH ANTI-RELAXATION($\boldsymbol{N}$, $\mathbf{r}$, $\tau$, $N_{max}$)
$\gamma_0 = \mathbf{0}_{n_c}$
$\hat{\gamma}_0 = \mathbf{1}_{n_c}$
$\mathbf{y}_0 = \gamma_0$
$\mathbf{r}_0 = \mathbf{r}$
$\theta_0 = 1$
$L_k = \frac{||\boldsymbol{N}(\gamma_0 - \hat{\gamma}_0)||_2}{||\gamma_0 - \hat{\gamma}_0||_2}$
$t_k = \frac{1}{L_k}$
for $k := 0$ to $N_{max}$
    $\mathbf{g} = \boldsymbol{N}\mathbf{y}_k + \mathbf{r}$
    $\gamma_{k+1} = \Pi_{\mathcal{K}}\left(\mathbf{y}_k - t_k g\right)$
    while $\frac{1}{2}\gamma_{k+1}^T \boldsymbol{N} \gamma_{k+1} + \gamma_{k+1}^T \mathbf{r} \geq \frac{1}{2}\mathbf{y}_k^T \boldsymbol{N} \mathbf{y}_k + \mathbf{y}_k^T \mathbf{r} + \mathbf{g}^T\left(\gamma_{k+1} - \mathbf{y}_k\right) + \frac{1}{2} L_k ||\gamma_{k+1} - \mathbf{y}_k||_2^2$
        $L_k = 2L_k$
        $t_k = \frac{1}{L_k}$
        $\gamma_{k+1} = \Pi_{\mathcal{K}}\left(\mathbf{y}_k - t_k g\right)$
    endwhile
    $\theta_{k+1} = \frac{-\theta_k^2 + \theta_k\sqrt{\theta_k^2 + 4}}{2}$
    $\beta_{k+1} = \theta_k \frac{1 - \theta_k}{\theta_k^2 + \theta_{k+1}}$
    $\mathbf{y}_{k+1} = \gamma_{k+1} + \beta_{k+1}\left(\gamma_{k+1} - \gamma_k\right)$
    $r = r\left(\gamma_{k+1}\right)$
    if $r < \epsilon_{min}$
        $r_{min} = r$
        $\hat{\gamma} = \gamma_{k+1}$
    endif
    if $r < \tau$
        break
    endif
    if $\mathbf{g}^T\left(\gamma_{k+1} - \gamma_k\right) > 0$
        $\mathbf{y}_{k+1} = \gamma_{k+1}$
        $\theta_{k+1} = 1$
    endif
    $L_k = 0.9L_k$
    $t_k = \frac{1}{L_k}$
    $\mathbf{v}_k = \boldsymbol{M}^{-1}\left(\mathbf{k} + \boldsymbol{D}\gamma_{k+1}\right)$
    for $i = 1$ to $n_c$
        $\lambda_i = \sqrt{\left(\boldsymbol{D}_{i,u}^T \mathbf{v}_k\right)^2 + \left(\boldsymbol{D}_{i,w}^T \mathbf{v}_k\right)^2}$
        $\mathbf{r} = \mathbf{r}_0$
        $\mathbf{r}_{i,n} = \mathbf{r}_{i,n} + \lambda_i \mu_i$
    endfor
endfor
return Value at time step $t_{l+1}$, $\gamma^{l+1} := \hat{\gamma}$ .
```

5.2 Numerical experiments

This section analyzes the effect of anti-relaxation using three different numerical experiments. The first numerical experiment in §5.2.1 thoroughly analyzes the simple case of a sphere transitioning from pure sliding to rolling. The second experiment in §4.2.2 investigates the effect of anti-relaxation over time for a filling test with $1,000$ spheres. The last experiment in §5.2.4 looks at several filling tests, each with a different number of bodies, to investigate the effect of the anti-relaxation as a function of the number of collisions.

5.2.1 A 2D example

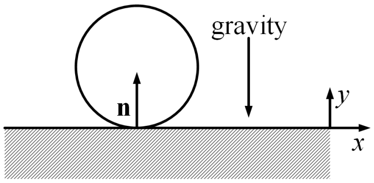


Figure 5.1: Sphere on a fixed horizontal plane.

The first numerical experiment thoroughly analyzes the dynamics of a 3D ball (two translational and one rotational degrees of freedom, $\mathbf{q} = [q_x, q_y, \omega]^T \in \mathbb{R}^3$) as it transitions from pure sliding to rolling [**?**]. In this case, $\gamma = [\gamma_n, \gamma_t]^T \in \mathbb{R}^2$ because there is only a normal and tangential impulse. Fig. 5.1 shows an elevation view of a rough uniform sphere of unit radius and mass in contact with a fixed horizontal plane in a uniform gravitational field. This example was chosen because of the existence of an easily obtainable closed-form solution of the dynamic motion of the sphere for certain conditions. Let the plane coincide with the xz-plane of a (right-handed) inertial frame, with the inertial y-direction upward. Thus, there will be a single contact with the $\mathbf{n}$ always pointed in the y-direction, the $\mathbf{u}$ always pointed in the x-direction (there is no $\mathbf{w}$ in this example), and the gap function $\Phi = q_y - 1$ so that $\mathbf{b} = \left[\frac{1}{h}(q_y - 1), 0\right]^T \in \mathbb{R}^2$. The coefficient of friction was assumed to have a constant value of 0.2. The matrices $\boldsymbol{D}$ and $\boldsymbol{M}$ can be seen to be constant throughout the motion. For this problem, the various matrices are:

$$\boldsymbol{M} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.4 \end{bmatrix} \tag{5.20}$$

$$\boldsymbol{D} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{5.21}$$

$$\mathbf{f_{ext}} = \begin{bmatrix} 0 \\ -9.81 \\ 0 \end{bmatrix} \tag{5.22}$$

Given the following initial conditions:

$$\mathbf{q}_0 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T \tag{5.23}$$

$$\mathbf{v}_0 = \begin{bmatrix} 2 & 0 & 0 \end{bmatrix}^T, \tag{5.24}$$

using $h = 0.01$ s the matrices $\boldsymbol{N}$ and $\mathbf{r}$ can be calculated from Eqs. 3.11a and 3.11b and will result in:

$$\boldsymbol{N} = \begin{bmatrix} 1 & 0 \\ 0 & 3.5 \end{bmatrix} \tag{5.25}$$

$$\mathbf{r} = \begin{bmatrix} -0.0981 \\ 2 \end{bmatrix} \tag{5.26}$$

The sphere initially slides in the x-direction, gathering angular velocity until the transition time, $t_{trn} = \frac{2v_{x0}}{7g\mu} \approx 0.291$ s where v_{x_0} is the initial velocity in the x-direction and $g = 9.81$ m/s^2. After the transition time, the sphere rolls with constant velocity in the x-direction and angular velocity

$\omega = \frac{5v_{x0}}{7} \approx 1.429$ m/s. The analytical solution for the contact impulses at $t = 0$ s can be shown to be $\gamma_{analytical} = [0.0981, -0.01962]^T$. Although the anti-relaxed method matched the analytical solution, shown in Fig. 5.3, using the relaxed CCQO (Fig. 5.2) of Eq. 3.12 results in incorrect behavior, even at the initial time step.

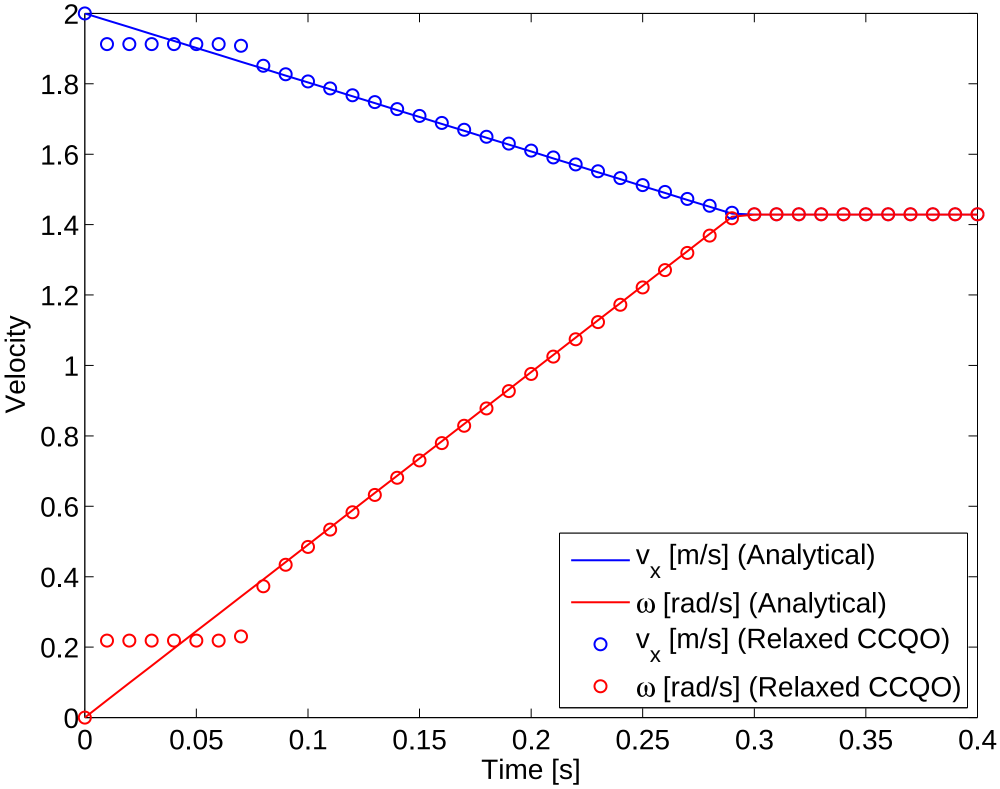


Figure 5.2: Rolling ball test for the analytical and numerical velocities with the relaxed CCQO.

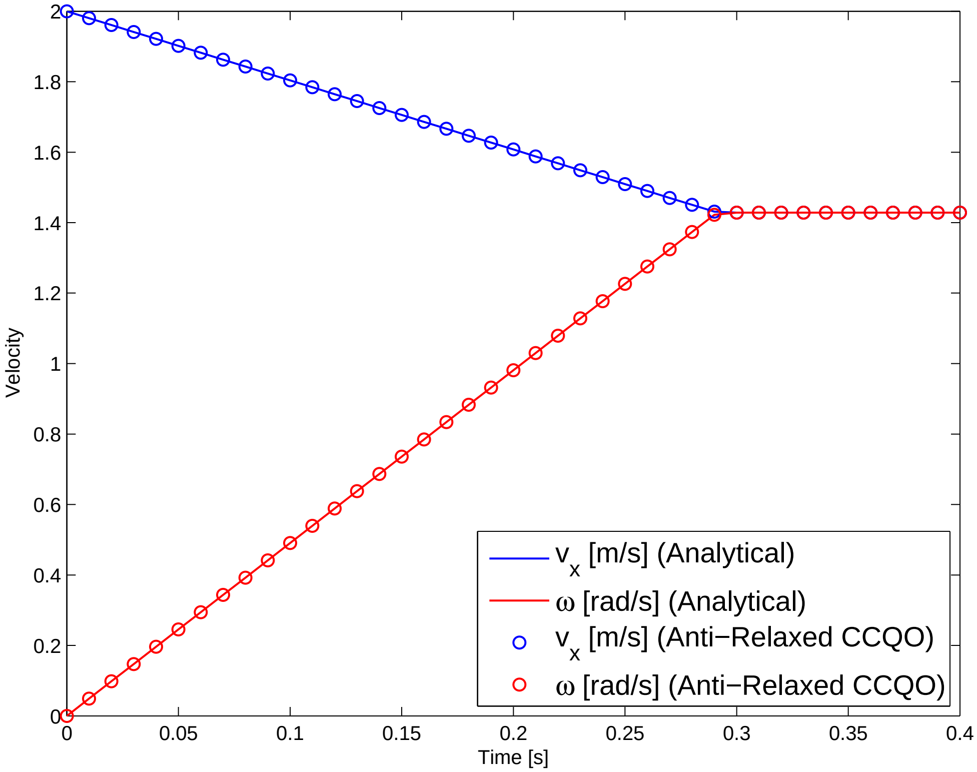
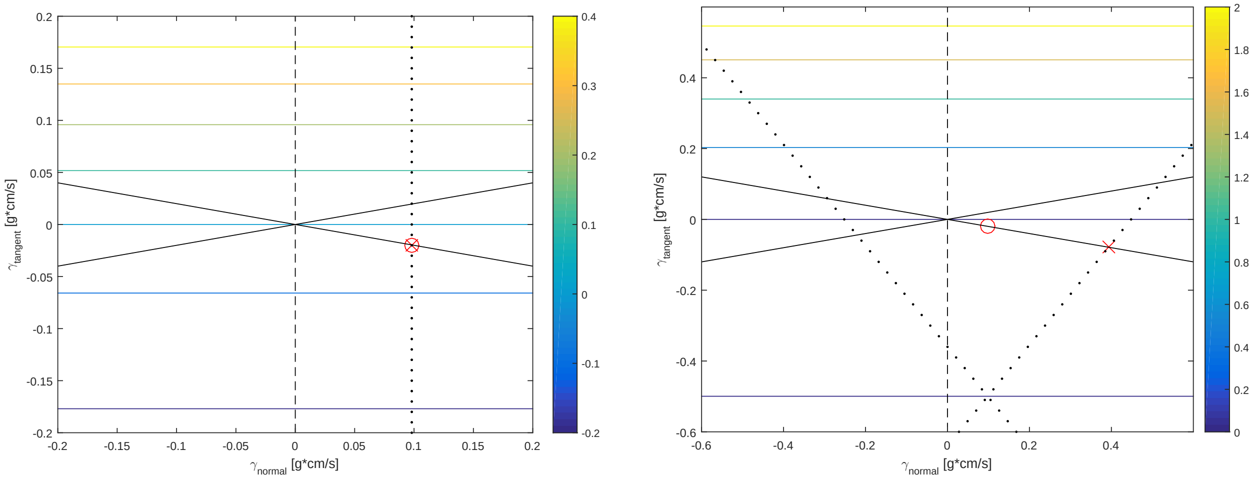


Figure 5.3: Rolling ball test for the analytical and numerical velocities with the anti-relaxed CCQO.

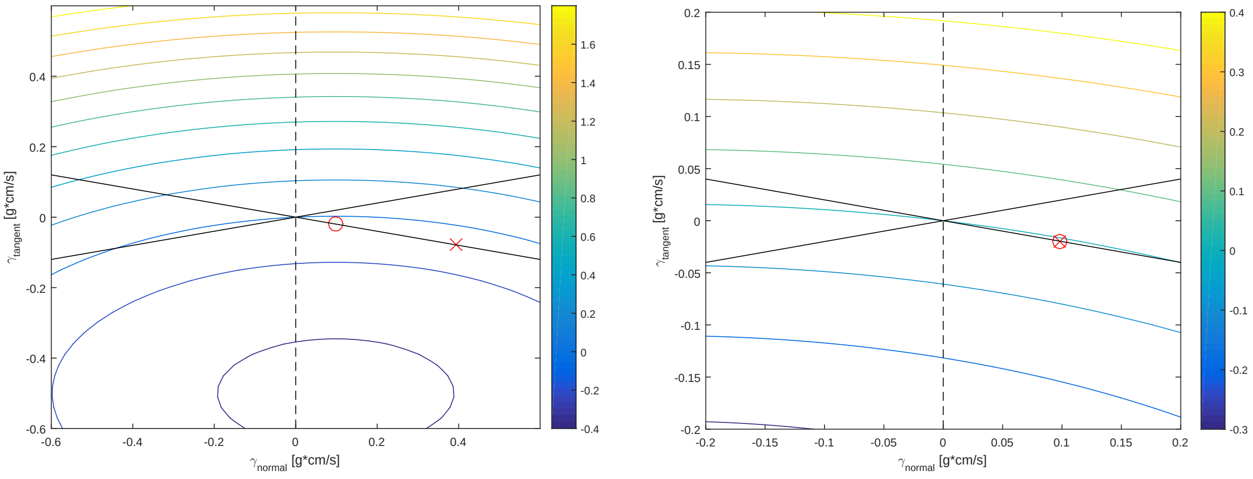
To graphically understand the non-physical behavior that is occurring at $t = 0$ s, four situations have been plotted in Figs. 5.4a-5.5b. Fig. 5.4a represents the original DVI that must be solved at $t = 0$ s plotted in the γ_n-γ_t domain by substituting the velocity terms for contact impulses using Eq. 3.8a. The contours, shown by the colored lines, represent the objective function in Eq. 3.8d, the cone constraints of which are shown by the solid black lines (the solution must lie in the interior of these lines). Lastly, the complementarity condition in Eq. 3.8c is captured by the dotted and dashed lines. If $\gamma_n = 0$, then γ_t must lie on the dashed line; if γ_n is nonzero, the solution must lie on the dotted line. The optimum and analytical solutions are indicated by the red X and circle, respectively. As this is the true DVI, the optimum and analytical solutions match. Fig. 5.4b shows what happens when Eq. 3.8c is replaced with the relaxed complementarity condition in Eq. 3.9. The complementarity condition associated with normal velocities bifurcates, resulting in the optimum being increased in magnitude. Physically, this would result in the ball being launched off of the ground. The relaxed CCQO of Eq. 3.12 is shown in Fig. 5.5a. Here the contours represent the quadratic objective function and the solid black lines represent the conic constraints (the solution must lie in the interior of these lines). The dashed line represents the non-negativity constraint on γ_n - the solution must lie on or to the right of this line. Since the relaxed CCQO is equivalent to the relaxed DVI/CCP, the optimum (represented by the red circle) will be identical to the one in Fig. 5.4b. Notice that, like the relaxed DVI/CCP, the optimum does not match the analytical solution to the problem. Lastly, using anti-relaxation from Eq. 5.1 results in a shift of the contours of the objective function in the negative γ_n-direction. This shift results in an optimum that coincides with the analytical solution. Additional surface plots of the objective functions and constraints can be viewed in Appendix B.



(a)

(b)

Figure 5.4: The original DVI problem based on Eq. 3.8 (left), and the relaxed DVI/CCP problem based on Eq. 3.9 (right).



(a) (b)

Figure 5.5: The relaxed CCQO based on Eq. 3.12 (left), and the anti-relaxed CCQO based on Eq. 5.1 (right).

5.2.2 Filling test

The second numerical experiment studies a model of spherical bodies falling into a container under gravity, the time evolution of which can be seen in Fig. 4.4. The goal of this study was to see how anti-relaxation effects the physical results of the simulation. Several statistics were monitored, such as the computation time, number of iterations performed at each step, maximum velocity of the bodies, and contact force on the container. In this case, there are $1,000$ spheres, each with a radius $r = 1$ m, mass $m = 1$ kg, and friction coefficient $\mu = 0.25$. The system had a gravitational acceleration in the vertical direction $g = 9.81$ m/s^2. The simulation was run for 5 s with a time step $h = 0.01$ s and a solver tolerance of $\tau = 1 \times 10^{-5}$.

The results of the simulation are plotted in Fig. 5.6 and 5.7. Fig. 5.6 shows the total contact force that the container experiences as a function of time. The dynamics behavior as determined by the relaxed CCQO (using Eq. 3.12) is represented by the blue line and dynamics behavior as determined by the anti-relaxed CCQO (using Eq. 5.1) is represented by the red line. Although both models converge to the total weight of the spheres, the dynamics are clearly different and the anti-relaxed solution settles much sooner than the relaxed solution. The complementarity error in the anti-relaxed solution, calculated using Eq. 3.8c and shown in Fig. 5.7, is several orders of magnitude less than the relaxed solution.

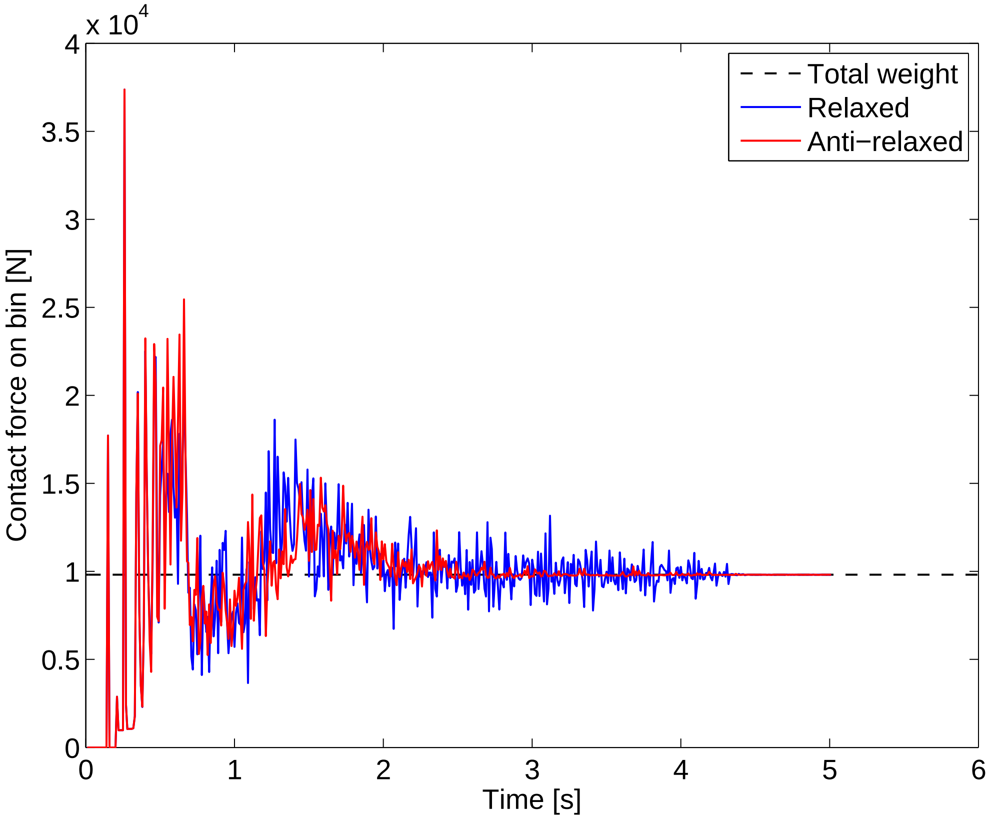


Figure 5.6: The contact force experienced by the container as a function of time.

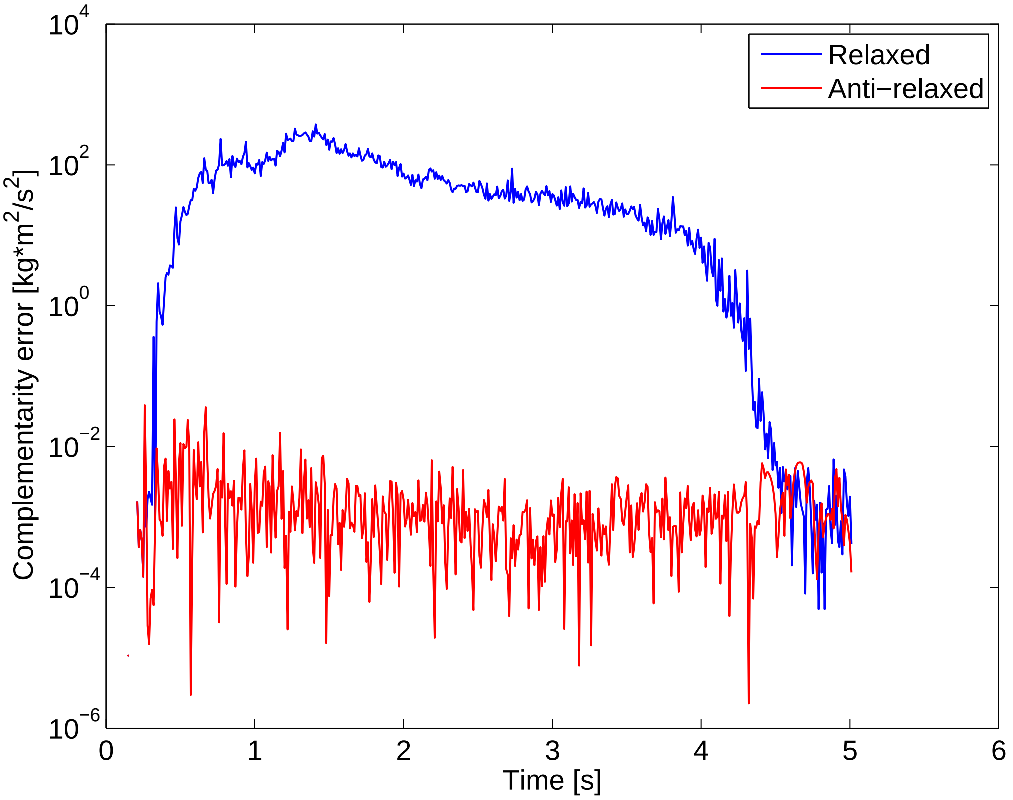


Figure 5.7: The complementarity error as a function of time.

5.2.3 Compaction test

The third numerical experiment studies a model of spherical bodies in a container being compacted by a slab, the time evolution of which can be seen in Fig. 4.9. The goal of this study was to see how anti-relaxation effects the physical results of the simulation. Several statistics were monitored, such as the computation time, number of iterations performed at each step, and contact force on the container. In this case, there are $1,000$ spheres, each with a radius $r = 1$ m, mass $m = 1$ kg, and friction coefficient $\mu = 0.25$. The upper slab prevents spheres from escaping the container and has a mass $m = 4,000$ kg and friction coefficient $\mu = 0.25$. The system has a gravitational acceleration in the vertical direction $g = 9.81$ m/s^2. The simulation was run for 4 s with a time step $h = 0.001$ s and a solver tolerance of $\tau = 1 \times 10^{-5}$.

The results of the simulation are plotted in Fig. 5.8 and 5.9. Fig. 5.8 shows the total contact force that the container experiences as a function of time. The dynamics behavior as determined by the relaxed CCQO (using Eq. 3.12) is represented by the blue line and dynamics behavior as determined by the anti-relaxed CCQO (using Eq. 5.1) is represented by the red line. Although both models converge to the total weight of the spheres and slab, the dynamics are clearly different and the anti-relaxed solution settles much sooner than the relaxed solution. The complementarity error in the anti-relaxed solution, calculated using Eq. 3.8c and shown in Fig. 5.9, is several orders of magnitude less than the relaxed solution.

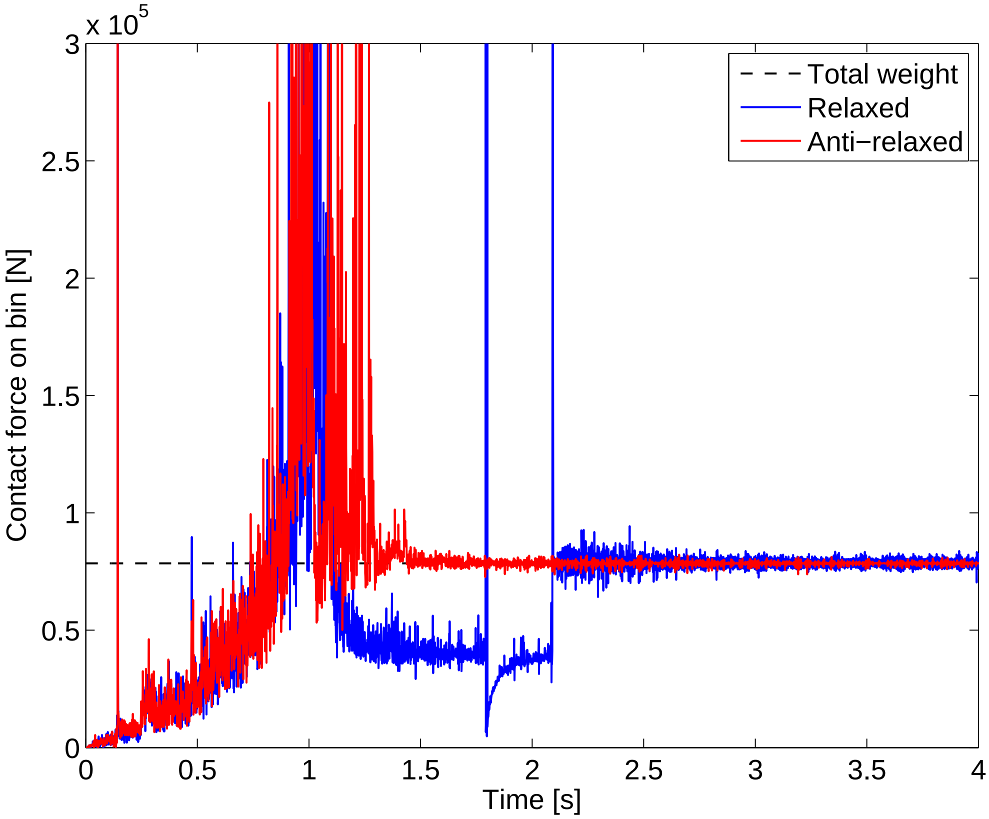


Figure 5.8: The contact force experienced by the container as a function of time.

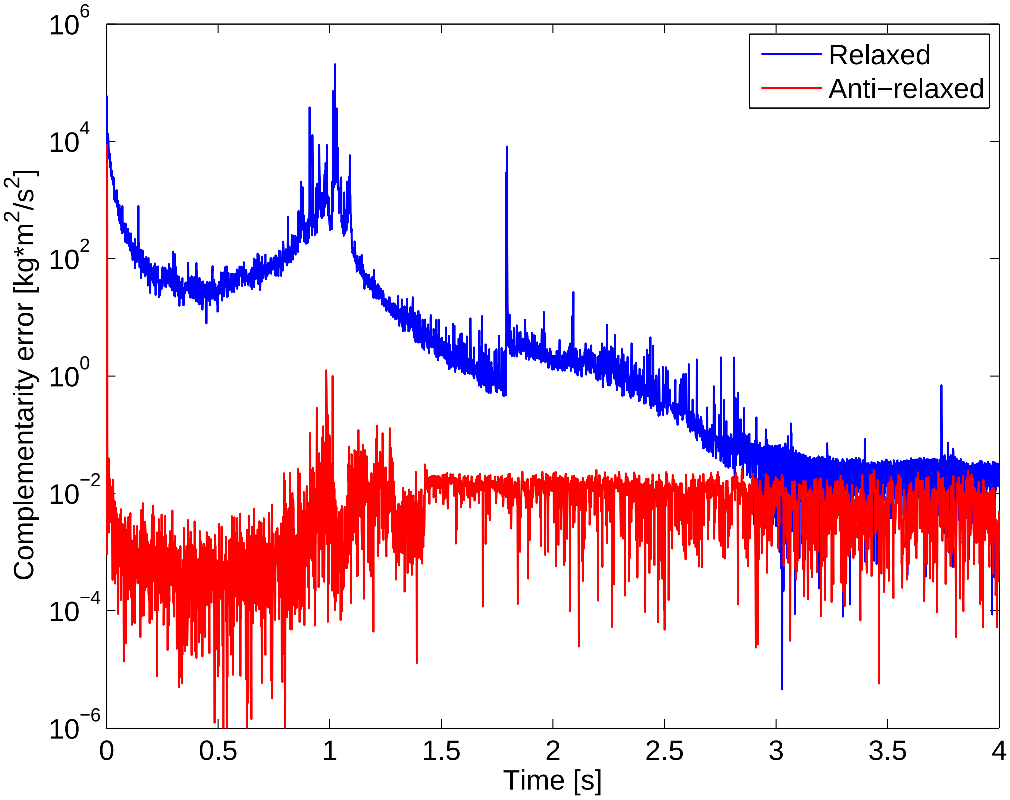


Figure 5.9: The complementarity error as a function of time.

5.2.4 Scaling analysis

The last numerical experiment studies several models of spherical bodies falling into a container under gravity to understand how the solver statistics change as a function of the number of collisions. Several statistics were monitored, such as the computation time, number of iterations performed at each step, maximum velocity of the bodies, and contact force on the container. The number of spheres is varied from 10 to $16,000$, each with a radius $r = 1$ m, mass $m = 1$ kg, and friction coefficient $\mu = 0.25$. The system had a gravitational acceleration in the vertical direction $g = 9.81$ m/s^2. Each simulation was run for 5 s with a time step $h = 0.01$ s and a solver tolerance of $\tau = 1 \times 10^{-5}$.

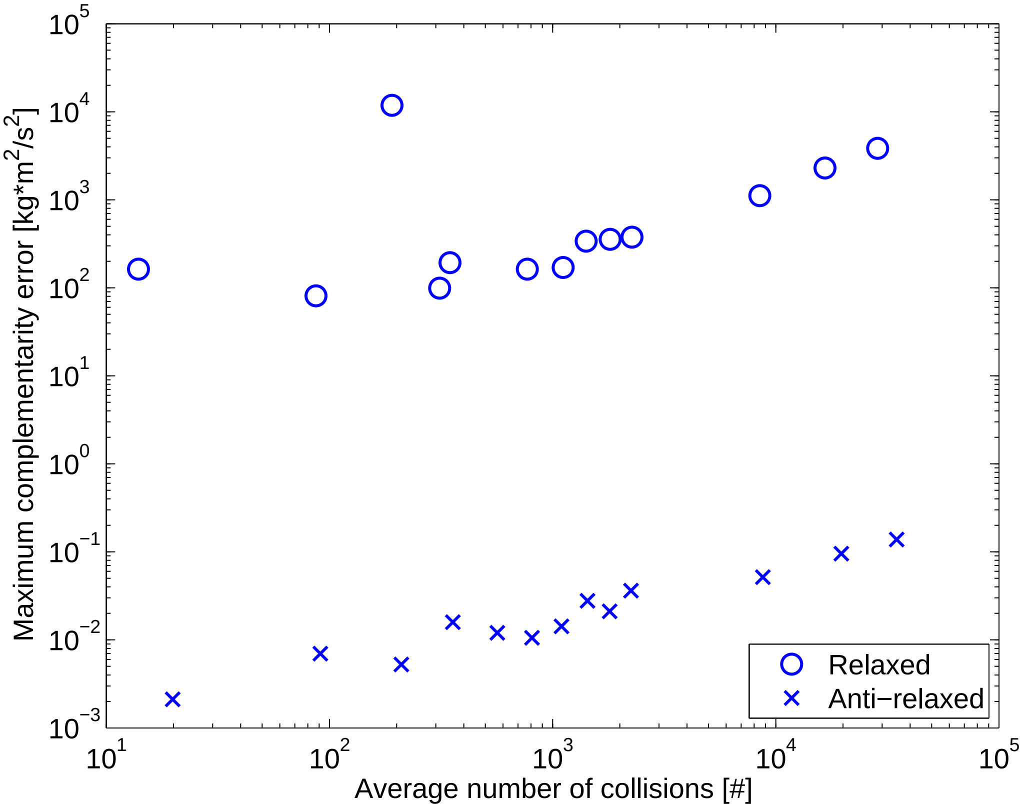


Figure 5.10: The maximum complementarity error as a function of the number of collisions.

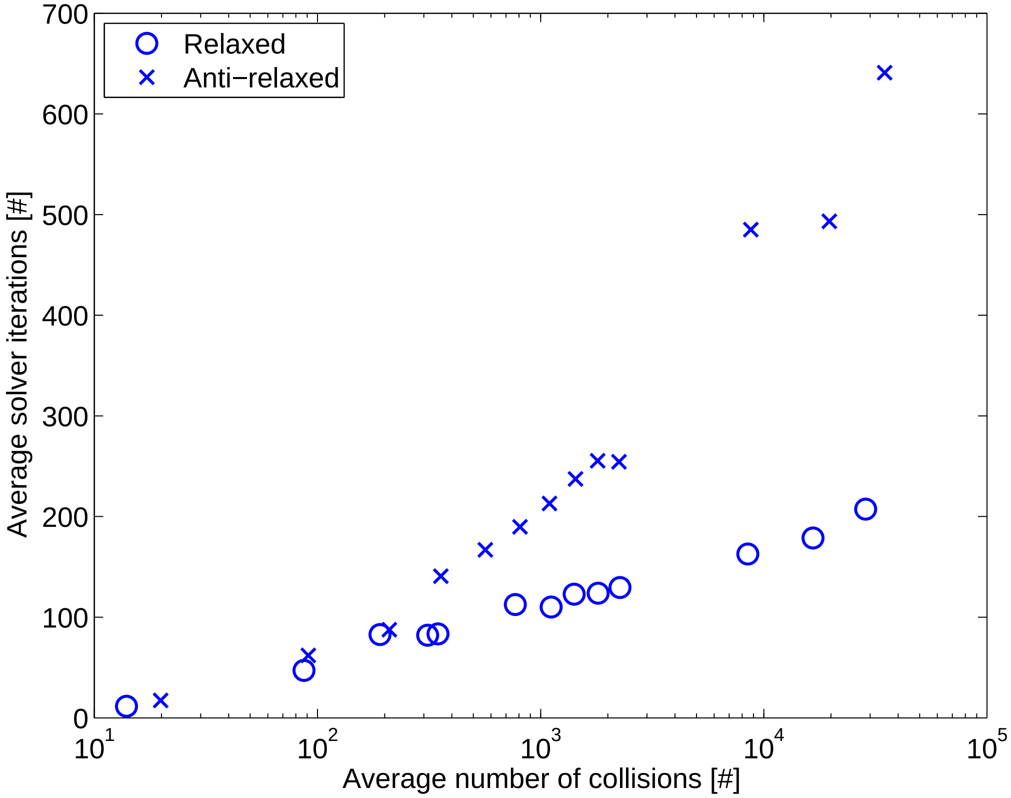


Figure 5.11: The average number of iterations as a function of the number of collisions.

The results of the simulation are plotted in Fig. 5.10 and 5.11. Fig. 5.10 shows the maximum complementarity error (calculated using Eq. 3.8c) that occurs over the entire simulation. It is important to note that the complementarity error in both models increases as the number of collisions increases, although the error in the anti-relaxed solution is again several orders of magnitude lower than the relaxed solution. Fig. 5.11 shows the average number of APGD iterations that are used to solve the CCQO over the entire simulation. Although the number of iterations are similar for low numbers of collisions (below 100 spheres), the anti-relaxed solution quickly becomes increasingly costly as the number of collisions increases. Despite this increase in computational cost, it makes sense to use the anti-relaxed approach, especially for scenarios with few contacts. In these cases, the computational cost is only slightly higher, yet the quality of the solution is significantly improved.

Chapter 6

Implications in flexible multibody dynamics

This chapter discusses how a DVI formulation can be used to model frictional contact between flexible bodies formulated with the ANCF method to simulate large flexible multibody systems. It is important to note that the DVI method can be coupled with any flexible body methodology, however, the ANCF approach has been used successfully in many multibody systems applications that involve large deformation and rotation [**?**]. The penalty-based approach, especially the Hertzian contact model, has been used to model frictional contact between ANCF beams in [**?**, **?**]. The penalty-based approach, though easy to implement, requires a very small integration step-size to maintain the stability and accuracy of the numerical solution. Moreover, it is always challenging to find the values of penalty parameters which minimize contact constraint violation. To overcome these drawbacks, a DVI-based approach for the analysis of frictional contact problem between flexible bodies is introduced. The DVI formulation has been used in modeling frictional contact between rigid bodies in the context of simulation of granular particles in [**?**]. In this work, this approach is extended to model the frictional contact between the ANCF plate/shell elements.

The velocity-impulse based time stepping scheme used in the DVI approach is combined with a Coulomb friction model to formulate a CCP [**?**]. At each time step, the solution of the CCP gives the reaction impulses for all the active contacts and generalized nodal velocities. For the case of frictionless contact the DVI formulation can be posed as a LCP which can be easily solved using standard iterative solvers. In this work we use a spherical decomposition approach (for details see [**?**, **?**]), which allows for self contact and multiple contacts between the flexible bodies.

Being computationally very intensive, the DVI and ANCF methodologies stand to benefit from the use of parallel computation. In the simulation of complex mechanical systems with many flexible beams (e.g. hair or polymer simulation), the equations of motion of each beam can be solved in parallel. The computation of the nonlinear internal forces as well as the external forces can also be done in parallel at the element level. Using the spherical decomposition approach, contacts between flexible bodies can be easily detected in parallel, and the complementarity problem can also be solved in a parallel for each active contact. All these aspects are anticipated to lead to reductions in simulation times for large flexible body systems.

6.1 Absolute nodal coordinate formulation

To model the friction contact between flexible bodies, a spherical decomposition approach is used for the contact discretization of the colliding bodies. In the spherical decomposition approach, each flexible body can be considered as a union of spheres that overlap and are distributed equidistantly across the surface of the elements. At each time step, the collision detection between the spheres in all the plates is performed and a signed distance $\Phi\left(\mathbf{q},t\right)$ and contact normal **n** is determined for each active contact. Note that the contact normal is along a line joining the centers of colliding spheres. The contacts are considered active if $\Phi\left(\mathbf{q},t\right)\leq 0$.

ANCF is useful for the large deformation analysis of flexible bodies in multibody system applications that are characterized by large rigid body rotations [**?**]. This approach is contrary to the typical general purpose finite element codes in that it uses a non-incremental solution procedure. Incremental finite element analyses are known to have problems with error accumulation over time due to linearization. Since it is desired to simulate many-body systems over long periods, the non-incremental nature of ANCF is attractive [**?**]. ANCF is ideal for multibody dynamics due to its lack of Coriolis effects and centrifugal forces, its uniquely defined rotation field, and its constant mass matrix.

The locking problems of fully parameterized ANCF finite elements based on the continuum mechanics approach have been addressed in [**?**, **?**, **?**]. These locking problems deteriorate the performance of ANCF finite elements especially for thin and stiff structures. To avoid these locking problems, the use of the elastic line approach along with the Hellinger-Reissner principle was proposed in [**?**] and the use of higher order elements was suggested in [**?**]. The coupled deformation modes in ANCF, which lead to numerical instabilities for thin and stiff structures, are discussed in [**?**]. The high frequencies that are induced along the thin direction of a plate element, which is the focus of this work, require an extremely small time step, resulting in longer simulation times. In the case where the aspect ratio (length divided by thickness) of the element is high, plane stress assumptions can be made that allow a reduced-order element to be accurate. Specifically, Kirchhoffs plate theory, which does not account for shear deformation, is used and results in a four-node, gradient-deficient plate element with 36 degrees of freedom, or nodal coordinates [**?**].

Each gradient-deficient plate element is composed of four nodes and each of these nodes have nine generalized coordinates: three to describe the position $\mathbf{r}^{n^T}$ of the node in Cartesian-coordinate, three to describe the slope of the plate in the local x-direction $\frac{\partial \mathbf{r}^{n^T}}{\partial x}$, and three to describe the slope of the plate in the local y-direction $\frac{\partial \mathbf{r}^{n^T}}{\partial y}$. The four nodes result in a total of 36 generalized coordinates per element, written as

$$\mathbf{e}^n = \begin{bmatrix} \mathbf{r}^{n^T} & \frac{\partial \mathbf{r}^{n^T}}{\partial x} & \frac{\partial \mathbf{r}^{n^T}}{\partial y} \end{bmatrix} \tag{6.1}$$

For thin plates, the deformation along the thickness of the plate can be neglected [**?**]. The normal vector of the mid-surface of an element can be defined by the cross product of the gradients in the x and y-directions at the location of the desired normal. The global position vector $\mathbf{r}$ of the material point P on the plate element can be defined by using the element shape functions and the nodal coordinate vector as follows

$$\mathbf{r} = \boldsymbol{S}\left(x, y, x\right)\mathbf{e} \tag{6.2}$$

where $\boldsymbol{S}$ is the element shape function matrix expressed in terms of the element spatial coordinates ξ and η. Specifically,

$$\boldsymbol{S} = [S_1\boldsymbol{I} \;\; S_2\boldsymbol{I} \;\; S_3\boldsymbol{I} \;\; S_4\boldsymbol{I} \;\; S_5\boldsymbol{I} \;\; S_6\boldsymbol{I} \;\; S_7\boldsymbol{I} \;\; S_8\boldsymbol{I} \;\; S_9\boldsymbol{I} \;\; S_{10}\boldsymbol{I} \;\; S_{11}\boldsymbol{I} \;\; S_{12}\boldsymbol{I}] \tag{6.3}$$

$$\boldsymbol{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.4}$$

$$\tag{6.5}$$

$$\begin{gathered}
S_1 = -(\xi - 1)(\eta - 1)(2\eta^2 - \eta + 2\xi^2 - \xi - 1) \\
S_2 = -l\xi(\xi - 1)^2(\eta - 1) \\
S_3 = -w\eta(\eta - 1)^2(\xi - 1) \\
S_4 = \xi(2\eta^2 - \eta + 3\xi + 2\xi^2)(\eta - 1) \\
S_5 = -l\xi^2(\xi - 1)(\eta - 1) \\
S_6 = w\xi\eta(\eta - 1)^2 \\
S_7 = -\xi\eta(1 - 3\xi - 3\eta + 2\eta^2 + 2\xi^2) \\
S_8 = l\xi^2\eta(\xi - 1) \\
S_9 = w\xi\eta^2(\eta - 1) \\
S_{10} = \eta(\xi - 1)(2\xi^2 - \xi - 3\eta + 2\eta^2) \\
S_{11} = l\xi\eta(\xi - 1) \\
S_{12} = -w\eta^2(\xi - 1)^2(\eta - 1)
\end{gathered} \tag{6.6}$$

These shape functions are used to translate between the element generalized coordinates and Cartesian coordinates. It is important to note that the gradient deficient plate element is not ideal. A consequence of ignoring the deformation along the thickness is that the element is non-conforming, meaning that the continuity between adjoined elements is not ensured.

Using the principle of virtual work of the continuum, the equation of motion is determined to be

$$\boldsymbol{M}\ddot{\mathrm{e}} + \boldsymbol{Q}_k = \boldsymbol{Q}_e \tag{6.7}$$

where $\boldsymbol{M}$ is the mass matrix, $\mathbf{Q}_k$ is the vector of internal forces, and $\mathbf{Q}_e$ is the vector of external forces. This equation is important because it governs the dynamics of the entire system and can be combined with algebraic equations to enforce constraints between nodes. The mass matrix is determined from the kinetic energy of the system and is defined as follows

$$\boldsymbol{M} = \int_{V_0} \rho \boldsymbol{S}^T \boldsymbol{S} dV_0 \tag{6.8}$$

which remains constant over time. The external force vector due to a concentrated force $\mathbf{f}$ is calculated using the following equation

$$\mathbf{Q}_e = \boldsymbol{S}^T \mathbf{f} \tag{6.9}$$

and the external force vector due to a force that acts over the entire volume of the element is given by:

$$\mathbf{Q}_e = \int_{V_0} \rho \boldsymbol{S}^T \mathbf{f} dV_0 \tag{6.10}$$

The strain energy of a gradient-deficient plate element is calculated from the strain ε and curvature κ using

$$U = \frac{1}{2} \int_{V_0} \varepsilon^T \mathbf{E} \varepsilon dV_0 + \frac{1}{2} \int_{V_0} \kappa^T \mathbf{E} \kappa dV_0 \tag{6.11}$$

where the strain is based on the non-linear Green-Lagrange strain measure [**?**] and the curvature is used to account for bending stiffness. The vector of internal forces is subsequently obtained by differentiating the strain energy with respect to the generalized coordinates

$$\mathbf{Q}_k = \left(\frac{\partial U}{\partial \mathbf{e}} \right)^T . \tag{6.12}$$

The above equations can be combined with the integration scheme described in §3, with the exception that the tangent space generators $\boldsymbol{D}_i = [\boldsymbol{D}_{i,n}, \boldsymbol{D}_{i,u}, \boldsymbol{D}_{i,w}] \in \mathbb{R}^{12n_b \times 3}$ are defined as

$$\boldsymbol{D}_{i,n}^T = \left[\mathbf{0}, \cdots, \mathbf{n}_i^T \boldsymbol{S}(\mathrm{s}_{i,A}), \mathbf{0}, \cdots, \mathbf{0}, -\mathbf{n}_i^T \boldsymbol{S}(\mathrm{s}_{i,B}), \cdots, \mathbf{0}\right] \tag{6.13a}$$

$$\boldsymbol{D}_{i,u}^T = \left[\mathbf{0}, \cdots, \mathbf{u}_i^T \boldsymbol{S}(\mathrm{s}_{i,A}), \mathbf{0}, \cdots, \mathbf{0}, -\mathbf{u}_i^T \boldsymbol{S}(\mathrm{s}_{i,B}), \cdots, \mathbf{0}\right] \tag{6.13b}$$

$$\boldsymbol{D}_{i,w}^T = \left[\mathbf{0}, \cdots, \mathbf{w}_i^T \boldsymbol{S}(\mathrm{s}_{i,A}), \mathbf{0}, \cdots, \mathbf{0}, -\mathbf{w}_i^T \boldsymbol{S}(\mathrm{s}_{i,B}), \cdots, \mathbf{0}\right] \tag{6.13c}$$

and are used to transform the contact forces from local to global frame, μ_i is the coefficient of friction for contact i, and N_c is the total number of possible contacts. In case of rigid bodies, the projection matrix has a different expression which is given in [**?**].

The collision detection problem can be a bottleneck in the simulation of physical systems involving a large number of bodies. For instance, for a polymer simulation problem, one has to consider systems with hundreds of thousands of flexible bodies interacting through friction and contact. The spherical decomposition approach simplifies the process of collision detection between millions of spheres that together make up the shape of the flexible bodies, as shown in Fig. 6.1. This collision detection task is performed once at each integration time step using the technique proposed in [**?**].

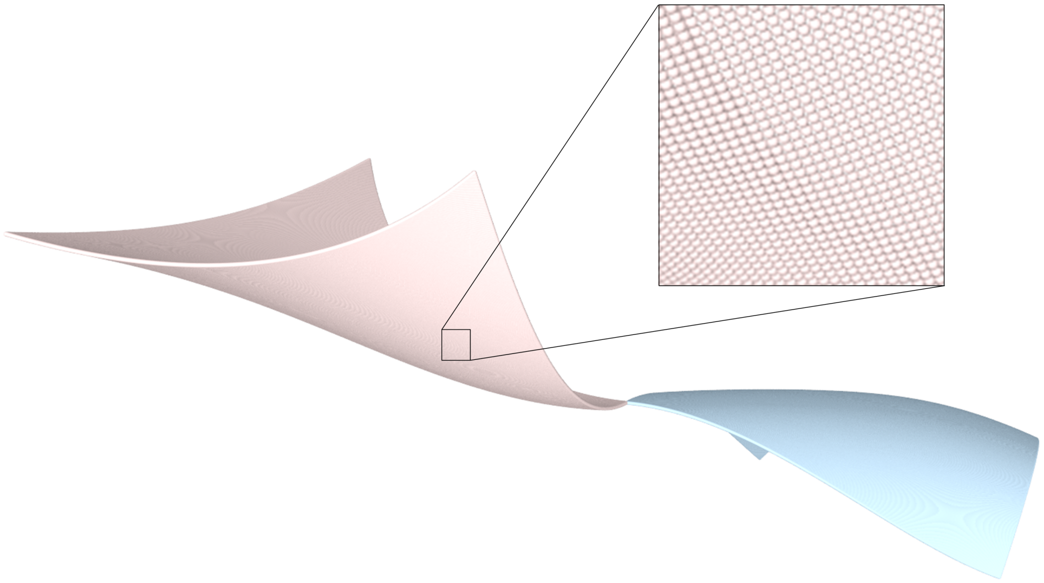


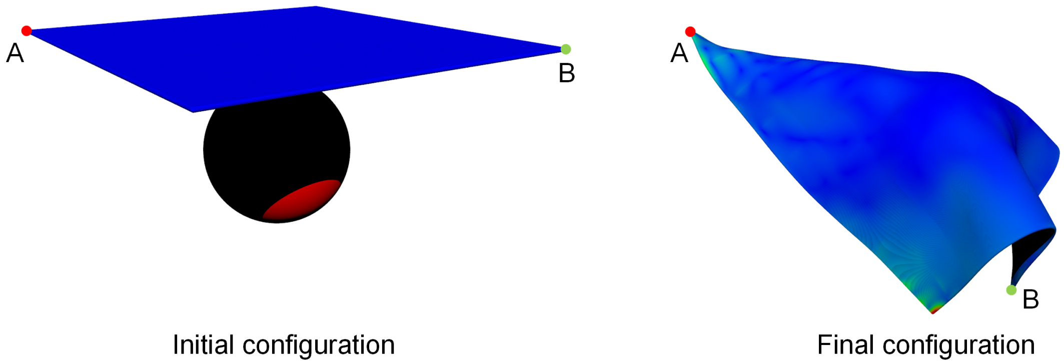
Figure 6.1: Spherical decomposition of a deformed mesh constructed from two plate elements. The decomposition of the plate geometry is used only for fast collision detection purposes.

6.2 Numerical experiments

This section reports on results obtained in the process of validating the solution methodology outlined in Section §6.1. As the gradient-deficient plate elements have been validated in previous work [**?**], numerical results focus on convergence as a function of solver properties.

6.2.1 Flexible cloth

Several instances of a flexible cloth model were dropped on a large, rigid sphere, shown in Fig. 6.2. The flexible plate is pinned at point *A* and the position of point *B* is monitored as a function of time. The plate has dimensions $1 \times 1 \times 0.01$ m for length, width, and thickness, respectively. The plate has a density $\rho = 7,200$ kg/m^3, an elastic modulus $E = 2 \times 10^6$ Pa, a friction coefficient $\mu = 0.25$, and a Poisson's ratio $\nu = 0.3$. The system has a gravitational field of $g = 9.81$ m/s^2 in the negative y-direction.



Initial configuration

Final configuration

Figure 6.2: A flexible cloth composed of gradient-deficient plate elements is dropped on a rigid sphere. The cloth is pinned at point *A* and the position of point *B* is monitored over time.

To test for the convergence of the gradient-deficient plate element, several flexible cloth models with varying numbers of elements were dropped. The steady-state y-position of point *B* as well as the steady state vertical contact force that the rigid ball experiences were plotted in Fig. 6.3 and 6.4, respectively. Along with varying the number of elements, the time step was decreased from

$h = 1 \times 10^{-3}$ to 1×10^{-5} s. Based on the results in Fig. 6.3 and 6.4, point *B* reaches a converged, steady-state position of -0.71 m and a steady-state contact force of -600 N after 7 elements per side, however, the $h = 1 \times 10^{-3}$ s case becomes unstable after 9 elements. Based on the results in Fig. 6.5 and 6.6, although the $h = 1 \times 10^{-3}$ s case requires the largest number of iterations per time step, the execution time is smaller than the $h =$ and $h = 1 \times 10^{-4}$ and 1×10^{-5} s cases due to the fewer number of required steps.

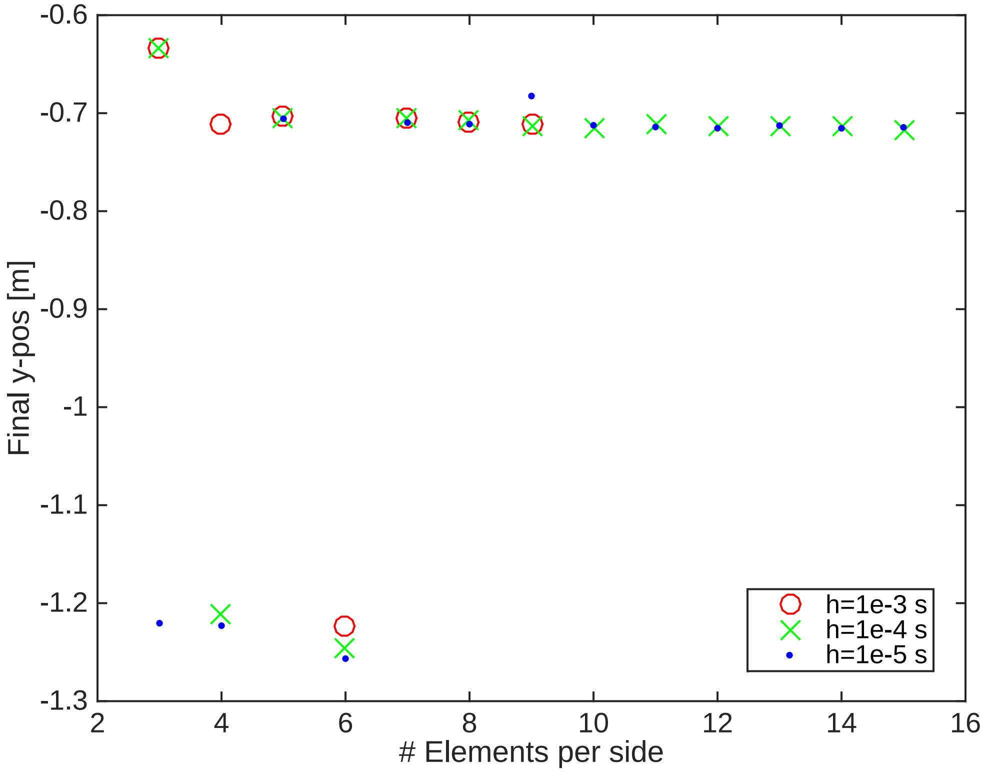


Figure 6.3: The steady-state value of the y-position of point *B* as a function of the number of elements used to model the cloth.

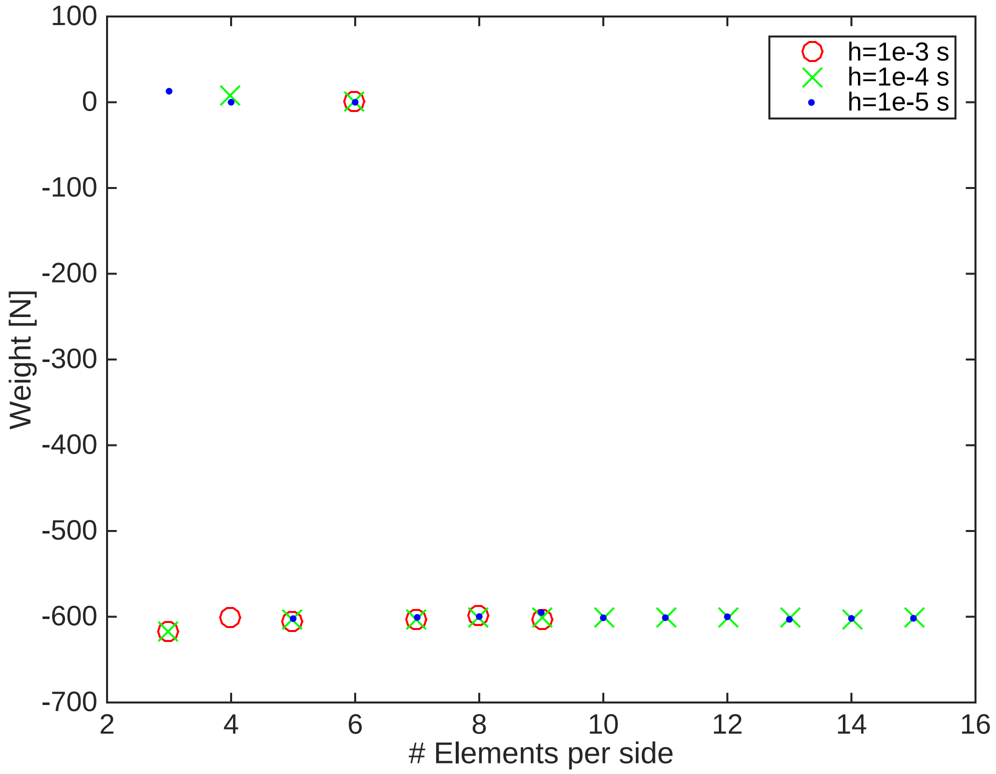


Figure 6.4: The steady-state value of the y-component of contact force as a function of the number of elements used to model the cloth.

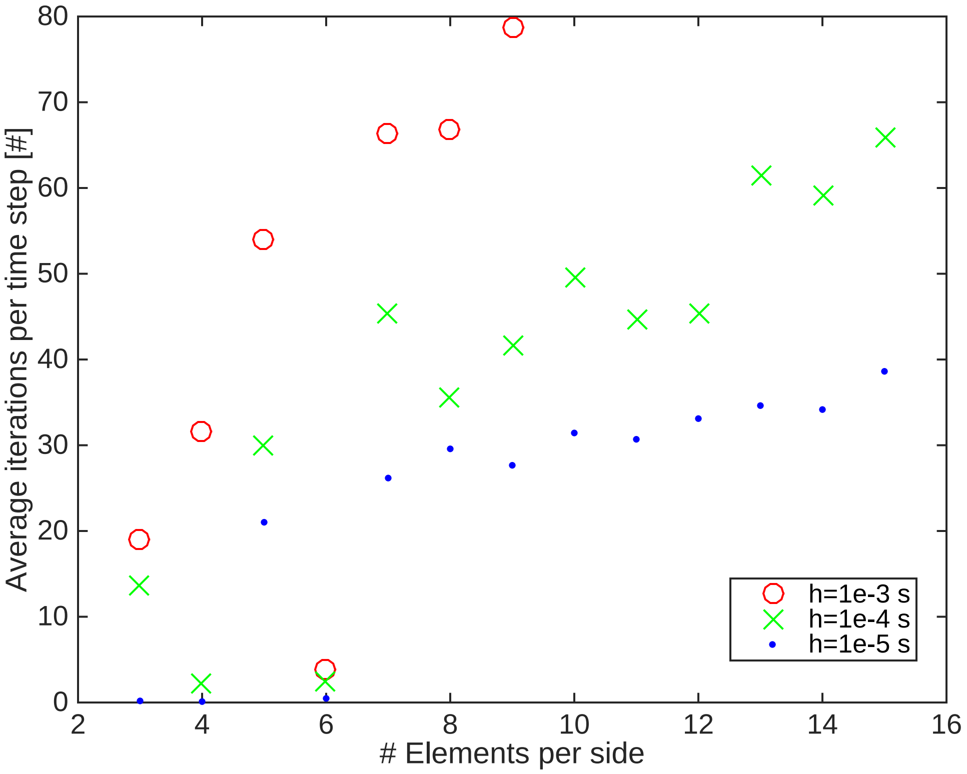


Figure 6.5: The average number of frictional contact iterations per time step as a function of the number of elements used to model the cloth.

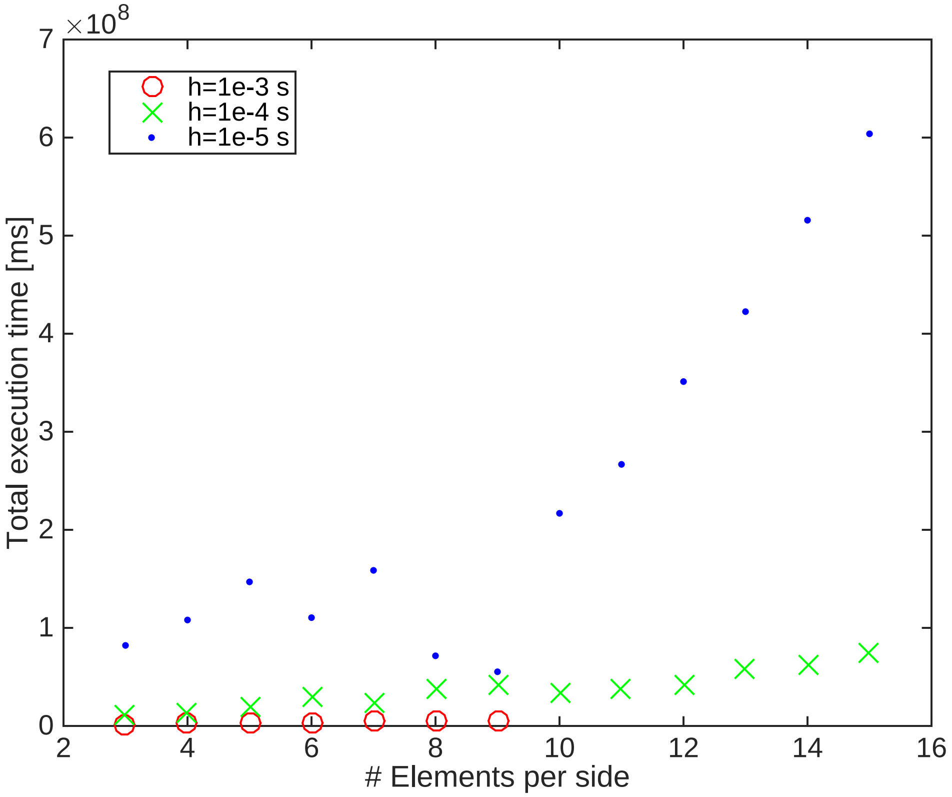


Figure 6.6: The total execution time as a function of the number of elements used to model the cloth. The simulations were run on an Intel Nehalem Xeon E5520 2.26GHz processor with an NVIDIA GeForce GTX 680 GPU.

6.2.2 Flexible tire

This section investigates a flexible tire, shown in Figs. 6.7 and 6.8, that is composed of gradient-deficient plate elements attached to a rigid rim. A single wheel test is used to investigate the tire's motion under controlled slip and normal loading conditions. The drawbar pull, torque, and rim position were measured for several cases to determine the effect that the number of lateral elements, the number of radial elements, the solver tolerance, the slip, and friction coefficient had on the tire performance. The tire used in this study had a width $b = 0.2$ m, an outer radius $r_o = 0.3$ m, and an inner radius $r_i = 0.15$ m. The elements had a density $\rho = 7,810$ kg/m^3, an elastic modulus $E = 2 \times 10^7$ Pa, and a Poisson's ratio $\nu = 0$. To produce a desired constant slip, the tire was rolled on rigid ground with a constant angular velocity of $\omega = 0.3$ rad/s and a certain fixed translational velocity v based on the slip defined as

$$v = (1.0 - slip)\,\omega\, r_o\ .$$

After settling for 3 seconds, the wheel was rolled at the desired slip ratio for $T_{final} = 15$ s.

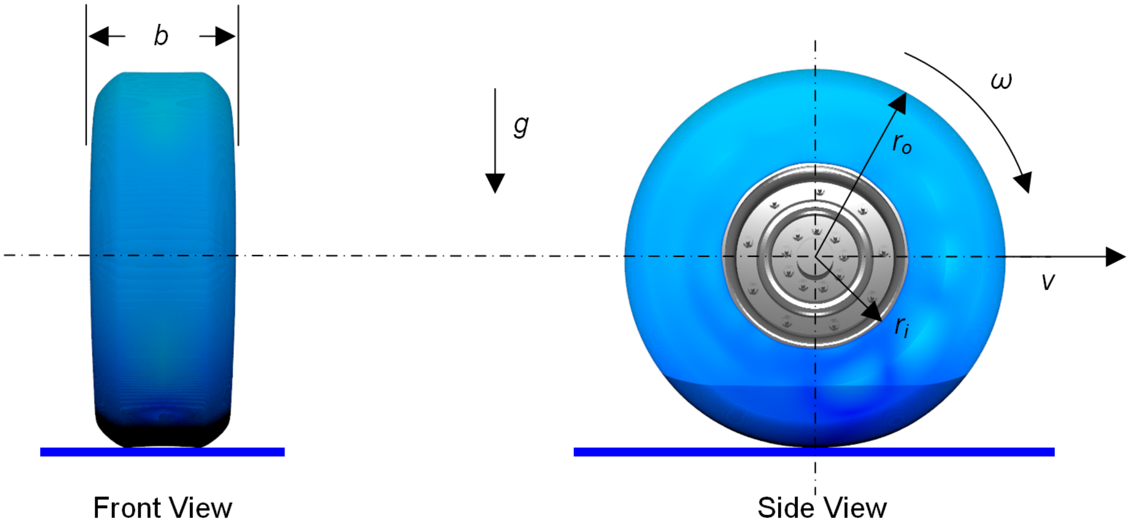


Figure 6.7: A flexible tire composed of gradient-deficient plate elements attached to a rigid rim.

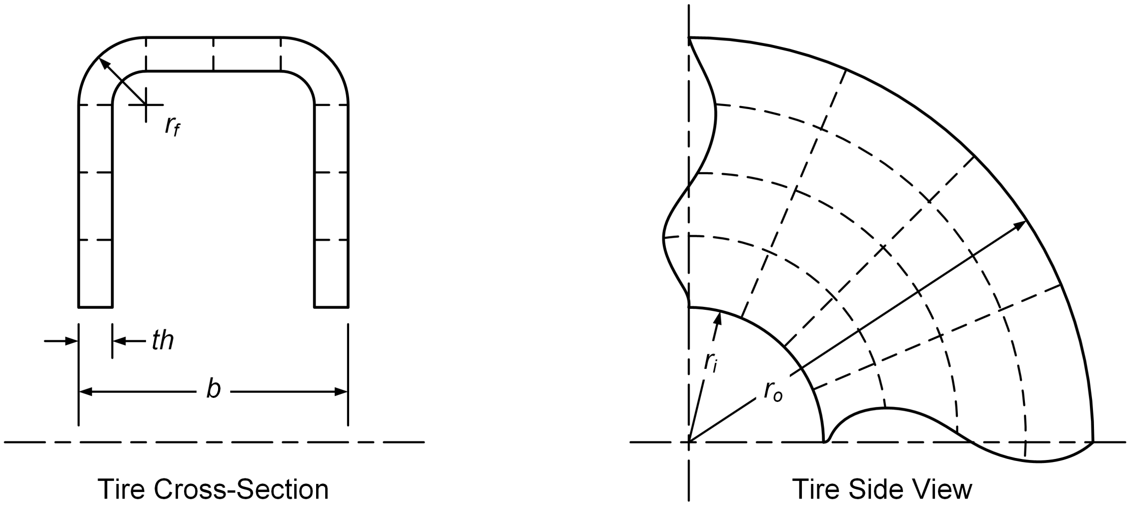


Figure 6.8: A schematic of the tire construction based on plate elements. The "lateral" elements refer to the number of elements along the width of the tire (shown in the cross-section, left) and the "radial" elements refer to the number of elements along the angular axis of the tire (shown in the side view, right).

6.2.2.1 Bounce test

A bounce test was performed to determine the required number of lateral elements, defined as the number of elements along the width b of the tire. A flexible tire was dropped from a height of 5 cm and allowed to come to rest for a varying number of lateral elements. To eliminate radial and solver effects, the tire was divided into 13 elements about the angular axis (or hub) of the tire and a solver tolerance of $\tau = 1 \times 10^{-4}$ was used. The height of the center of mass of the rim and the vertical contact force experienced by the ground are plotted in Fig. 6.10 and 6.9, respectively. Although the contact force of the tire converges after 6 lateral elements, Fig. 6.10 indicates that at least 9 lateral elements are required for convergence.

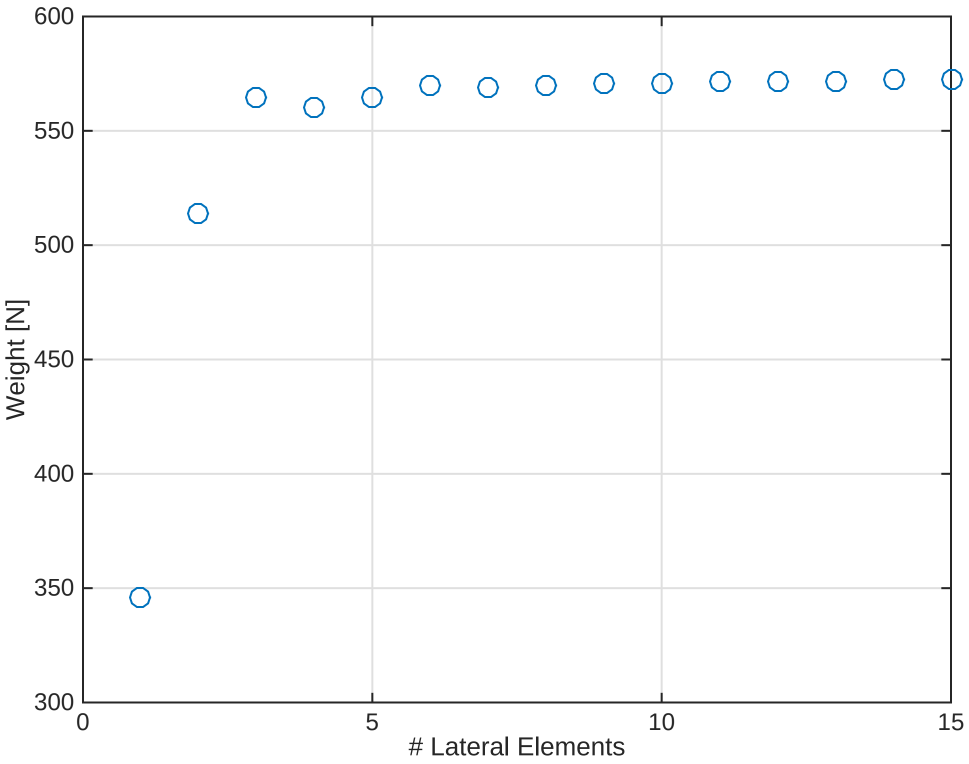


Figure 6.9: The steady-state value of the vertical contact force as a function of the number of lateral elements used to model the tire.

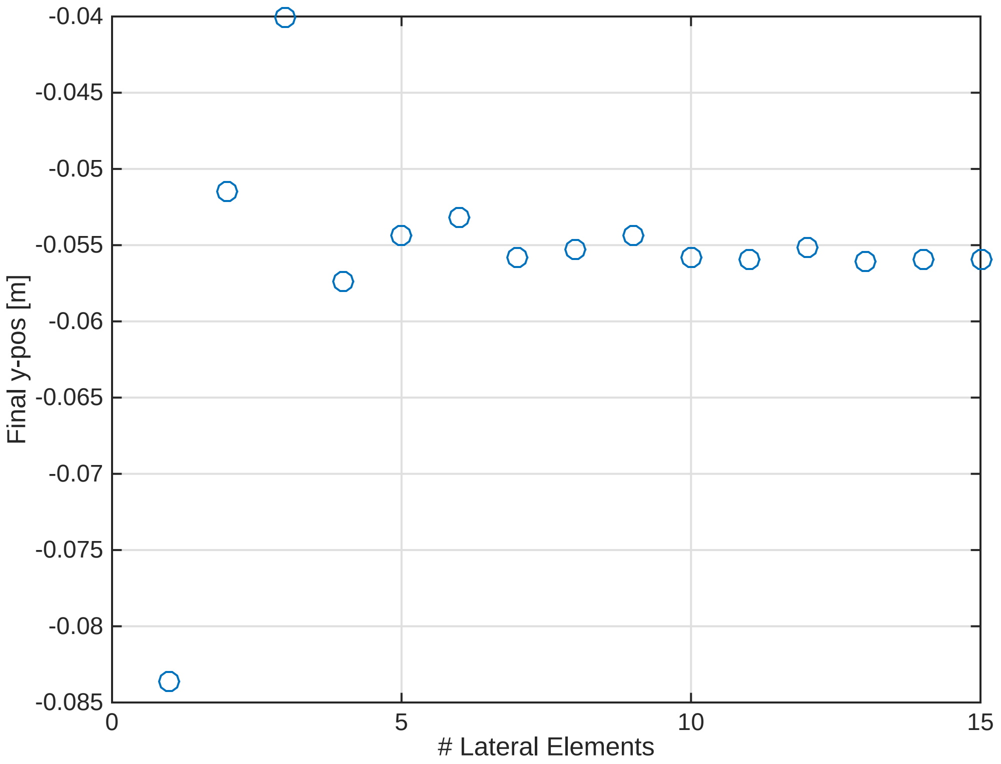


Figure 6.10: The steady-state value of the y-position of the tire hub as a function of the number of lateral elements used to model the tire.

6.2.2.2 Longitudinal roll test

Longitudinal roll tests were used to determine the number of radial elements that were required for convergence. A tire with a varying number of radial elements was rolled with 30% slip, 9 lateral elements, and a solver tolerance of $\tau = 1 \times 10^{-4}$. The drawbar pull coefficient, or ratio of normal force to tractive force, was plotted in Fig. 6.11. Although the normal contact force of the tire is not very sensitive to the number of radial elements, Fig. 6.11 indicates that at least 11 radial elements are required for convergence.

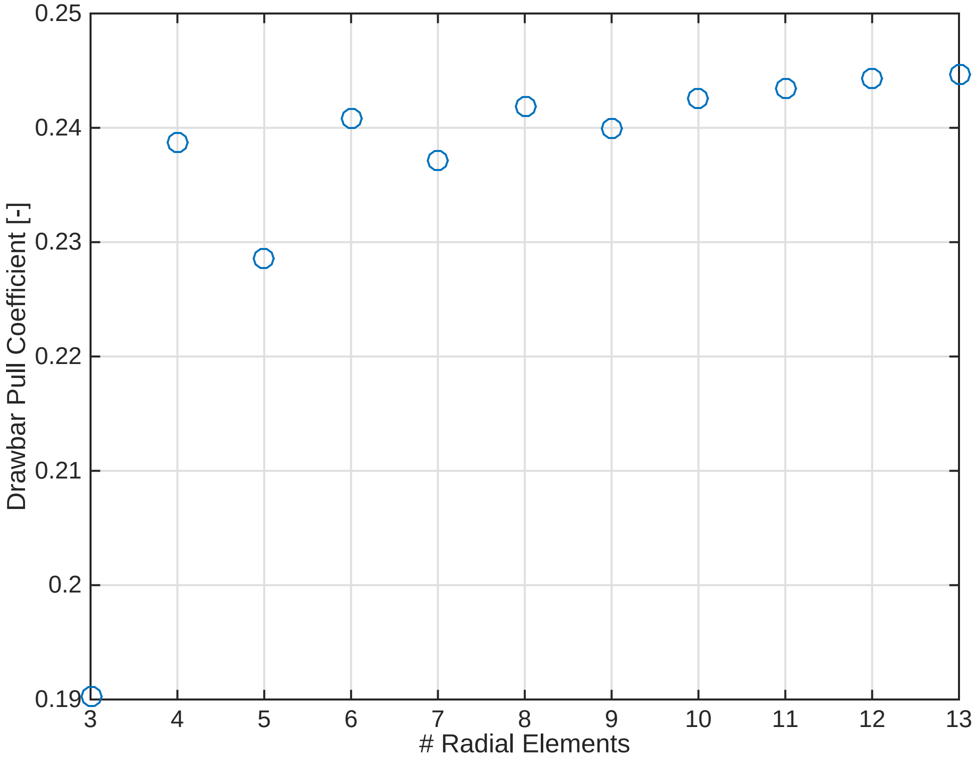


Figure 6.11: The steady-state value of the drawbar pull coefficient as a function of the number of radial elements used to model the tire.

Similar to the radial element tests, longitudinal roll tests were used to determine the solver tolerance that was required for convergence. A tire solved to a varying tolerance was rolled with 30% slip, 9 lateral elements, and 11 radial elements. The drawbar pull coefficient, or ratio of normal force to tractive force, and the average number of iterations required to solve a single step of steady-state dynamics were plotted in Fig. 6.12 and 6.13, respectively. Based on the results in Fig. 6.12 and 6.13, the drawbar pull coefficient converges after a tolerance of $\tau = 1 \times 10^{-3}$. In general, more iterations are required for a smaller (or “tighter”) tolerance.

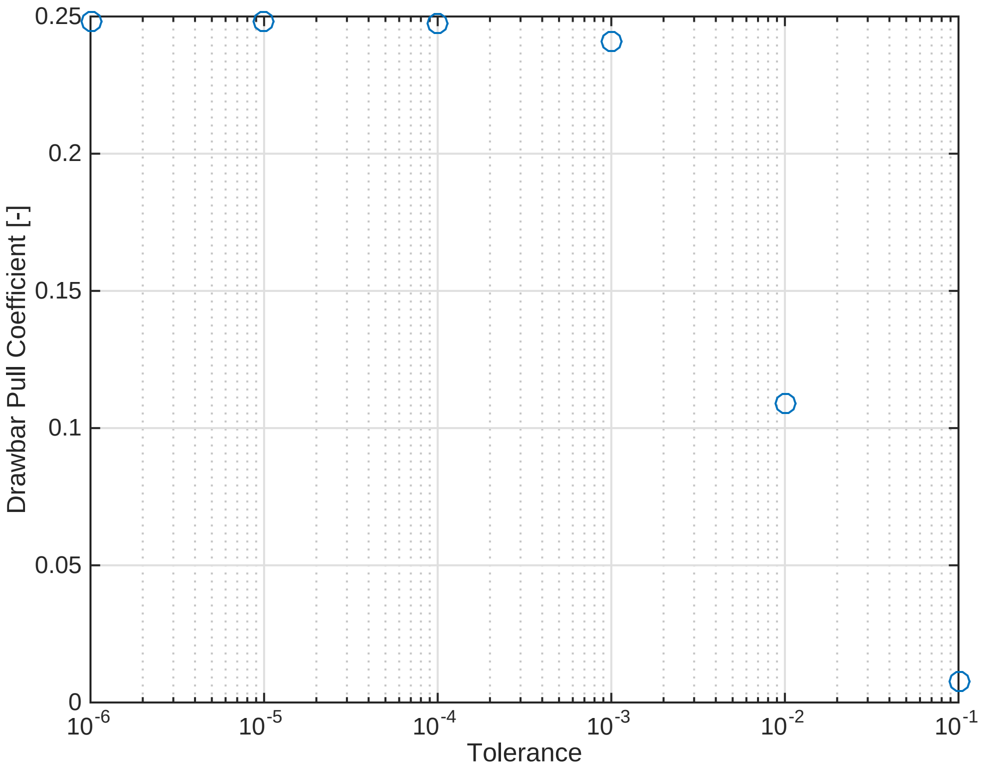


Figure 6.12: The steady-state value of the drawbar pull coefficient as a function of the solver tolerance.

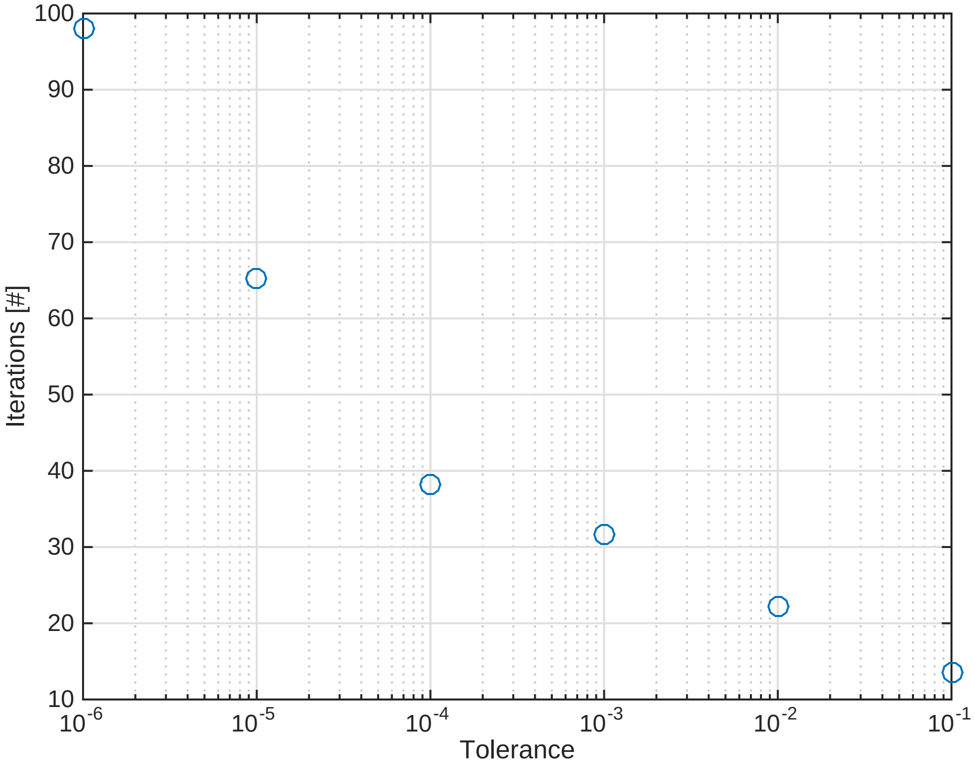


Figure 6.13: The number of APGD frictional contact iterations required to solve a single step of steady-state dynamics as a function of the solver tolerance.

Next, to determine the effect of the friction coefficient, a tire with a varying friction coefficient was rolled with 100% slip, 9 lateral elements, 11 radial elements, and a tolerance of $\tau = 1 \times 10^{-4}$. The drawbar pull coefficient, or ratio of normal force to tractive force, and the average number of iterations required to solve a single step of steady-state dynamics were plotted in Fig. 6.14 and 6.15, respectively. Based on the results in Fig. 6.14 and 6.15, both the drawbar pull coefficient and number of solver iterations increase proportionally with the friction coefficient.

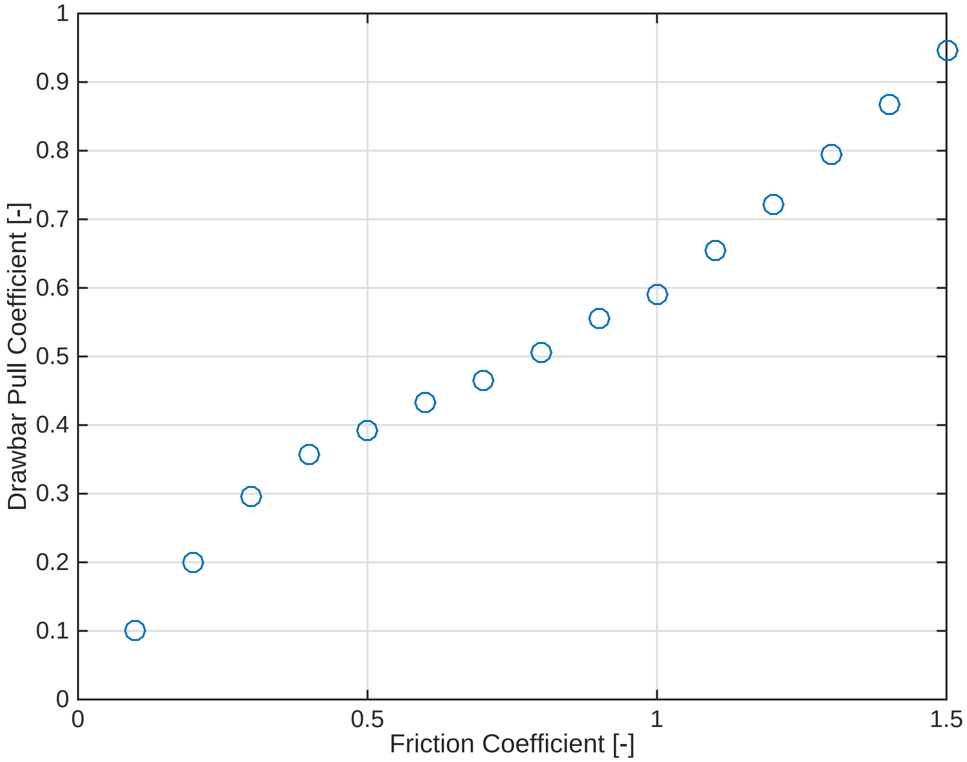


Figure 6.14: The steady-state value of the drawbar pull coefficient as a function of the tire friction coefficient.

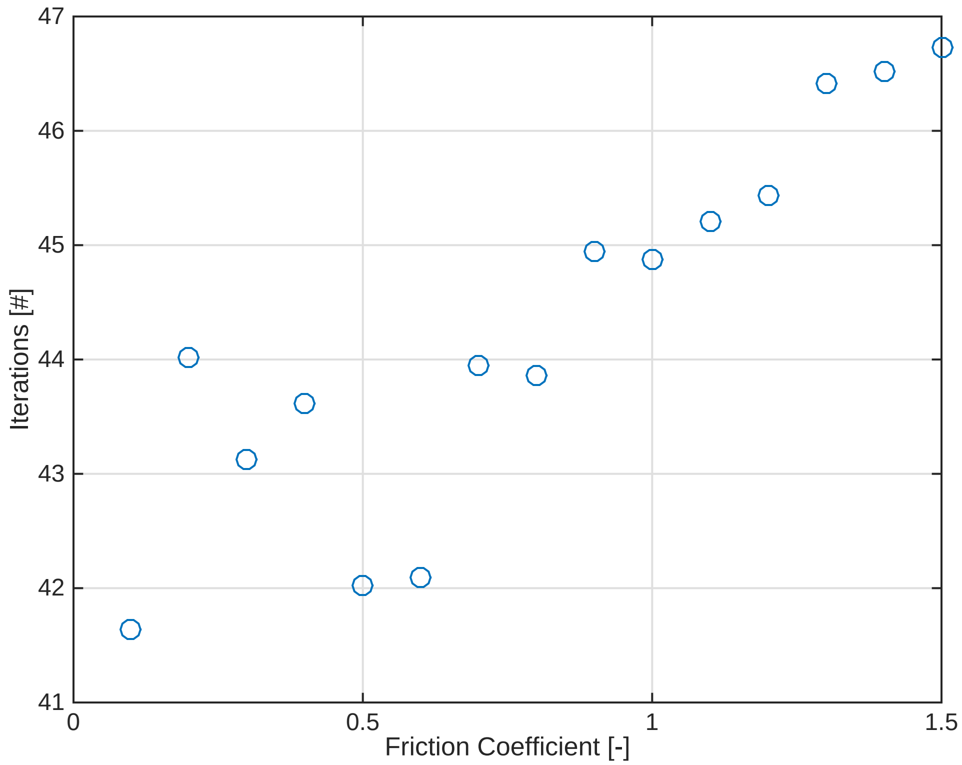


Figure 6.15: The average number of APGD frictional contact solver iterations per time step as a function of the tire friction coefficient.

Finally, to determine the effect of the longitudinal slip ratio, a tire with a varying slip was rolled with 9 lateral elements, 11 radial elements, and a tolerance of $\tau = 1 \times 10^{-4}$. The drawbar pull coefficient, or ratio of normal force to tractive force, and the torque were plotted in Fig. 6.16 and 6.17, respectively. Based on the results in Fig. 6.16 and 6.17, both the drawbar pull coefficient and torque increase with the longitudinal slip for low friction, with a transition from towed to driven (negative to positive) occurring at perfect rolling, or zero slip. As the friction coefficient increases, the drawbar pull as a function of slip becomes much more sinusoidal and accurately captures the experimental data for a HMMWV tire [**?**].

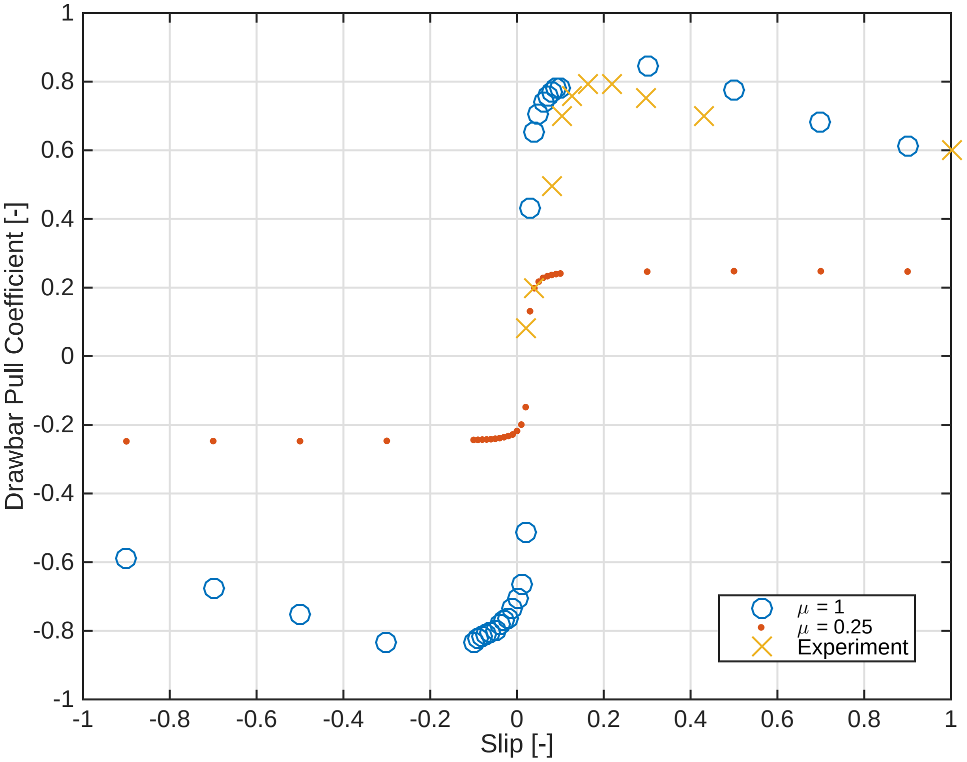


Figure 6.16: The steady-state value of the drawbar pull coefficient as a function of the longitudinal slip.

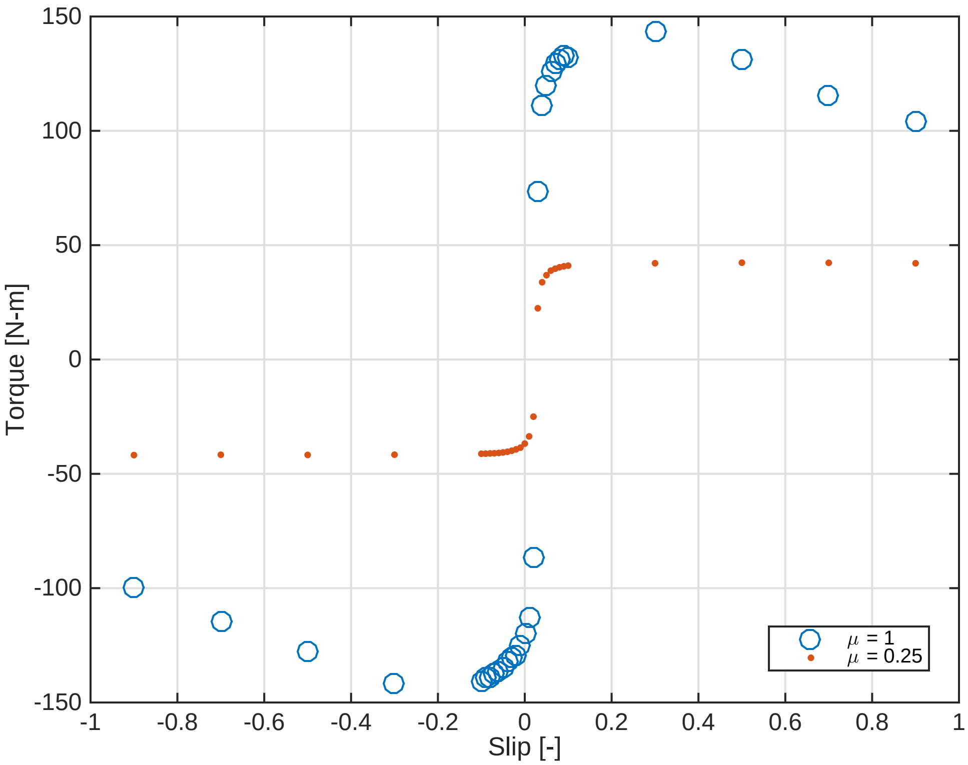


Figure 6.17: The steady-state value of the torque as a function of the longitudinal slip.

Chapter 7

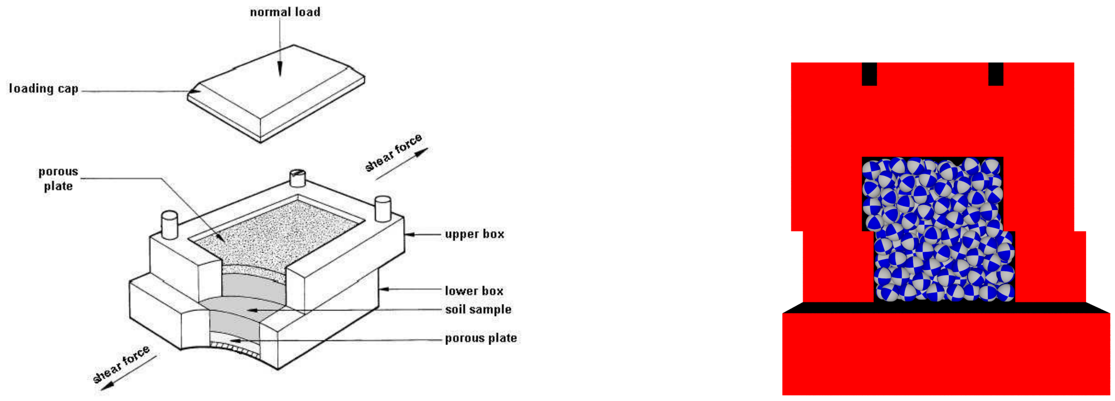
Experimental validation

This chapter reports on results obtained in the process of validating the solution methodology outlined in Chapter 3. To this end, the Chrono-generated results were compared against experimental data for three tests: direct shear, pressure-sinkage, and single wheel [**?**]. Quikrete, a commercially available concrete mix, was used in all of the experiments for this paper. Quikrete is poorly graded, having an average particle radius of approximately 0.4 mm and bulk density $\rho_b = 1.39$ g/cm^3 [**?**]. Based on this information, the DEM simulations used uniformly-sized spheres or ellipsoids with a major radius of 8 mm (approximately $20\times$ larger than the actual particle size) to reduce the number of bodies. Although the exact material density was not measured, the Quikrete particle density was estimated to be $\rho_g = 2.6$ g/cm^3 using the following equation

$$\rho_b = M_g/V_t = \rho_g V_g/V_t \tag{7.1}$$

based on the volume of the shear box enclosure (V_t), the total mass of the granular material (M_g), and the total volume of the granular material (V_g).

7.1 Direct shear test



(a) (b)

Figure 7.1: Schematic of the direct shear test (left), `Chrono` simulation of the direct shear test in the sheared/final configuration (right).

The setup for the direct shear test is shown in Fig. 7.1. The test is commonly used to measure the shear strength properties of a soil, specifically the cohesion, angle of friction, and shear modulus [**?**]. A soil sample is placed in a shear box aligned under a load cell, which applies a normal force to the soil. The top of the shear box is clamped while the lower half can be moved in a controlled fashion by a specified displacement. The horizontal force required to displace the soil is measured to produce a plot of the shear stress as a function of shear displacement. The shear box used in the experiments had an enclosure that contained granular material and was approximately $60 \times 60 \times 60$ mm in size. Two normal pressures: 16.9 and 71.4 kPa, were tested for loosely-packed, dry soil. Four tests were performed at each normal load to determine an experimental average and standard deviation. The 3D DEM simulations were conducted by generating an upper and lower shear box using four box collision primitives. Due to the larger particle size in DEM, a larger shear enclosure ($120 \times 120 \times 120$ mm) was used for the simulation. Approximately 561 bodies were randomly generated in the interior of the two shear boxes and allowed to settle under the normal load applied by an upper box collision geometry. After settling, the lower shear box was translated at a constant rate of 6.6×10^{-4} m/s for a distance of 3 mm requiring $T_{final} = 4.55$ s for the shearing phase. Shear stress was calculated by measuring the contact force on the upper shear box and dividing by the contacted shearing area. In general, the shear stress vs. displacement relationship levels off when the material stops expanding or contracting, and when interparticle bonds are broken. The theoretical state at which the shear stress remains constant while the shear displacement increases may be called the critical state, steady state, or residual shear stress. In the case of this paper, the residual shear stress is used to study how the behavior of the direct shear simulations vary as a function of the DEM parameters, such as the friction coefficient μ.

7.1.1 Tuning of solver parameters

Before calibrating the physical parameters of the model, a study was performed to select `Chrono` solution parameters that yielded sufficiently accurate results. Indeed, if too large of an integration step size h is used or an excessively lax convergence stopping parameter τ is adopted, the numerical results might not be "converged", thus compromising the predictive character of the simulation. To determine the appropriate h and τ, several analyses were run using granular material composed of uniformly-sized spheres with a radius of 8 mm, a density of 2.6 g/cm^3, and a sliding friction coefficient of $\mu = 0.5$. The normal load applied was $\sigma = 16.9$ kPa. To determine the necessary step size h, a tight solver tolerance, $\tau = 1 \times 10^{-2}$ N, was used while running analyses at various h values. The results of this exercise are summarized in Fig. 7.2. Conversely, given that a step size $h = 0.001$ s appeared sufficiently small, a parametric sweep was carried out over τ to gauge the sensitivity of the analysis results with respect to the solver tolerance. The outcomes of this sweep are summarized in Fig. 7.3. The results obtained elicit the following conclusions: the residual shear stress increases as the solver tolerance is tightened; the shape of the numerical shear stress profiles comes in line with expectations; and the results obtained are "converged" when $h = 0.001$ s and $\tau = 5 \times 10^{-2}$ N.

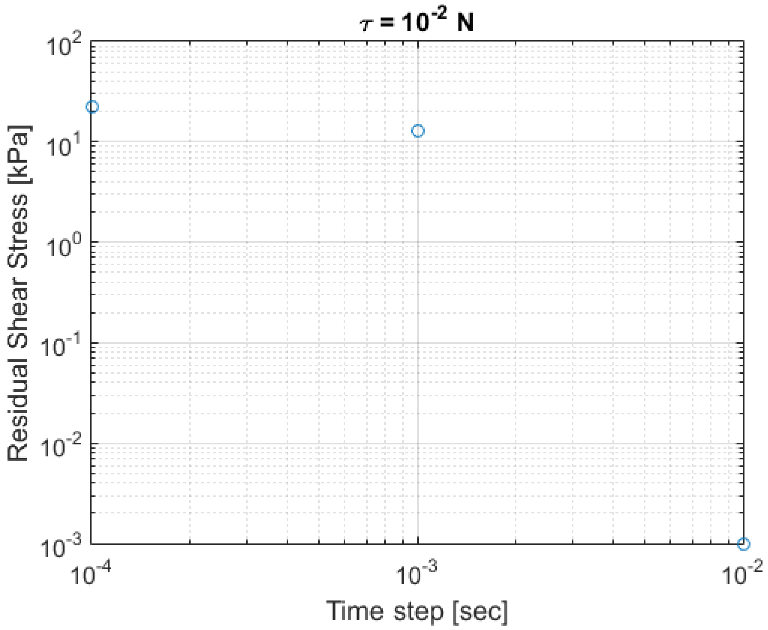


Figure 7.2: Residual shear stress for the direct shear test with a varying solver time step and a fixed tolerance of $\tau = 1 \times 10^{-2}$ N.

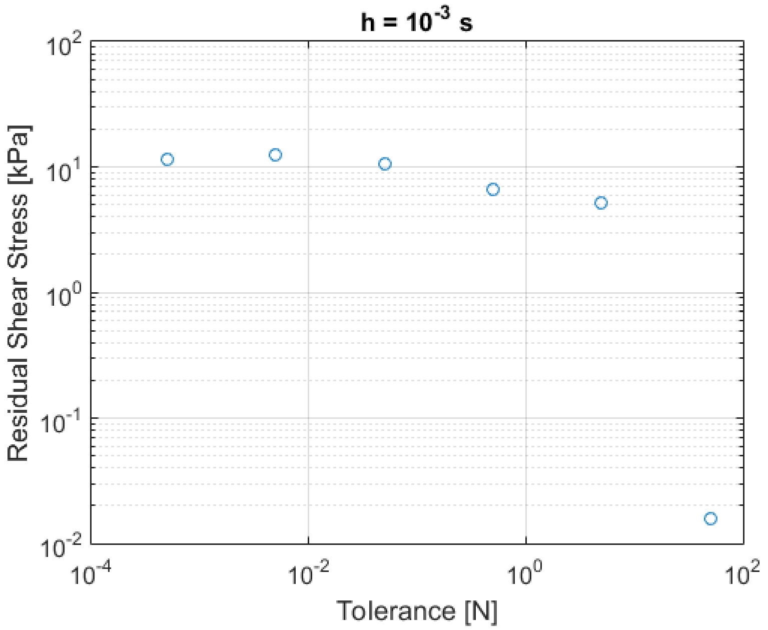


Figure 7.3: The residual shear stress for the direct shear test with a varying solver tolerance and a fixed time step of $h = 0.001$ s.

Since granular dynamics analysis is compute intensive, one would like to advance the numerical solution at a large step size and take a relatively small number of iterations to resolve each time step. Based on the results above, the step size and convergence tolerance required for good results was $h = 0.001$ s and $\tau = 5 \times 10^{-2}$ N, respectively.

7.1.2 Calibration of model parameters

After gaining confidence in the solver parameters, a model parameter calibration aimed at determining the friction coefficient and understanding whether particle shape can improve the quality of the results can be performed. The results for the friction coefficient study are reported in Fig. 7.4. The normal load applied was $\sigma = 16.9$ kPa, the granular material used in the experiment had density 2.6 g/cm^3 and the shape was approximately spherical with radius 8 mm. With the exception of $\mu = 0.9$, the numerical solution obtained suggests that the shear stress profile raises as the sliding friction coefficient increases. Qualitatively, the shape of the numerical shear stress curves matches well to experimental data, with a sliding friction coefficient $\mu = 0.5$ giving the closest agreement.

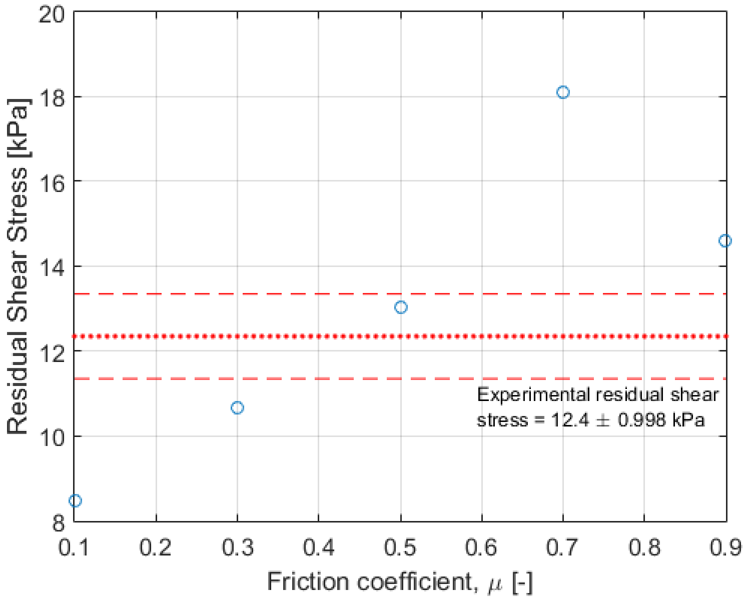


Figure 7.4: Residual shear stress for the direct shear test with a varying friction coefficient and a constant shape ratio $s_r = 1$. The dotted and dashed lines correspond to the experimental residual shear stress average and standard deviation, respectively.

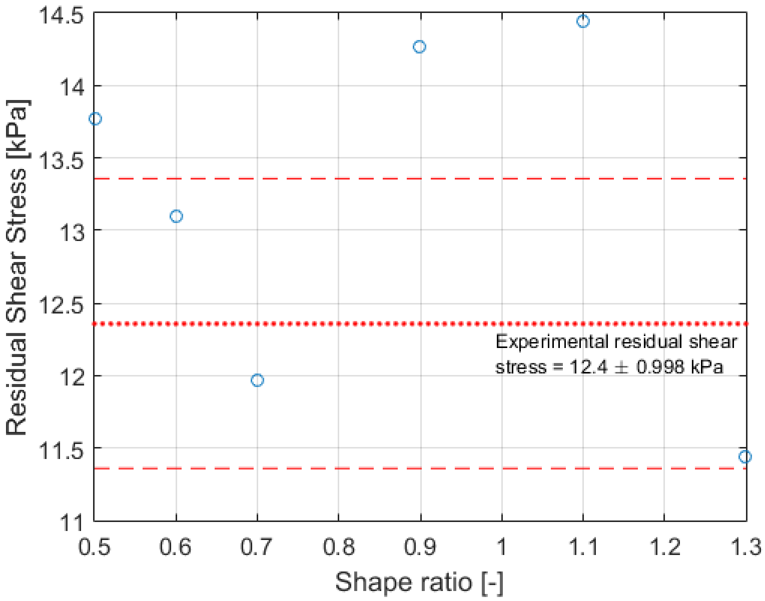


Figure 7.5: Residual shear stress for the direct shear test with a varying shape ratio and a constant friction coefficient $\mu = 0.5$. The dotted and dashed lines correspond to the experimental residual shear stress average and standard deviation, respectively.

The second model parameter tuning study was performed to quantify the effect of particle shape on the direct shear test numerical results. The granular material was made up of uniformly-sized ellipsoids with a major radius of 8 mm and a varying shape ratio from 0.5 - 1.3. This ratio was defined as

$$r = \begin{cases} r_x = r,\ r_y = s_r \times r,\ r_z = r, & \text{if } s_r < 1.0 \\ r_x = r/s_r,\ r_y = r,\ r_z = r/s_r, & \text{otherwise} \end{cases} \tag{7.2}$$

This numerical experiment, which was carried out with $\mu = 0.5$, $h = 0.001$ s, $\tau = 5 \times 10^{-2}$ N, and $\sigma = 16.9$ kPa, led to the results in Fig. 7.5. Qualitatively, the shear stress curves are not very sensitive to the shape ratio. The shape of the simulated direct shear profiles matches quite well to the experimental data.

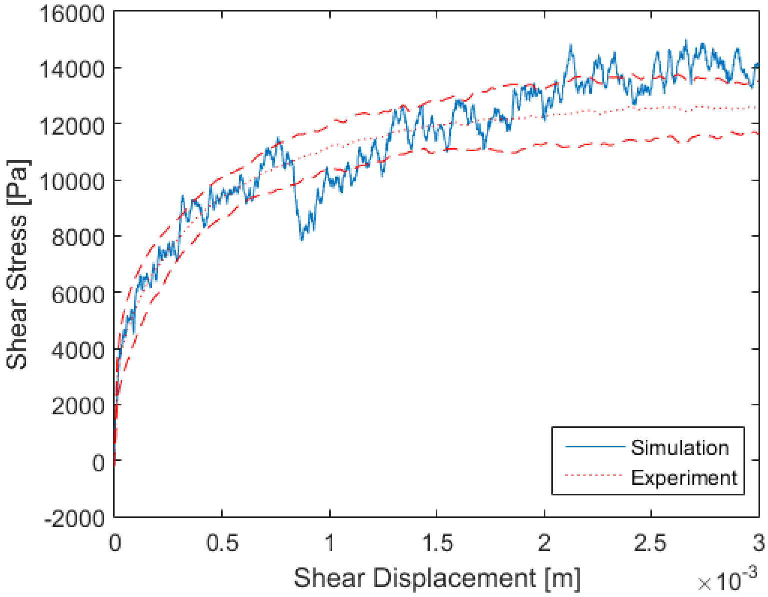


Figure 7.6: Shear stress vs. displacement for the direct shear test with $\sigma = 16.9$ kPa.

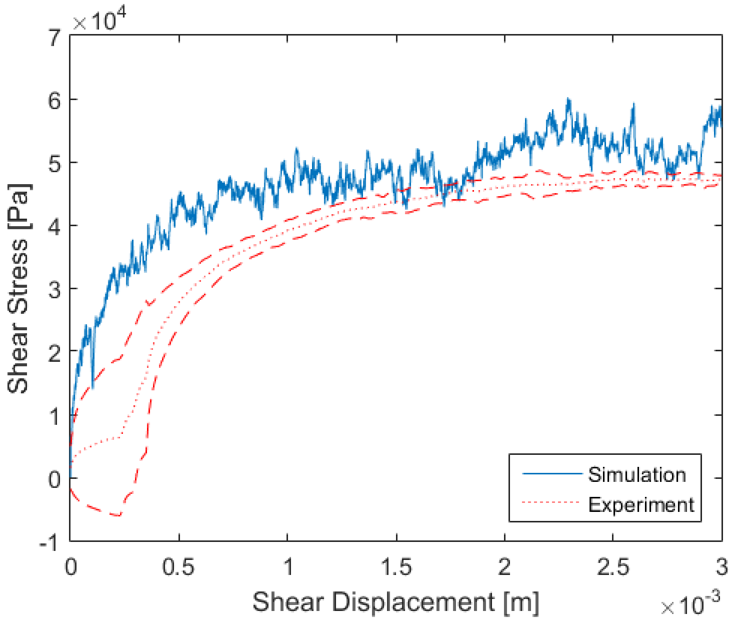


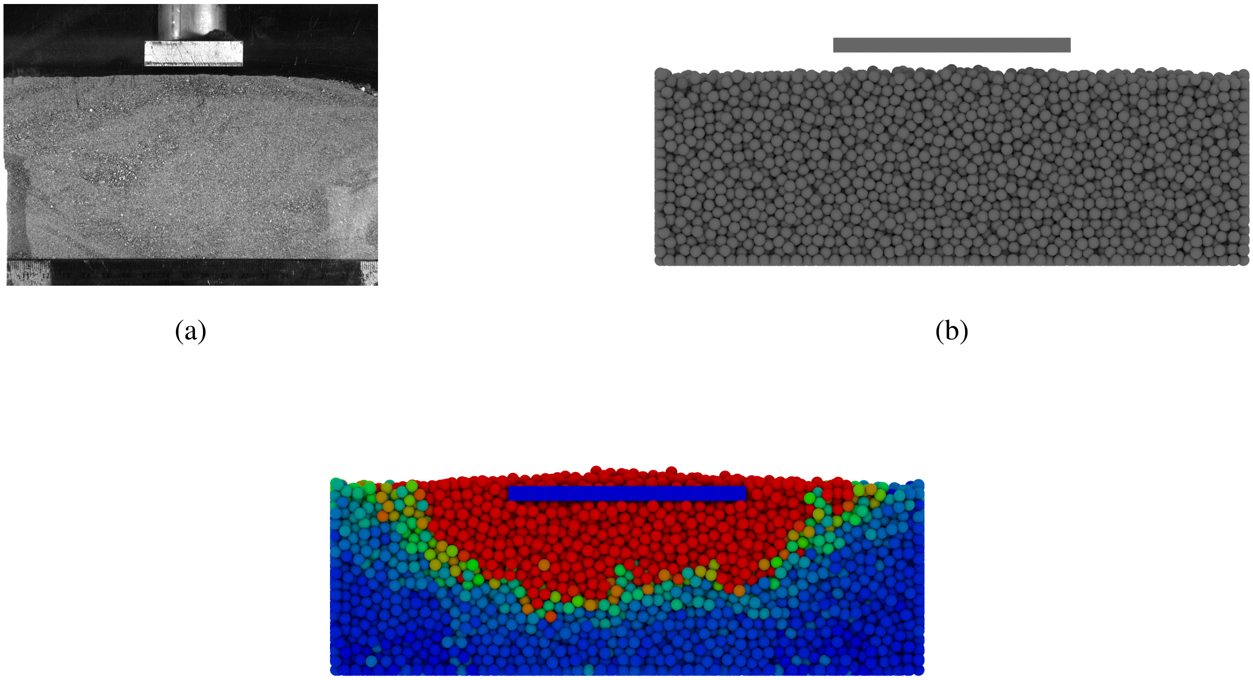
Figure 7.7: Shear stress vs. displacement for the direct shear test with $\sigma = 71.4$ kPa.

7.1.3 Predictive attribute assessment

For a normal load $\sigma = 16.9$ kPa, the parameter selection process carried out in sections §7.1.1 and §7.1.2 suggests that the numerical results come close to experimental data when $\mu = 0.5$, $h = 0.001$ s, $\tau = 5 \times 10^{-2}$ N, and $s_r = 0.6$. The predictive attribute of `Chrono` is assessed next by keeping these solver and model parameters constant and modifying the experimental setup; i.e., the normal loading σ. Should `Chrono` be predictive, the numerical results for $\sigma = 71.4$ kPa would continue to be close to corresponding experimental data measured for this loading scenario. Indeed, the plots in Fig. 7.6 and 7.7 confirm that the numerical results and experimental data are similar for both loading scenarios. The direct shear simulations were run on an AMD Opteron 6274 2.2 GHz processor using eight cores. For the normal load $\sigma = 16.9$ kPa, the simulation required 23.2 s of compute time per step ($23,200\times$ slower than real time) and used an average of $4,537$ APGD iterations to resolve an average of $2,112$ collisions per step. For the normal load $\sigma = 71.4$ kPa, the simulation required 80.6 s of compute time per step ($80,600\times$ slower than real time) and used an average of $8,925$ APGD iterations to resolve an average of $2,129$ collisions per step.

7.2 Pressure-sinkage test

The pressure-sinkage test, shown in Fig. 7.8, is used to measure the load bearing properties of a soil. To this end, a plate is pushed with a constant downward velocity to penetrate a soil sample. Designed to simulate loading rates similar to the ones exerted by a wheel during motion, the test uses a load cell to record the force induced by the plate. The soil bin used in the experiments was $150 \times 400 \times 160$ mm (W $\times$ L $\times$ H) in size. Two different plates were used to penetrate into the loosely-packed, dry soil. Both of the plates had a length of 160 mm and a height of 10 mm. Each plate had a different width; i.e., 30 and 50 mm. The plate penetrated the soil at 10 mm/s. Fifteen experimental tests were performed at each plate size. The `Chrono` tests were conducted by generating a soil bin in which $5,377$ bodies were randomly generated, dropped, and allowed to settle under gravity. After settling, a plate was moved down at a constant rate of 10 mm/s for a distance of 30 mm requiring $T_{final} = 3$ s for the pressing phase. The pressure due to sinkage was calculated by measuring the contact force on the plate and dividing by the length and width of the plate geometry.



(a)

(b)

(c)

Figure 7.8: A photograph of the pressure-sinkage test experiment in the initial configuration (a), `Chrono` simulation of the pressure-sinkage test in the filled/initial configuration (b), `Chrono` simulation of the pressure-sinkage test in the pressed/final configuration (c). The colors represent the relative magnitude of the linear velocity of the bodies (red = fast, blue = slow).

7.2.1 Tuning of solver parameters

Before calibrating the model parameters, a study was performed to determine the necessary solver step size and termination criteria for the pressure-sinkage test. The study was performed with granular material composed of uniformly-sized spheres with a radius of 8 mm, a density of 2.6 g/cm^3, and $\mu = 0.5$. To determine the necessary time step, the solver tolerance was set to

$\tau = 5 \times 10^{-2}$ N and the step size h was varied to determine its effect on the numerical results. The outcomes of this study, carried out for a 50 mm plate width, are reported in Fig. 7.9. The numerical integration step size $h = 0.001$ s, deemed appropriate for the shear test, yields good results for the pressure sinkage test as well. Finally, using a step size $h = 0.001$ s, a parametric sweep was carried out over τ to gauge the sensitivity of the analysis results with respect to the solver tolerance. The outcomes of this sweep are summarized in Fig. 7.10. Similar to the analysis in §7.1.1, these results suggest that a solver tolerance $\tau = 5 \times 10^{-2}$ N and a solver time step $h = 0.001$ s is sufficient to characterize the physics of interest in the pressure-sinkage test.

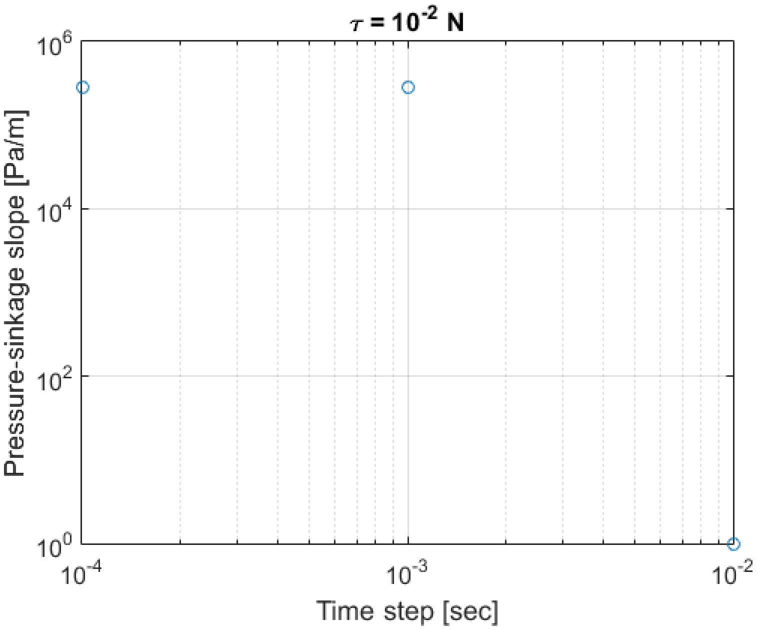


Figure 7.9: Slope of the pressure vs. sinkage curve for various solver step sizes h.

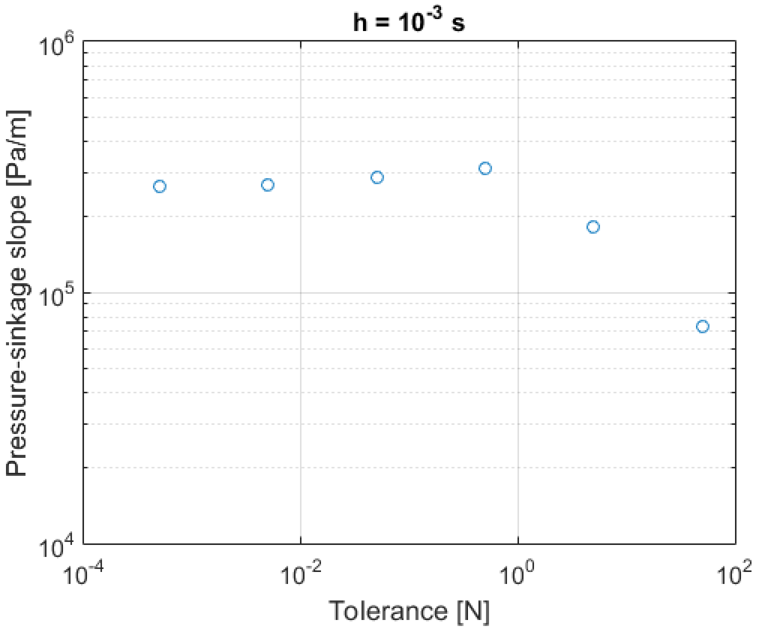


Figure 7.10: Slope of the pressure vs. sinkage curve for various solver tolerances τ.

7.2.2 Calibration of model parameters

The interest here is gauging the influence of the friction coefficient μ and of the particle shape, controlled through the coefficient s_r, on the numerical results of the pressure-sinkage test. The calibration tests were performed with granular material composed of uniformly-sized spheres with a radius of 8 mm, a density of 2.6 g/cm^3, and a varying sliding friction coefficient. In accordance with the conclusions reached in Section 7.2.1, this calibration was carried out using a simulation time step $h = 0.001$ s and a solver tolerance $\tau = 5 \times 10^{-2}$ N. The plate width was 50 mm.

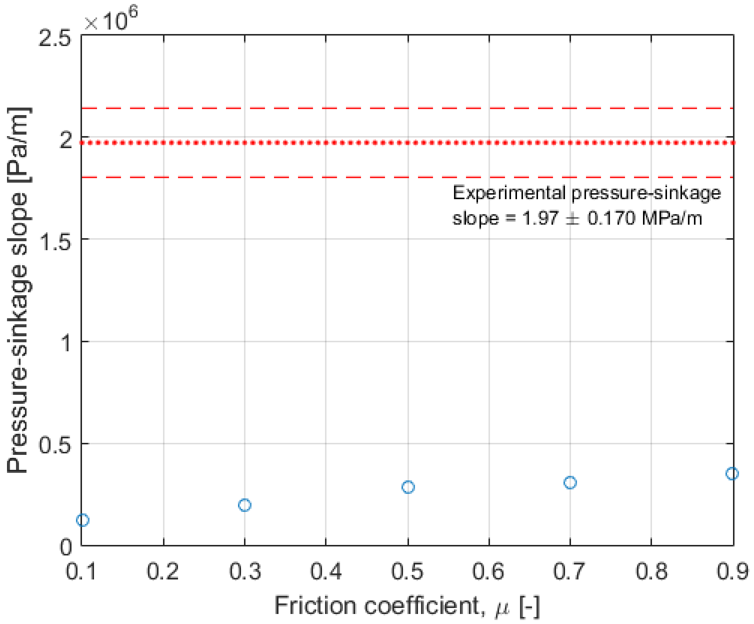


Figure 7.11: Slope of the pressure vs. sinkage curve for the pressure-sinkage test with a varying friction coefficient and a constant shape ratio $s_r = 1$. The dotted and dashed lines correspond to the experimental residual shear stress average and standard deviation, respectively.

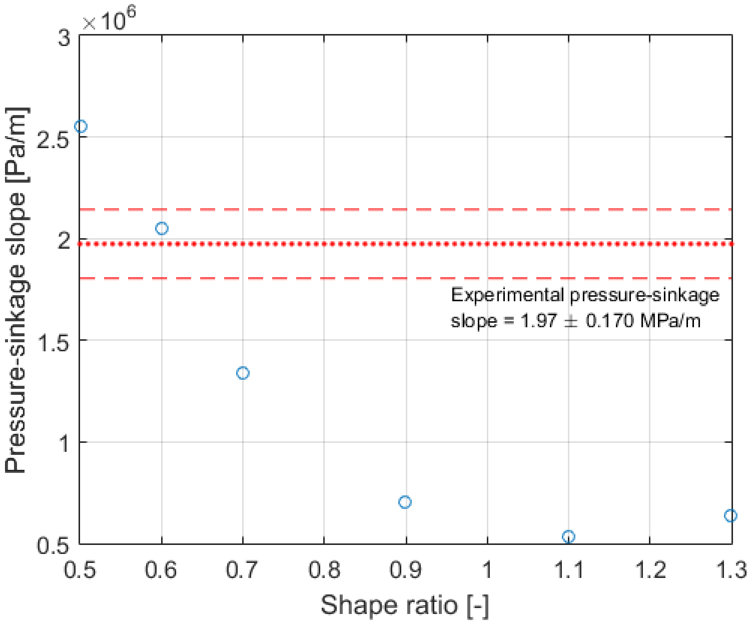


Figure 7.12: Slope of the pressure vs. sinkage curve for a set of shape ratios s_r and constant friction coefficient $\mu = 0.5$. The dotted and dashed lines correspond to the experimental residual shear stress average and standard deviation, respectively.

The results in Fig. 7.11 suggest that the slope of the pressure-sinkage curve obtained by `Chrono` is smaller than the experimental one by roughly a factor of five. In fact, varying the friction coefficient has vary little effect on the slope of the pressure-sinkage curve. We posit that this discrepancy is due to the element dimension and/or the actual shape of the quikrete material grains. By changing the shape of the `Chrono` elements, for $s_r = 0.6$ the experimental and simulation results match very well. For $\mu = 0.5$, the sensitivity of the results with respect to the shape factor s_r is summarized in Fig. 7.12, for s_r between 0.5 and 1.3.

7.2.3 Predictive attribute assessment

The next set of numerical experiments used the solver and model parameters identified in sections §7.2.1 and §7.2.2 for a plate of width 50 mm. Incidentally, these parameters assumed values identical to the ones in the shear-stress test. Specifically, the granular material was composed of uniformly-sized spheres with a radius of 8 mm, $s_r = 0.6$, $\mu = 0.5$, $\tau = 5 \times 10^{-2}$ N, and $h = 0.001$

s. Maintaining these parameters, the predictive attribute of Chrono was assessed by comparing the numerical results to experimental data when the width of the plate changed from 30 to 50 mm. The results obtained, summarized in Fig. 7.13 and 7.14, confirm that, as expected, the pressure-sinkage profiles migrate to higher values as the plate width increases. Moreover, the predicted values are close to the experimental results in Fig. 7.14. The pressure-sinkage simulations were run on an AMD Opteron 6274 2.2 GHz processor using eight cores. For a plate of width 50 mm, the simulation required 27.9 s of compute time per simulation step ($27,900\times$ slower than real time) and used an average of 179 APGD iterations to resolve an average of $18,973$ collisions per step. For a plate of width 50 mm, the simulation required 30.1 s of compute time per simulation step ($30,100\times$ slower than real time) and used an average of 199 APGD iterations to resolve an average of $19,058$ collisions per step.

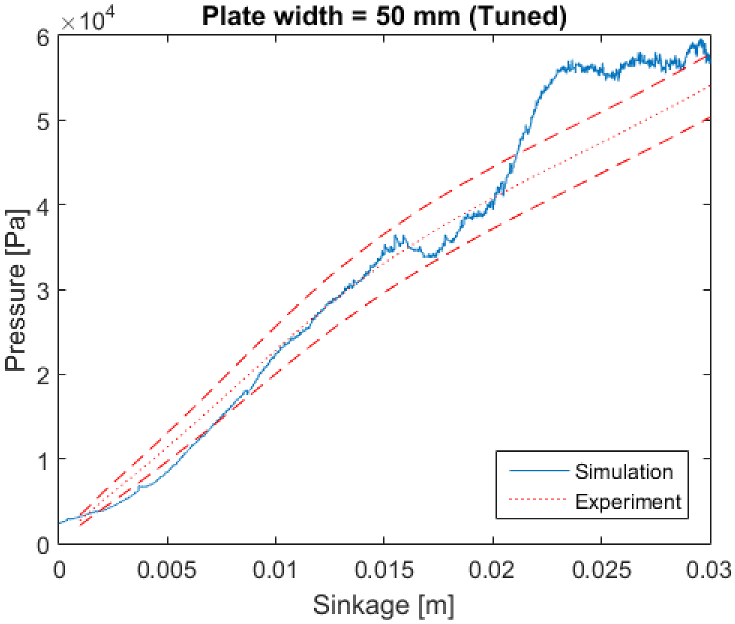


Figure 7.13: A comparison of the experimental and tuned pressure-sinkage profiles with a width of 50 mm.

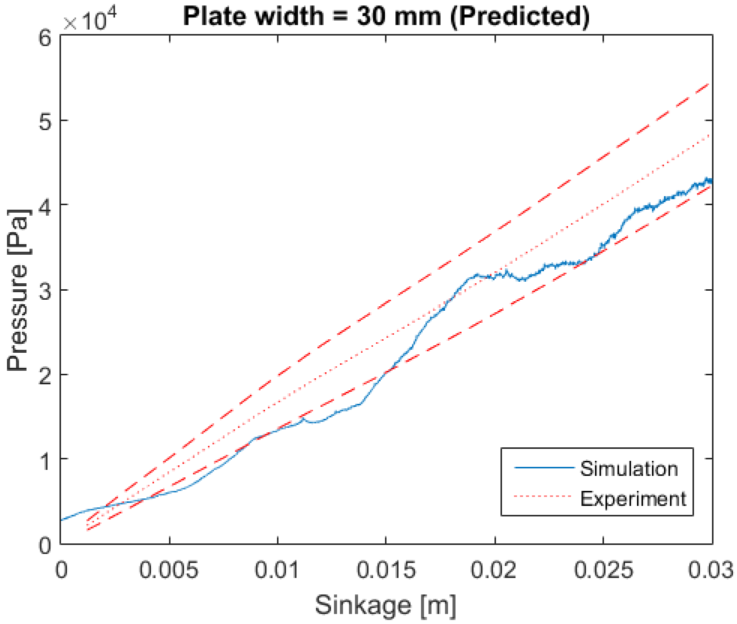


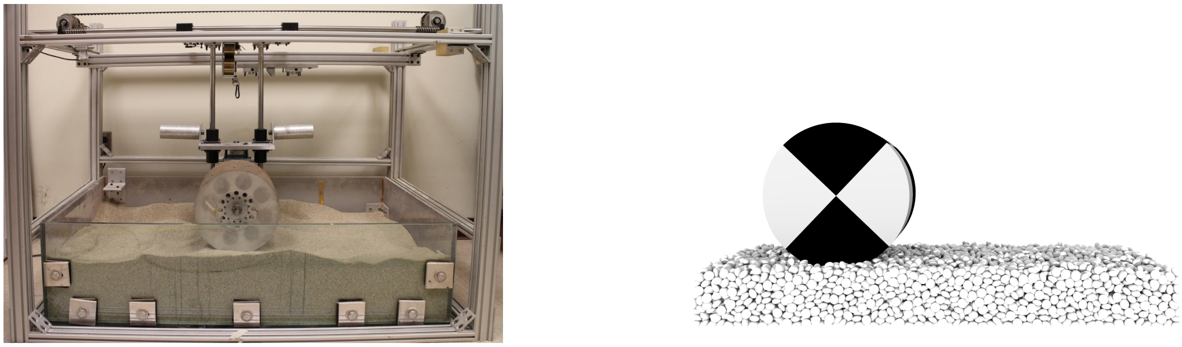
Figure 7.14: A comparison of the experimental and tuned pressure-sinkage profiles with a width of 30 mm.

7.3 Single wheel test

A single wheel test is used to investigate a wheel's motion under controlled slip and normal loading conditions within a confined soil bin of dimensions $320 \times 800 \times 150$ mm (W × L × H). The drawbar pull, wheel torque, and sinkage were measured for a lug-less rigid wheel for several slip cases and loading scenarios. The wheel used in this study had a width of 160 mm and a radius $r_w = 130$ mm. To produce a desired constant slip, the wheel was rolled on the soil with a constant angular velocity of $\omega = 0.3$ rad/s and a certain fixed translational velocity v based on the slip defined as

$$v = (1.0 - slip)\ \omega\ r_w\ . \tag{7.3}$$

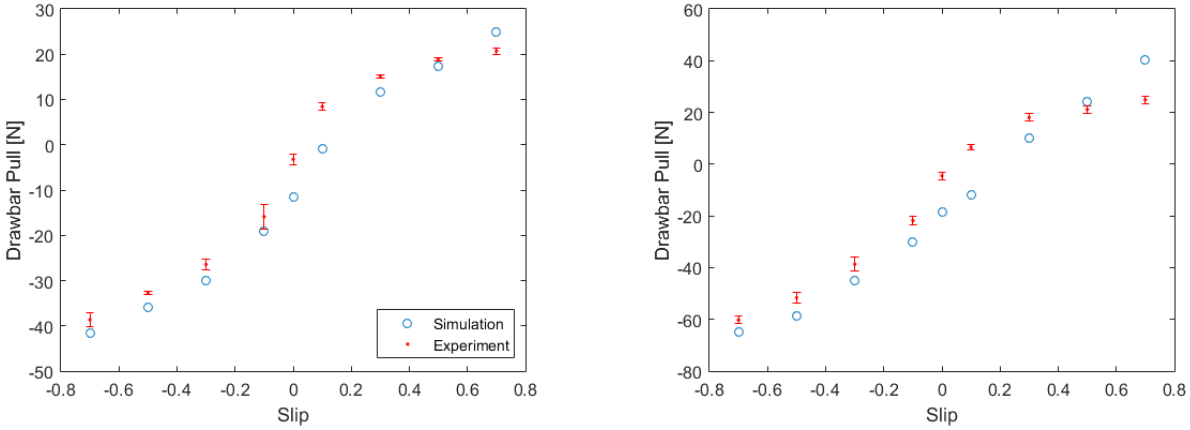
The numerical tests in Chrono were conducted by simulating a soil bin in which $10,790$ bodies were randomly generated and allowed to settle under gravity. After settling, the wheel was rolled at the desired slip ratio for $T_{final} = 8$ s.



(a) (b)

Figure 7.15: A photograph of the single wheel test experiment at MIT's Robotic Mobility Group [**?**] (left), `Chrono` simulation of the single wheel test (right).

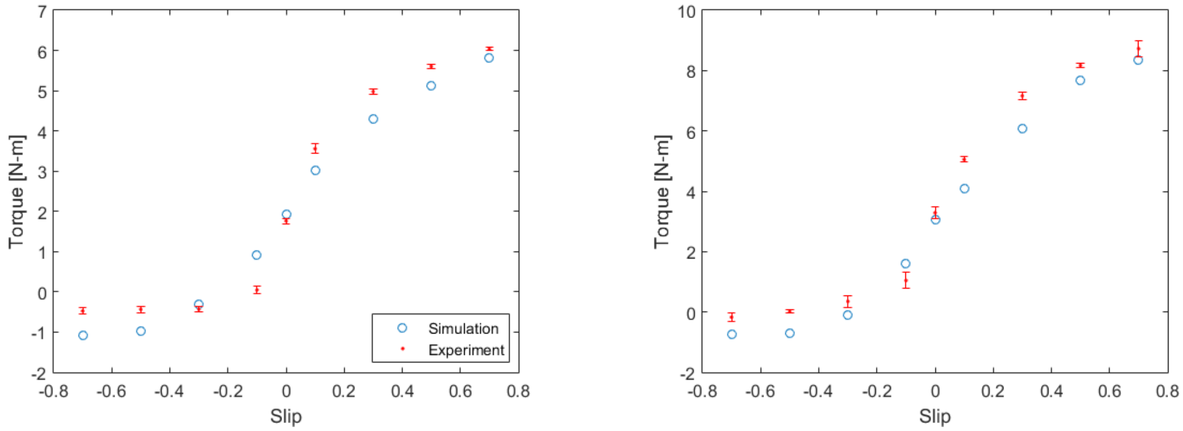
Using the set of solver and model parameters selected in sections §7.1 and §7.2, a study was performed to gauge whether `Chrono` can reproduce the experimental results at varying values of wheel slips. These "predictive attribute" verification tests were performed with granular material composed of uniformly-sized ellipsoids with a major radius of 8 mm, a density of 2.6 g/cm^3, $s_r = 0.6$, $\mu = 0.5$, $h = 0.001$ s, $\tau = 5 \times 10^{-2}$ N. The quantitative results of this study are summarized in Figs. 7.16 through 7.18 for the drawbar pull, torque, and sinkage, respectively. It can be seen that as the slip of the wheel increases, the drawbar pull, torque, and sinkage also increase and the values obtained depend on the normal loads applied to the wheel; i.e., 80 and 130 N. Moreover, the numerical values are close to the experimental results. The single wheel simulations were run on an AMD Opteron 6274 2.2 GHz processor using eight cores. For a normal load of 80 N, the simulation required 279 s of compute time per simulation step ($279,000\times$ slower than real time) and used an average of 473 APGD iterations to resolve an average of $39,090$ collisions per step. For a normal load of 130 N, the simulation required 314 s of compute time per simulation step ($314,000\times$ slower than real time) and used an average of 513 APGD iterations to resolve an average of $39,379$ collisions per step.



(a)

(b)

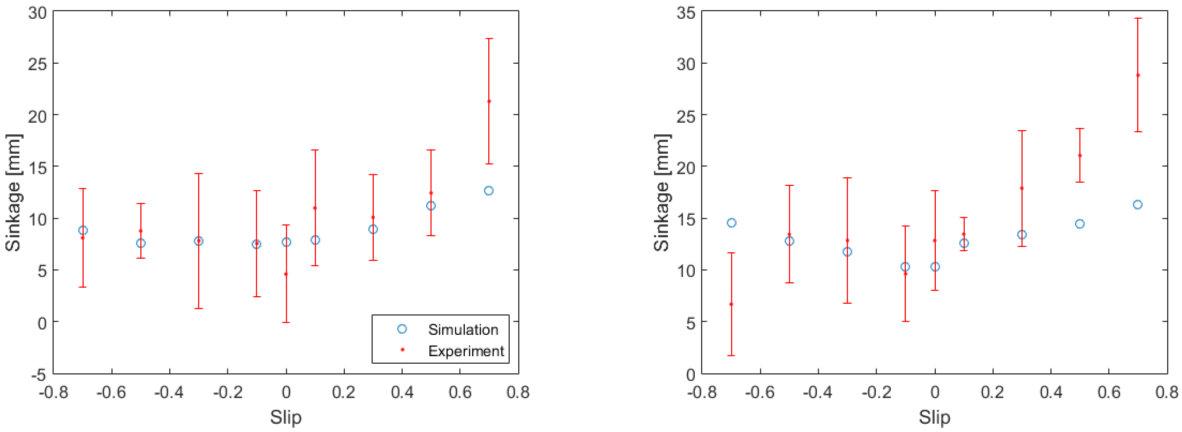
Figure 7.16: Drawbar pull vs. slip curves for a normal load 80 N (left), and 130 N (right).



(a)

(b)

Figure 7.17: Torque vs. slip curves for a normal load 80 N (left), and 130 N (right).



(a) (b)

Figure 7.18: Sinkage vs. slip curves for a normal load 80 N (left), and 130 N (right).

In addition to investigating the overall wheel performance, the direction of the velocities of the individual soil elements were plotted as blue lines in Figs. 7.19 - 7.21. In this study, the wheel has a 13 cm radius and is 16 cm wide. The wheel has a weight of 80 N and each discrete element is an ellipsoid $(4 \times 1.6 \times 4$ mm) with a density of $2,600$ kg/m^3 and a friction coefficient $\mu = 0.5$. There are approximately $200,000$ discrete elements in each simulation. The center of mass of the wheels are at the same position in each figure, although the rotation is different due to different longitudinal slips (30%, 0, -30%).

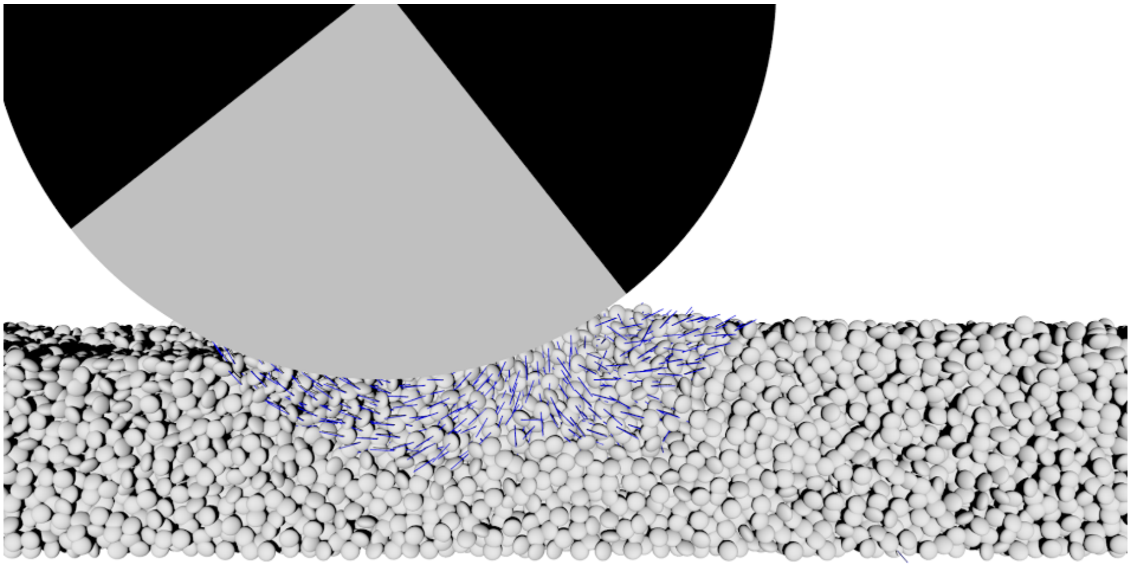


Figure 7.19: Cross-section of a `Chrono` simulation of the single wheel test with 30% slip.

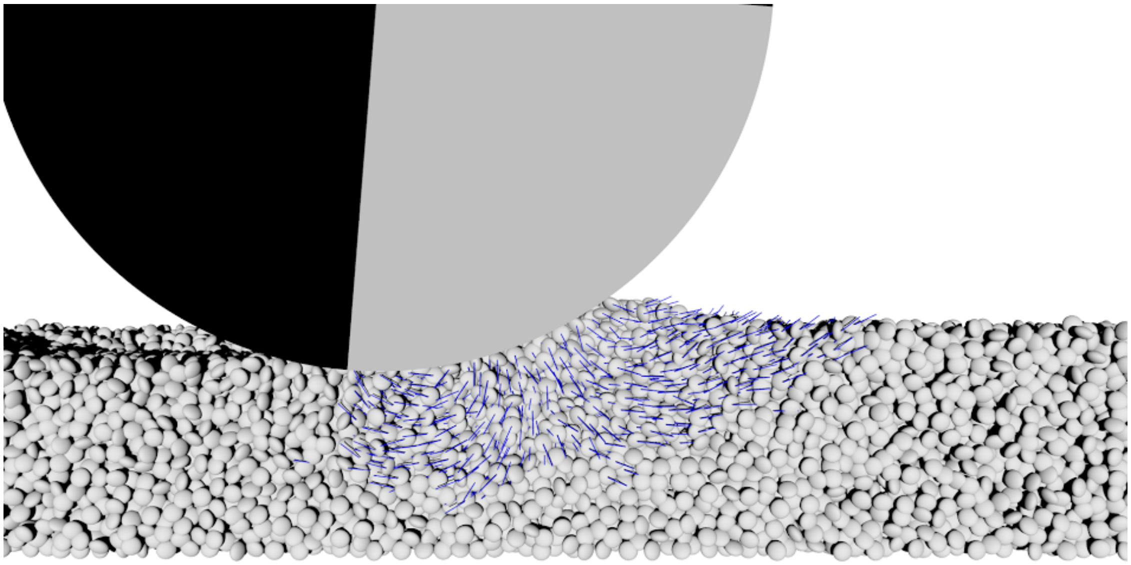


Figure 7.20: Cross-section of a `Chrono` simulation of the single wheel test with no (zero) slip.

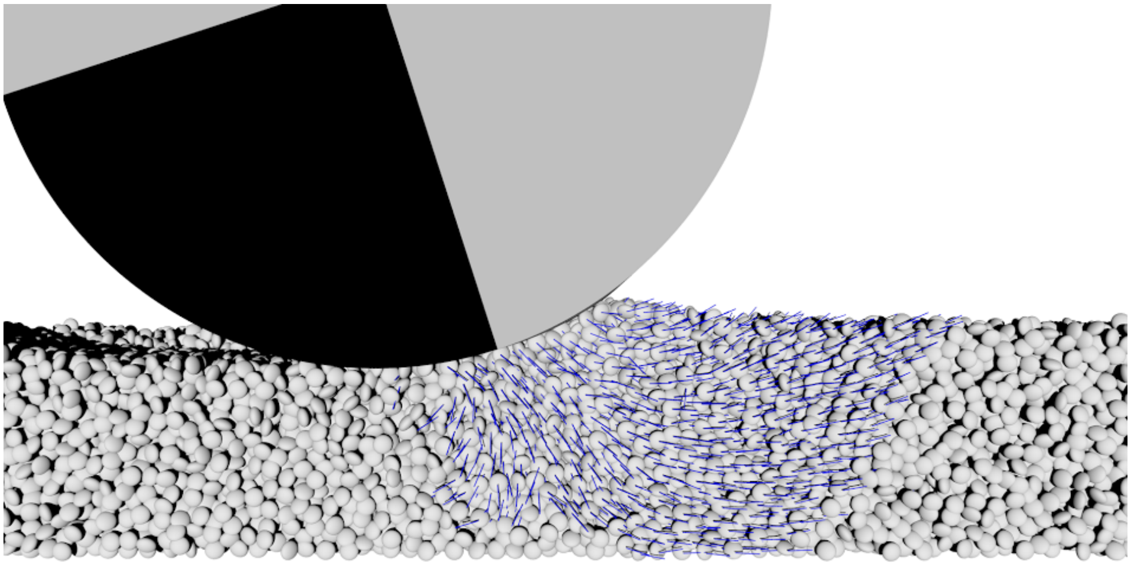


Figure 7.21: Cross-section of a Chrono simulation of the single wheel test with -30% slip.

Based on these figures, it is qualitatively apparent that as the slip goes from positive to negative, the rear velocity ”bulb“ gets smaller and the front bulb gets larger. In fact, the area of the velocity bulbs appear to be approximately equal in the perfect rolling case (zero slip). Additionally, negative slips affect particles that are deeper down. Lastly, the point where the bulbs converge does not change for different slips.

Chapter 8

Demonstration of technology

At the present time, `Chrono::Vehicle` can be used to model and simulate a significant variety of wheeled ground vehicles with various topologies [**?**]. Modeling of vehicle systems is done in a modular fashion, with a vehicle defined as an assembly of instances of various subsystems (suspension, steering, driveline, etc.). Flexibility in modeling is provided by adopting a template-based design. In `Chrono::Vehicle` templates are parameterized models that define a particular implementation of a vehicle subsystem. As such, a template defines the basic modeling elements (bodies, joints, force elements), imposes the subsystem topology, prescribes the design parameters, and implements the common functionality for a given type of subsystem (e.g. suspension) particularized to a specific template (e.g. double wishbone).

Currently, the following vehicle subsystems and associated templates have been implemented:

suspension: double wishbone, reduced double wishbone (with the A-arms modeled as distance constraints), multi-link, solid-axle;

steering: Pitman arm, rack-and-pinion;

driveline: 2WD shaft-based, 4WD shaft-based; these templates are based on specialized `Chrono` modeling elements, named `ChShaft`, with a single rotational degree of freedom and various shaft coupling elements (gears, differentials, etc.);

wheel: in `Chrono::Vehicle`, a wheel only carries additional mass and inertia appended to the suspension's spindle body and, optionally, visualization information;

brake: simple brake (constant torque modulated by the driver braking input).

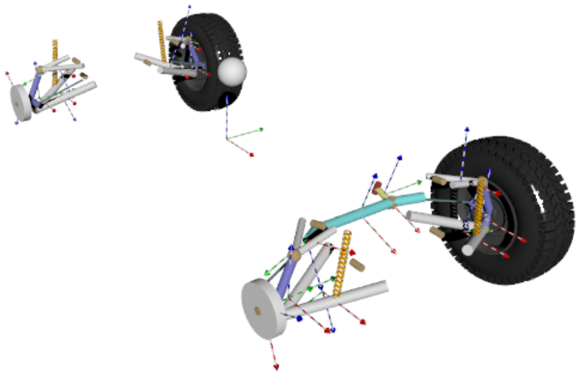


Figure 8.1: Chrono::Vehicle visualization of a HMMWV vehicle with POV-Ray.

For additional flexibility and to facilitate inclusion in larger simulation frameworks, Chrono::Vehicle allows formally separating various systems (the vehicle itself, powertrain, tires, terrain, driver) and provides the inter-system communication API for a co-simulation framework based on force-displacement couplings. For consistency, these systems are themselves templatized:

vehicle: the vehicle template is a collection of references to instantiations of templates for its constitutive subsystems;

powertrain: shaft-based template using an engine model based on speed-torque curves, torque converter based on capacity factor and torque ratio curves, and transmission parameterized by an arbitrary number of forward gear ratios and a single reverse gear ratio;

tire: rigid tire (based on the Chrono rigid contact model), Pacejka, and a Lugre friction tire model;

driver: interactive driver model (with user inputs from keyboard for real-time simulation), file-based driver model (interpolated driver inputs as functions of time).

Visualization support is provided both for run-time, interactive simulation (through the Chrono built-in Irrlicht visualization support) and for high-quality post-processing rendering (using for example the POV-Ray ray-tracing package, see Fig. 8.1).

8.1 HMMWV on flat granular terrain

A flat, deformable terrain, shown in Figs. 8.2 and 8.3, was modeled in `Chrono` to represent a common obstacle that vehicles face in off road operations. The terrain is composed of approximately $300,000$ discrete elements that fill a ditch that is 6 m long, 2.5 m wide, and 0.4 m deep. Each element is an ellipsoid that can be circumscribed in a sphere with a radius of 2 cm. The elements have a coefficient of friction $\mu 0.8$ and a density $\rho = 3.5$ g/cm^3. The APGD solver was used to solve the CCQO at each time step with the maximum number of iterations set to 40. The single wheel simulations were run on an AMD Opteron 6274 2.2 GHz processor using eight cores and took 7.987 hr to simulate 1 s of dynamics.

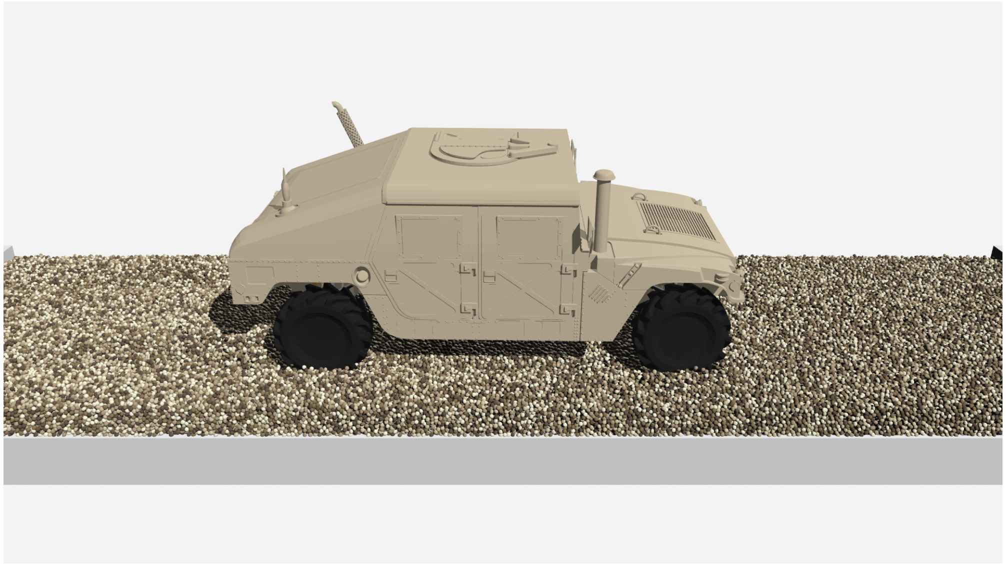


Figure 8.2: A HMMWV operating on granular terrain composed of approximately $300,000$ bodies. Each granular particle is an ellipsoid that can be circumscribed in a sphere with a radius of 2 cm. The solver uses the complementarity form of contact with a time step of 0.001 seconds.

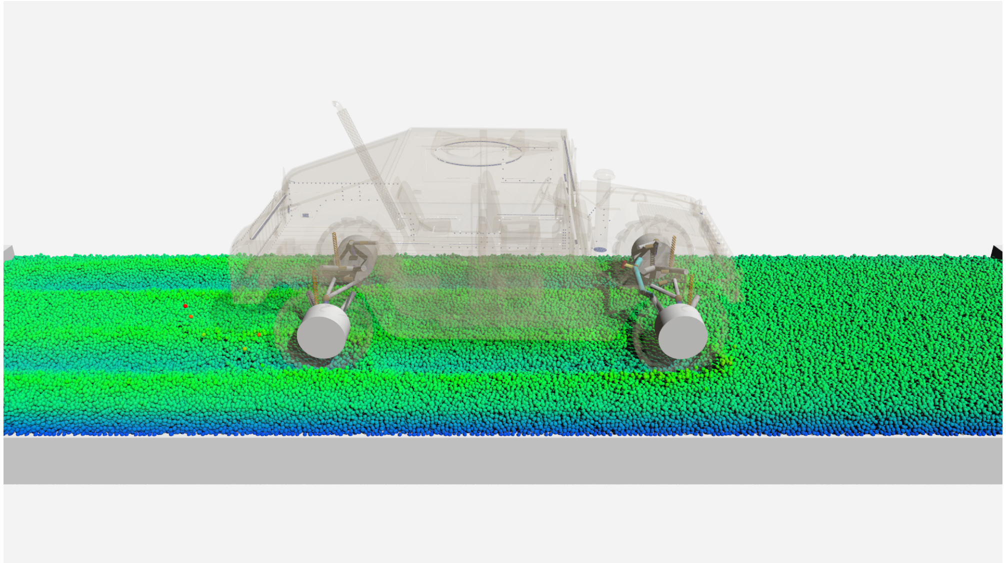


Figure 8.3: A HMMWV operating on granular terrain composed of approximately $300,000$ bodies. Each granular particle is an ellipsoid that can be circumscribed in a sphere with a radius of 2 cm. The solver uses the complementarity form of contact with a time step of 0.001 seconds.

A scaling analysis, shown in Figs. 8.4 - 8.7, was performed to determine the computational requirements for the HMMWV operating on flat, granular terrain. Fig. 8.4 and 8.5 show that the number of bodies scales inversely to the element radius and that the number of bodies is linearly related to the number of contacts. Indeed, there are approximately 4 contacts for each body in the simulation. Fig. 8.6 demonstrates that as the radius of the element is decreased, the residual that is obtained within 40 solver iterations increases. Similarly, the real-time factor increases as more elements are added to the system, shown in Fig. 8.7.

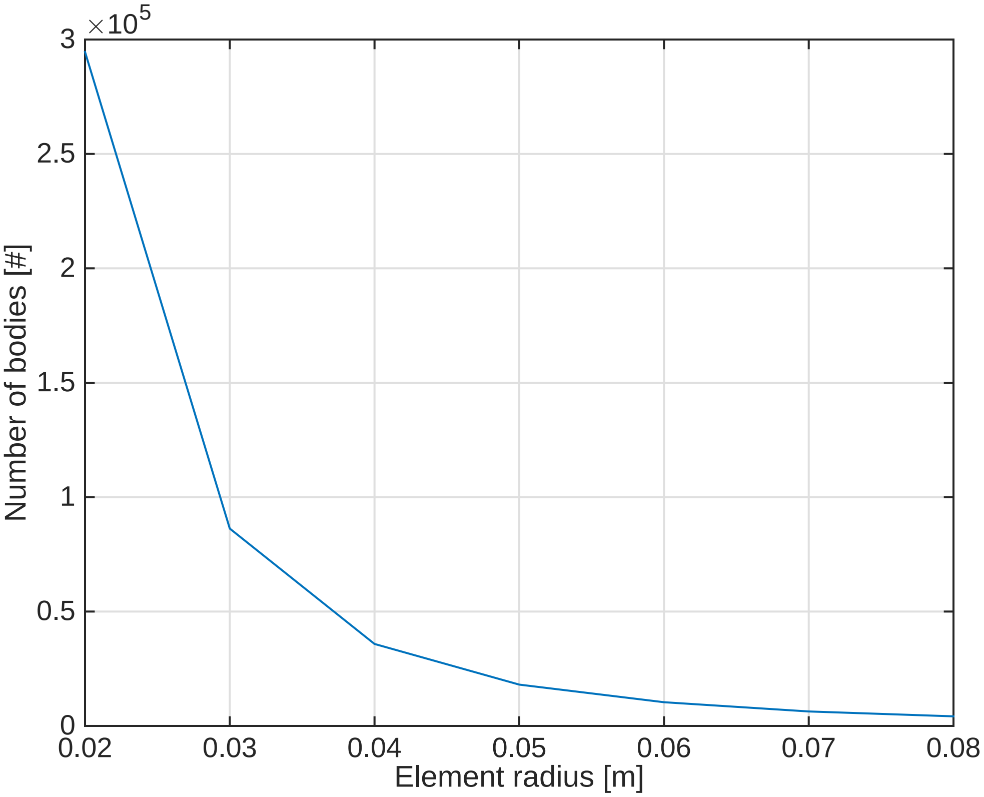


Figure 8.4: The total number of bodies as a function of the element radius.

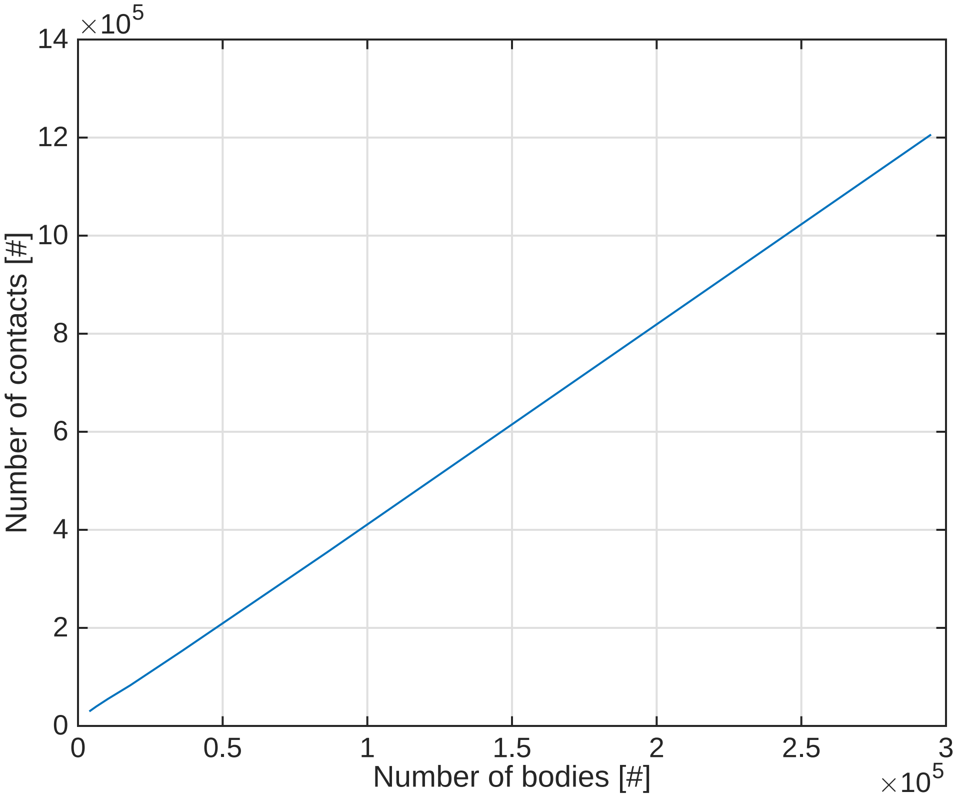


Figure 8.5: The total number of contacts as a function of the average number of bodies.

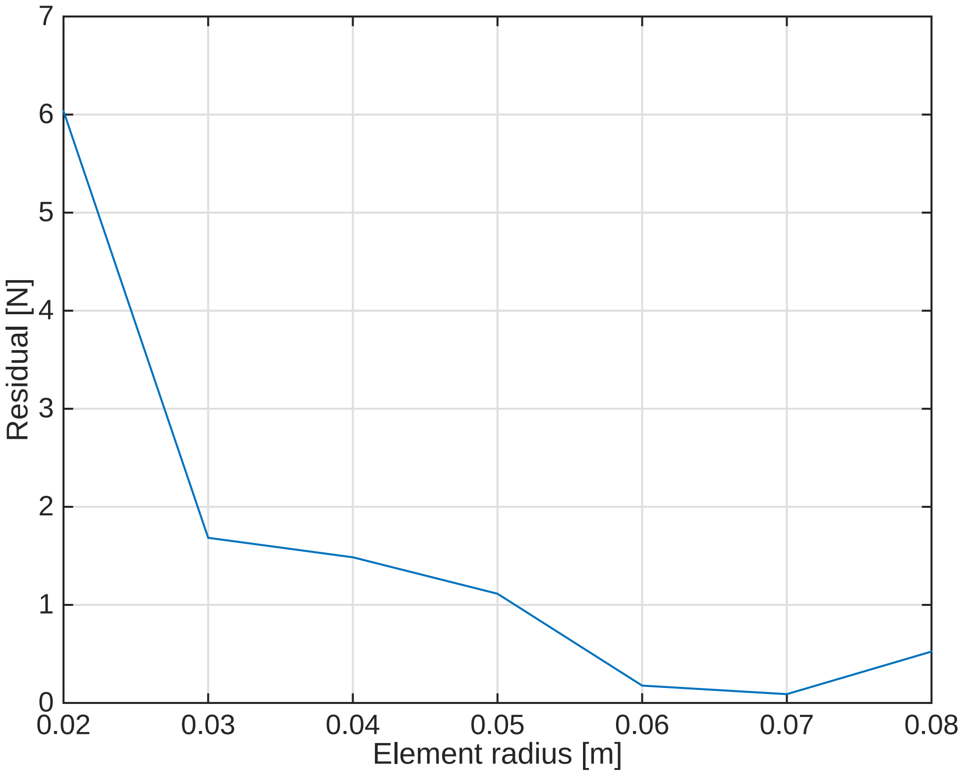


Figure 8.6: The average solver residual per step as a function of the element radius.

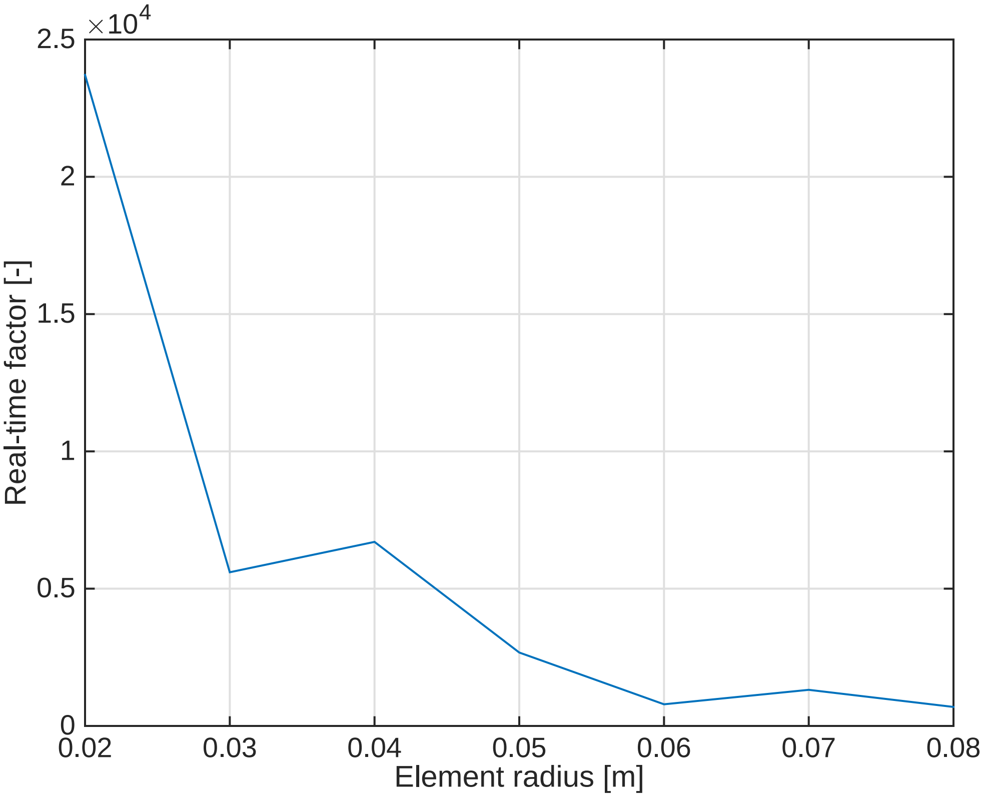


Figure 8.7: The real-time factor as a function of the element radius.

8.2 HMMWV on uneven granular terrain

A bumpy, deformable terrain, shown in Fig. 8.8 and 8.9, was modeled in Chronoto represent a common obstacle that vehicles face in off road operations. The terrain is composed of approximately $400,000$ discrete elements that fill a ditch that is 6 m long, 2.5 m wide, and 0.4 m deep. Each element is an ellipsoid that can be circumscribed in a sphere with a radius of 2 cm. The elements have a coefficient of friction $\mu = 0.8$ and a density $\rho = 3.5$ g/cm^3. The APGD solver was used to solve the CCQO at each time step with the maximum number of iterations set to 40. The single wheel simulations were run on an AMD Opteron 6274 2.2 GHz processor using eight cores and took 8.571 hr to simulate 1 s of dynamics.

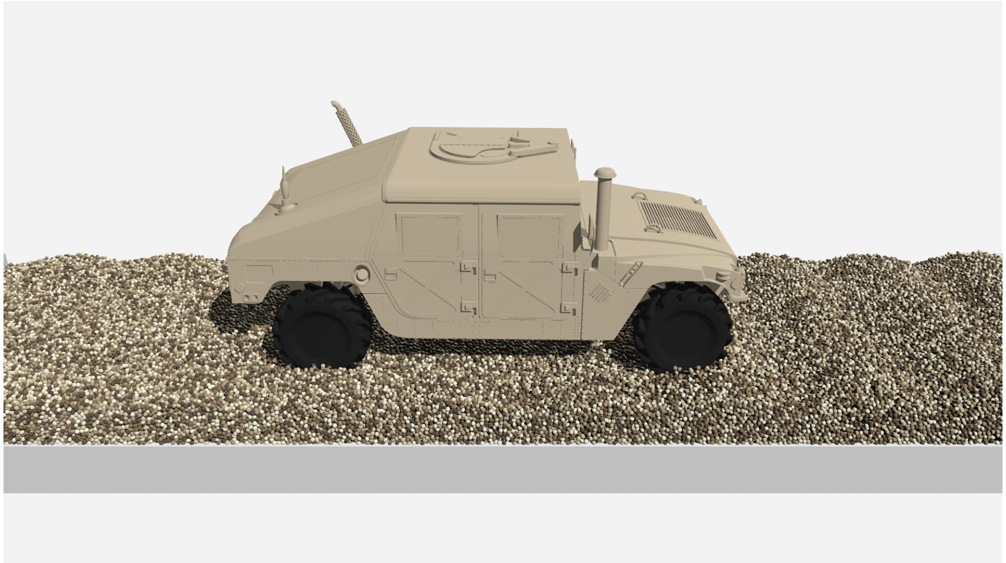


Figure 8.8: A HMMWV operating on bumpy granular terrain composed of approximately $400,000$ bodies. Each granular particle is an ellipsoid that can be circumscribed in a sphere with a radius of 2 cm. The solver uses the complementarity form of contact with a time step of 0.001 seconds.

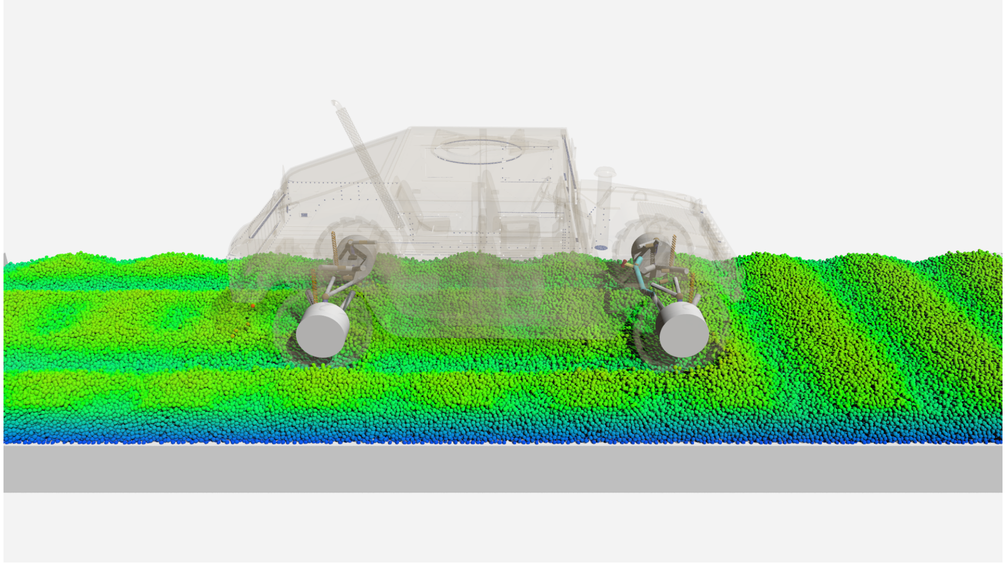


Figure 8.9: A HMMWV operating on bumpy granular terrain composed of approximately $400,000$ bodies. Each granular particle is an ellipsoid that can be circumscribed in a sphere with a radius of 2 cm. The solver uses the complementarity form of contact with a time step of 0.001 seconds.

8.3 HMMWV on sloped granular terrain

A granular slope, shown in Fig. 8.10 and 8.11, was modeled in `Chrono`to represent a common obstacle that vehicles face in off road operations. The terrain is composed of approximately $300,000$ discrete elements that fill a ditch that is 6 m long, 2.5 m wide, and 0.4 m deep. Each element is an ellipsoid that can be circumscribed in a sphere with a radius of 2 cm. The elements have a coefficient of friction $\mu = 0.8$ and a density $\rho = 3.5$ g/cm^3. The APGD solver was used to solve the CCQO at each time step with the maximum number of iterations set to 40. The single wheel simulations were run on an AMD Opteron 6274 2.2 GHz processor using eight cores and took 8.530 hr to simulate 1 s of dynamics.

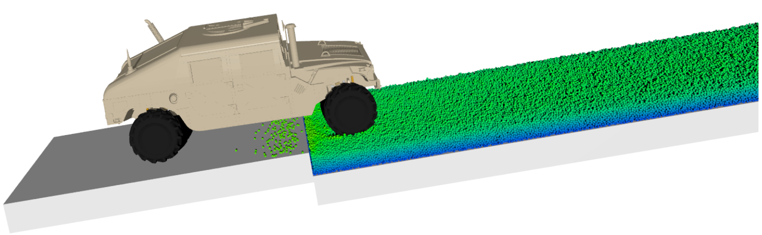


Figure 8.10: A HMMWV operating on sloped granular terrain composed of approximately $300,000$ bodies. Each granular particle is an ellipsoid that can be circumscribed in a sphere with a radius of 2 cm. The solver uses the complementarity form of contact with a time step of 0.001 seconds.

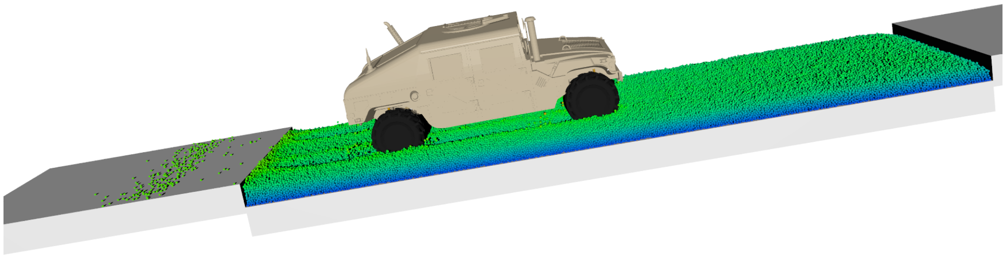


Figure 8.11: A HMMWV operating on sloped granular terrain composed of approximately $300,000$ bodies. Each granular particle is an ellipsoid that can be circumscribed in a sphere with a radius of 2 cm. The solver uses the complementarity form of contact with a time step of 0.001 seconds.

8.4 Flexible tire on granular terrain

A single wheel test, shown in Figs. 8.13 and 8.14, was used to investigate a tire's motion under controlled slip and normal loading conditions within a confined soil bin of dimensions $0.66\times2\times0.2$ m (W × L × H). The drawbar pull, wheel torque, and sinkage were measured for a lug-less flexible tire for several stiffness cases. The tire used in this study had a width of $b = 0.2$ m and an undeformed radius $r_o = 0.3$ m. The numerical tests in were conducted by simulating a soil bin in which $99,005$ spherical bodies with a radius $r = 0.007$ m and density $\rho = 2,500$ kg/m^3 were randomly generated and allowed to settle under gravity. Much like the tests in §6.2.2, the wheel was rolled at a slip ratio of 30% for $T_{final} = 8$ s. A global friction coefficient of $\mu = 0.3$ was used for all of the bodies in the simulation.

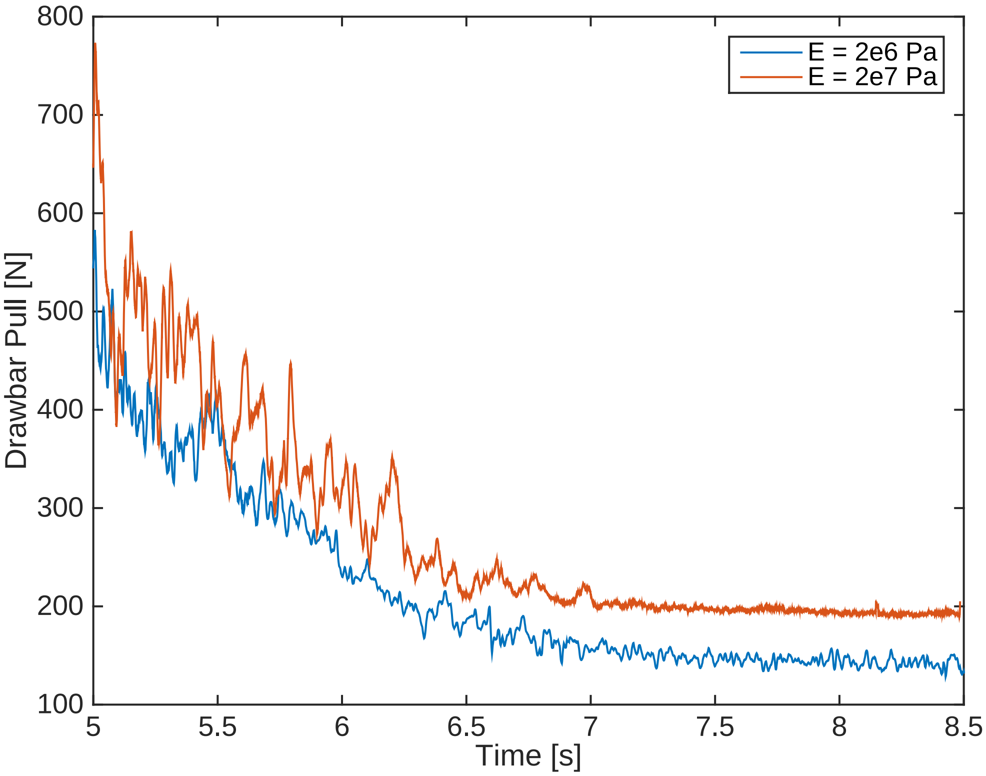


Figure 8.12: The drawbar pull as a function of time for the single wheel test with a flexible tire on deformable soil.

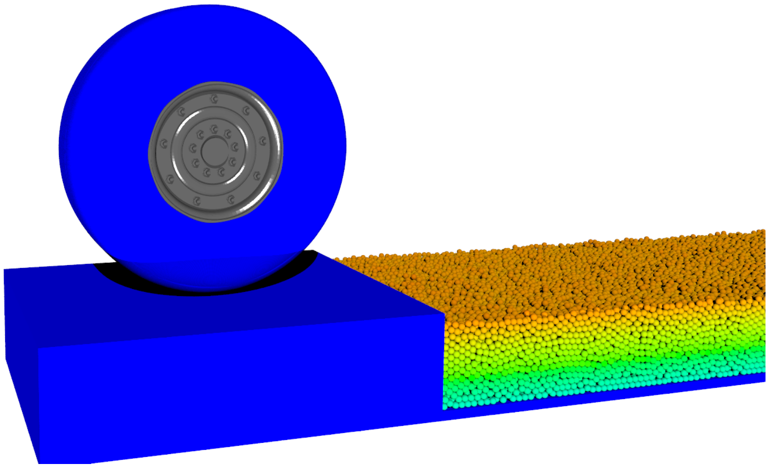


Figure 8.13: A single wheel test of a flexible tire on deformable soil at time $t = 0$ s.

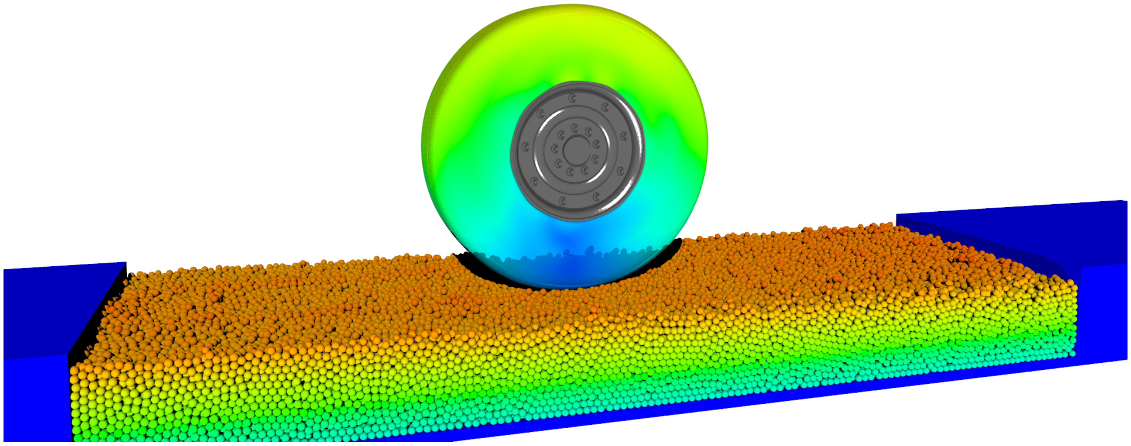


Figure 8.14: A single wheel test of a flexible tire on deformable soil at time $t = 8$ s.

Based on the results shown in Fig. 8.12, it is clear that the tire with the higher stiffness results in a larger drawbar pull for this particular tire geometry. The tire enters into steady-state operation at approximately $t = 7$ s and has an average drawbar pull $D = 145.9$ and 196.3 N for stiffnesses $E = 2 \times 10^6$ and 2×10^7 Pa, respectively.

Chapter 9

Conclusions

This thesis details several enhancements to the complementarity method of contact for discrete element applications in terramechanics. This work is motivated by the degree of fidelity that the discrete element method lends to terramechanics modeling and the advantages that the complementarity formulation provides over alternate contact formulations. Specific enhancements focus on physical modeling and numerical methods, with analytical and experimental techniques used for validation. This basic research is ultimately used to solve a real-world, engineering application, specifically the study of military vehicles over off-road terrain. The specific contributions of this work are summarized as follows:

- Applied contact through complementarity to discrete element simulations in large-scale mobility and terramechanics analyses:
  - Developed several case studies to highlight the potential of DEM with nonsmooth contact for large-scale mobility analysis [**?**]
  - Validated the complementarity approach for contact using standard terramechanics tests (direct shear, pressure-sinkage, and single wheel test) [**?**]
  - Proved the inaccuracy of traditional terramechanics techniques using uncertainty analysis [**?**, **?**, **?**]
  - Demonstrated the convergence of discrete element simulations for systems with over a million degrees of freedom using APGD and identified the inherent numerical challenges when solving a large-scale CCQO [**?**]

  - Characterized the effects of body shape and local friction coefficient on emergent behavior in terramechanics [**?**]

- Investigated new numerical methods for the differential variational inequality formulation which demonstrate improved convergence properties [**?**]:
  - Implemented PDIP to run in parallel with OpenMP or GPU programming
  - Extended and improved the SCIP method for frictional contact problems
  - Developed a consistent termination criteria for the CCQO and linked it to physical phenomena

- Improved the accuracy of the frictional contact solution in the differential variational inequality framework [**?**]:
  - Proved a mathematical artifact in the existing differential variational framework that resulted in non-physical behavior based on a case with an analytical solution
  - Demonstrated that the relaxation of the differential variational framework can be alleviated via an iterative refinement technique
  - Implemented the anti-relaxation via iterative refinement technique for large-scale discrete element simulations on the GPU

- Implemented an efficient model of the DVI-based frictional contact for flexible ANCF tires on deformable terrain [**?**]:
  - Developed a rigid-flexible multiphysics simulation engine on the GPU that relies on DVI-based contact for millions of degrees of freedom
  - Validated rigid-flexible contact for simple scenarios with analytical solutions
  - Demonstrated capabilities through simulation of a single flexible tire operating on granular terrain composed of over $100,000$ terrain bodies

These developments provide an important step towards more accurate and robust simulations. To determine the most appropriate numerical method for solving the large-scale, nonsmooth dynamics problem, three first-order methods (PJ, PGS, APGD) and three second-order methods (PDIP, SCIP, P-SCIP) have been compared when used in conjunction with the time evolution of collections of rigid bodies interacting through friction and contact. At each time step, these methods solve a conically constrained quadratic optimization problem whose solution yields the friction and normal forces at each contact point. The six methods drew on parallel computing on GPU cards to carry out sparse-matrix operations and, for PDIP and P-SCIP, to solve linear systems associated with the interior point method. Three types of numerical experiments; i.e., drafting, filling, and compression, were considered to evaluate the accuracy and efficiency of the methods. Special attention was paid to providing a fair comparison across the different solvers by ensuring that the definition of the solution accuracy metric was consistent across the methods. The numerical experiments reported here lead to the conclusion that for packed-body scenarios such as granular dynamics or terrain modeling, APGD is the most efficient; i.e., it reaches a certain level of accuracy in a shorter amount of time. For problems that involve less than 500 bodies, the second-order methods perform better. Overall, the interior point methods require much fewer iterations than the first-order solvers with the SCIP method requiring the least. The interior point methods are computationally more expensive due in large part to the linear system that must be solved at each iteration of the interior point algorithm. Finally, while PDIP and SCIP are relatively similar in terms of performance, for the first-order methods there is a wide efficiency gap between the performance of APGD on the one hand, PJ and PGS on the other hand.

Along with determining the most suitable numerical method for solving the nonsmooth dynamics problem, the accuracy of the current frictional contact solution was improved. The DVI model is based on complementarity conditions for contact and differential inclusions for handling friction forces. By adding a term to mathematically relax the DVI, the model can be posed as a CCP and the solution can be found by solving a CCQO. In some cases, such as a ball that transitions from pure sliding to perfect rolling, the relaxation results in non-physical behavior. In this work, the objective function of the CCQO is modified by an anti-relaxation term to result in a solution that is equivalent to the optimum of the original DVI. Three numerical experiments were carried out to analyze the effect of anti-relaxation. The first numerical experiment thoroughly analyzed the simple case of a sphere transitioning from pure sliding to rolling and described in graphical form the error that relaxation can introduce. The second experiment investigated the effect of anti-relaxation over time for a filling test with $1,000$ spheres. It was determined that the anti-relaxation results in less noise, or "jitter", in the settled configuration and results in a complementarity error that is several orders of magnitude lower than the relaxed solution. The last experiment looked at several filling tests, each with a different number of bodies, to investigate the effect of the anti-relaxation as a function of the number of collisions. It was found that the number of iterations are similar for low numbers of collisions (below 100 spheres) but the anti-relaxed solution quickly becomes increasingly costly as the number of collisions increases.

Additionally, a DVI formulation was used to model frictional contact between flexible bodies formulated with ANCF to simulate large flexible multibody systems. The DVI model is based on complementarity conditions for contact and differential inclusions for handling friction forces. This contact model is preferred over penalty methods because it has no restriction on the size of the time step and requires only a single parameter to be specified. To demonstrate the lack of restriction on time step, a model of a cloth constructed from gradient-deficient plate elements is dropped on a rigid sphere. The results for position and contact force demonstrate that convergence can be obtained by varying the number of elements and this convergence is independent of the size of the time step. To highlight the advantages that DVI provides in parameter identification, a model of a tire is constructed and matched to experimental data from a HMMWV tire by simply varying the friction coefficient. Finally, a single wheel test is performed with a flexible tire on deformable terrain to investigate the effect of tire stiffness on traction. The results indicate that a higher tire stiffness results in a higher drawbar pull for the tire.

Lastly, the predictive attribute of the DVI-based frictional contact model was validated for fundamental terramechanics problems, including a single wheel test in which the soil is represented using a large number of discrete elements. The DVI model is based on complementarity conditions for contact and differential inclusions for handling friction forces. The modeling methodology has been implemented in an open source dynamics engine called `Chrono`, which is used for soft soil ground vehicle mobility studies. The conclusion of this study is that the predictive attribute of the modeling methodology, as exposed by its implementation in `Chrono`, is good. Indeed, a unique set of solution and model parameters were used to match experimental data in three tests: granular material shearing, pressure sinkage, and drawbar pull at various wheel slip levels. The simulation times are larger than hoped for, yet it is not clear whether a penalty-based DEM approach would have yielded a numerical solution more expeditiously. This tuned and validated terrain model was demonstrated in a simulation of a full-scale, high-fidelity HMMWV model traversing several off-road scenarios, including a flat granular terrain, uneven terrain, and a sloped terrain.

This work advances the state of the art in DVI formulations for handling friction and contact with applications in vehicle mobility investigations via discrete element simulations. Simulation efficiency was increased through better numerical algorithms. Simulation robustness was improved through iterative refinement techniques. The physical modeling capability has been expanded by the incorporation of flexible multibody dynamics. Finally, the method has been validated for several complex scenarios relevant to terramechanics.

Chapter 10

Future work

The results of this thesis present several directions for continued work. Regarding the numerical solvers, future work is focused on investigating additional improvements to the PDIP solver. The μ solver parameter, for example, has been observed to have a large impact on the solver convergence. Since the solver is capable of superlinear convergence given a suitable guess value, "warm-starting" methods are being investigated to use information from the previous time step. Lastly, computing a new preconditioner at every iteration is very expensive which motivates the investigation of preconditioner reuse.

Perhaps the most intriguing result of this work, "anti-relaxation" via iterative refinement of the conically constrained, quadratic optimization problem warrants additional investigation. Current results indicate that iterative refinement results in a solution that satisfies the original complementarity solutions, but the solution comes at a higher cost. Future work would focus on proving that this technique is guaranteed to converge and comparing it to other methods for solving the exact CCP problem [**?**]. Since this method effectively computes a velocity at the next time step, continued effort should be focused on seeing if this technique can be used for fully-implicit time-stepping schemes. Although the current, symplectic scheme has been shown to be adequate for rigid body scenarios, a fully-implicit scheme would aid in accurately satisfying bilateral constraints and solving flexible body simulations at large time steps.

As always, continued validation work would be valuable. Although this work shows that the method can accurately predict a variety of fundamental terramechanics tests, these tests are restricted to longitudinal motion. Testing cases for lateral movement, such as the slip angle test for wheels, would increase our confidence in the method. Additionally, applying statistical techniques for uncertainty estimation and parameter investigation would give greater insight into the capabilities of the terrain model.

D2P

Appendix A: Algorithms

The algorithms below draw on notation introduced in Chapter 4 and several other quantities defined below.

A.1 Projected Gauss-Jacobi method

In both PJ and PGS, the matrix $\mathbf{J}$ is block diagonal. Each 3×3 block $\mathbf{J}_i$ is given as $\mathbf{J}_i = \frac{1}{j_i}\boldsymbol{I}$, where

$$j_i = \frac{tr\left(\boldsymbol{D}_i^T \boldsymbol{M}^{-1} \boldsymbol{D}_i\right)}{3}, \tag{A.1}$$

The PJ method used parameters $\omega = 0.3$ and $\lambda = 2/3$.

```
ALGORITHM JACOBI($\mathbf{N}$, $\mathbf{r}$, $\tau$, $N_{max}$, $\gamma_0$)
for $k := 0$ to $N_{max}$
    $\hat{\gamma}_{(k+1)} = \Pi_{\mathcal{K}}\left(\gamma_{(k)} - \omega \mathbf{J}\left(\mathbf{N}\gamma_{(k)} + \mathbf{r}\right)\right)$
    $\gamma_{(k+1)} = \lambda \hat{\gamma}_{(k+1)} + (1 - \lambda)\gamma_{(k)}$
    $r = r\left(\gamma_{(k+1)}\right)$
    if $r < \tau$
        break
endfor
return Value at time step $t^{(l+1)}$, $\gamma^{(l+1)} := \gamma_{(k+1)}$ .
```

A.2 Projected Gauss-Seidel method

The PGS method used parameters $\omega = 0.3$ and $\lambda = 2/3$.

```
ALGORITHM GAUSS-SEIDEL($\mathbf{N}$, $\mathbf{r}$, $\tau$, $N_{max}$, $\gamma_0$)
for $k := 0$ to $N_{max}$
    for $i = 1$ to $n_c$
        $\hat{\gamma}_{i,(k+1)} = \Pi_{\mathcal{K}}\left(\gamma_{i,(k)} - \omega \mathbf{J}_i \left(\mathbf{N}\gamma_k + \mathbf{r}\right)_i\right)$
        $\gamma_{i,(k+1)} = \lambda \hat{\gamma}_{i,(k+1)} + (1-\lambda)\,\gamma_{i,(k)}$
    endfor
    $r = r\left(\gamma_{k+1}\right)$
    if $r < \tau$
        break
endfor
return Value at time step $t^{(l+1)}$, $\gamma^{(l+1)} := \gamma_{(k+1)}$ .
```

A.3 Accelerated projected gradient descent method

ALGORITHM APGD($\mathbf{N}$, $\mathbf{r}$, τ, N_{max})

```
$\gamma_0 = \mathbf{0}_{n_c}$
$\hat{\gamma}_0 = \mathbf{1}_{n_c}$
$\mathbf{y}_0 = \gamma_0$
$\theta_0 = 1$
$L_k = \frac{||\mathbf{N}(\gamma_0 - \hat{\gamma}_0)||_2}{||\gamma_0 - \hat{\gamma}_0||_2}$
$t_k = \frac{1}{L_k}$
for $k := 0$ to $N_{max}$
    $\mathbf{g} = \mathbf{N}\mathbf{y}_k + \mathbf{r}$
    $\gamma_{k+1} = \Pi_{\mathcal{K}}\left(\mathbf{y}_k - t_k g\right)$
    while $\frac{1}{2}\gamma_{k+1}^T\mathbf{N}\gamma_{k+1} + \gamma_{k+1}^T\mathbf{r} \geq \frac{1}{2}\mathbf{y}_k^T\mathbf{N}\mathbf{y}_k + \mathbf{y}_k^T\mathbf{r} + \mathbf{g}^T\left(\gamma_{k+1} - \mathbf{y}_k\right) + \frac{1}{2}L_k||\gamma_{k+1} - \mathbf{y}_k||_2^2$
        $L_k = 2L_k$
        $t_k = \frac{1}{L_k}$
        $\gamma_{k+1} = \Pi_{\mathcal{K}}\left(\mathbf{y}_k - t_k g\right)$
    endwhile
    $\theta_{k+1} = \frac{-\theta_k^2 + \theta_k\sqrt{\theta_k^2 + 4}}{2}$
    $\beta_{k+1} = \theta_k\frac{1-\theta_k}{\theta_k^2 + \theta_{k+1}}$
    $\mathbf{y}_{k+1} = \gamma_{k+1} + \beta_{k+1}\left(\gamma_{k+1} - \gamma_k\right)$
    $r = r\left(\gamma_{k+1}\right)$
    if $r < \epsilon_{min}$
        $r_{min} = r$
        $\hat{\gamma} = \gamma_{k+1}$
    endif
    if $r < \tau$
        break
    endif
    if $\mathbf{g}^T\left(\gamma_{k+1} - \gamma_k\right) > 0$
        $\mathbf{y}_{k+1} = \gamma_{k+1}$
        $\theta_{k+1} = 1$
    endif
    $L_k = 0.9L_k$
    $t_k = \frac{1}{L_k}$
endfor
return Value at time step $t_{l+1}$, $\gamma^{l+1} := \hat{\gamma}$ .
```

A.4 Primal-dual interior point method

The PDIP method used parameters $\chi = 10$, $\alpha = 0.01$, and $\beta = 0.8$.

```
ALGORITHM PDIP($\mathbf{N}$, $\mathbf{r}$, $\tau$, $N_{max}$, $\gamma_0$)
    $\mathbf{f} = \mathbf{f}\left(\gamma_0\right)$
    $\lambda_0 = -1/\mathbf{f}$
    for $k := 0$ to $N_{max}$
        $\mathbf{f} = \mathbf{f}\left(\gamma_k\right)$
        $\hat{\eta} = -\mathbf{f}^T\lambda_k$
        $t = \frac{\chi m}{\hat{\eta}}$
        $\mathbf{A} = \mathbf{A}\left(\gamma_k, \lambda_k, \mathbf{f}\right)$
        $\mathbf{r}_t = \mathbf{r}_t\left(\gamma_k, \lambda_k, t\right)$
        Solve the linear system $\mathbf{Ay} = -\mathbf{r}_t$
        $s_{max} = sup\{s \in [0,1]\,|\lambda + s\Delta\lambda \succeq 0\} = min\{1, min\{-\lambda_i/\Delta\lambda_i|\Delta\lambda_i < 0\}\}$
        $s = 0.99s_{max}$
        while $max\left(\mathbf{f}\left(\gamma_k + s\Delta\gamma\right) > 0\right)$
            $s = \beta s$
        endwhile
        while $||\mathbf{r}_t\left(\gamma_k + s\Delta\gamma, \lambda_k + s\Delta\lambda\right)||_2 > \left(1 - \alpha s\right)||\mathbf{r}_t||_2$
            $s = \beta s$
        endwhile
        $\gamma_{k+1} = \gamma_k + s\Delta\gamma$
        $\lambda_{k+1} = \lambda_k + s\Delta\lambda$
        $r = r\left(\gamma_{k+1}\right)$
        if $r < \tau$
            break
        endif
    endfor
    return Value at time step $t_{l+1}$, $\gamma^{l+1} := \gamma_{k+1}$ .
```

A.5 Symmetric cone interior point method

```
ALGORITHM SCIP($\mathbf{N}$, $\mathbf{r}$, $\tau$, $N_{max}$, $\gamma_0$)
    $\bar{N} = T_y N T_x^{-1}$
    $\bar{\mathbf{r}} = T_y \mathbf{r}$
    Guess a feasible initial $\mathbf{x}^{(0)} = T_x \gamma^{(0)}$
    Construct $\mathbf{d}$ and $s^{(0)}$ as in Eqs. 4.25 - 4.27
    $\bar{\mathbf{y}}^{(0)} = \bar{N}\bar{\mathbf{x}}^{(0)} + \mathbf{r} + s^{(0)}\mathbf{d}$
    Set $feasible = 0$
    for $k := 0$ to $N_{max}$
        Calculate $P(\mathbf{w})$ using Eqs. 4.33 and 4.34
        $A = P(\mathbf{w}) + \bar{N}$
        $\Delta s^{(k)} = (1 - feasible) \cdot (2\alpha - s^{(k)})$
        $\mathbf{b} = \alpha(\mathbf{x}^{(k)})^{-1} - \bar{\mathbf{y}}^{(k)} - \Delta s^{(k)}\mathbf{d}$
        Solve $A\Delta\mathbf{x} = \mathbf{b}$
        Calculate stepping length $\theta$ using Eq. 4.31
        $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \theta\Delta\mathbf{x}$
        $\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \theta(\bar{N}\Delta\mathbf{x} + \Delta s^{(k)}\mathbf{d})$
        $s^{(k+1)} = s^{(k)} + \theta\Delta s^{(k)}$
        if $feasible = 0$ and $\bar{\mathbf{y}}^{(k+1)} - s^{(k+1)}\mathbf{d} \in$ int $\Upsilon$ then
            $\bar{\mathbf{y}}^{(k+1)} = \bar{\mathbf{y}}^{(k+1)} - s^{(k+1)}\mathbf{d}$
            $s^{(k+1)} = 0$
            $feasible = 1$
        $\alpha = \beta\frac{\mathbf{x}^{(k)T}\bar{\mathbf{y}}^{(k)}}{2N_c}$ for some $\beta \in (0, 1]$
        $\gamma^{(k+1)} = T_x^{-1}\mathbf{x}^{(k+1)}$
        if $r < \tau$
            break
        endif
    endfor
    return Value at time step $t_{l+1}$, $\gamma^{l+1} := \gamma^{(k+1)}$ .
```

Appendix B: Additional plots

Additional anti-relaxation plots.

B.1 Surface plots of the objective functions

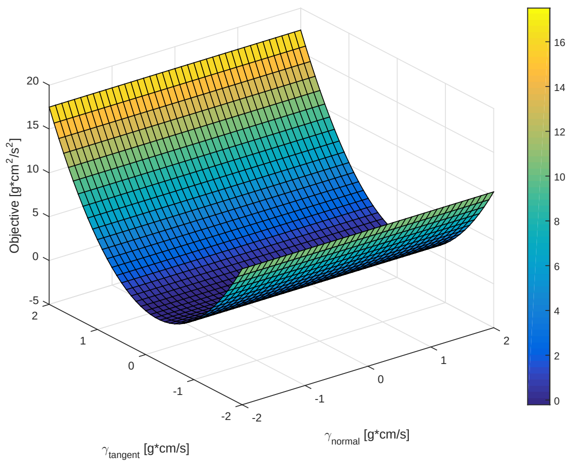


Figure B.1: The original DVI problem based on Eq. 3.8.

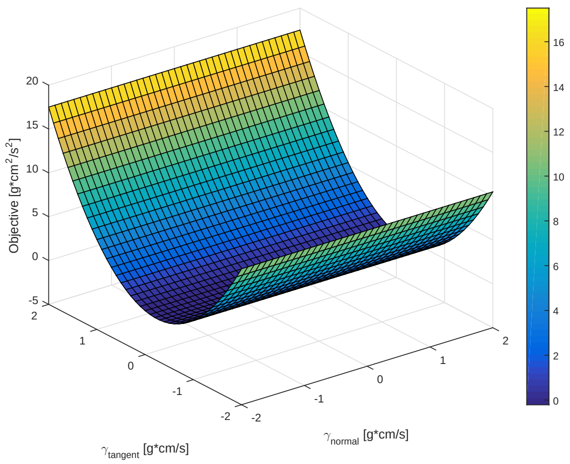


Figure B.2: The relaxed DVI/CCP problem based on Eq. 3.9.

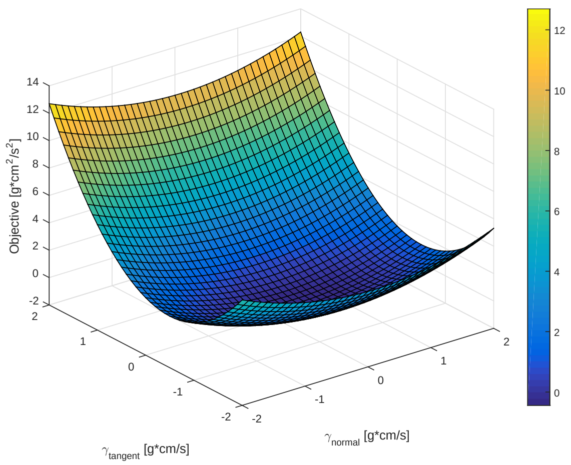


Figure B.3: The relaxed CCQO based on Eq. 3.12.

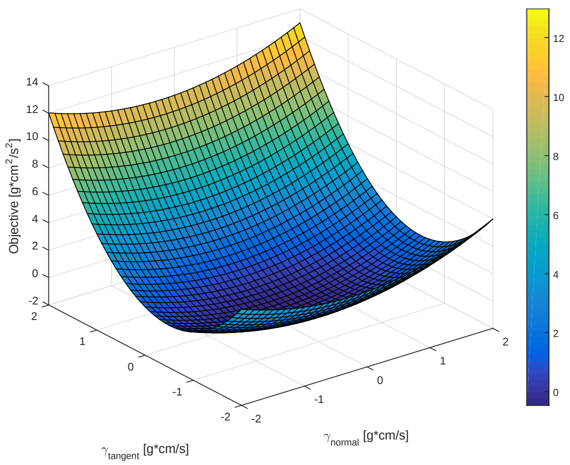


Figure B.4: The anti-relaxed CCQO based on Eq. 5.1.

B.2 Constraint plots with objective contours

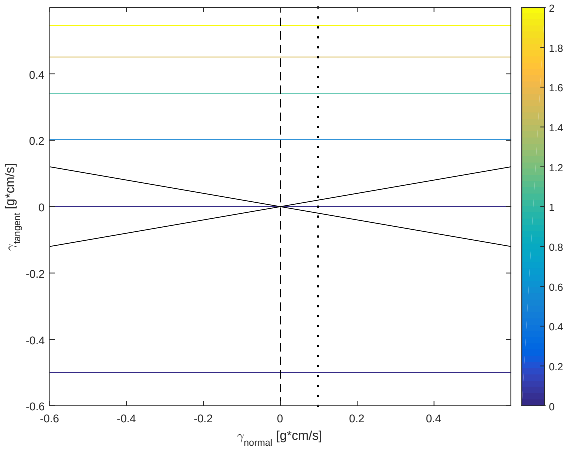


Figure B.5: The original DVI problem based on Eq. 3.8.

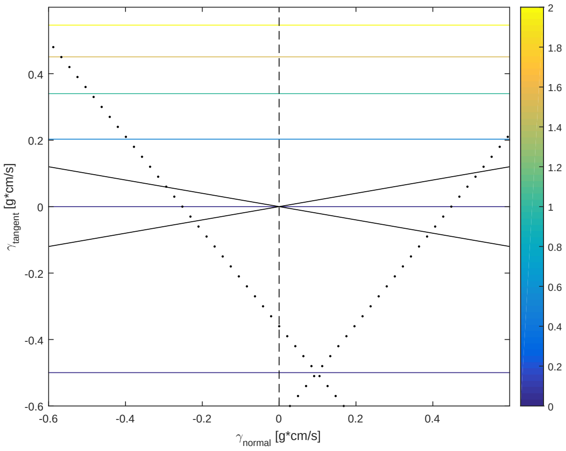


Figure B.6: The relaxed DVI/CCP problem based on Eq. 3.9.

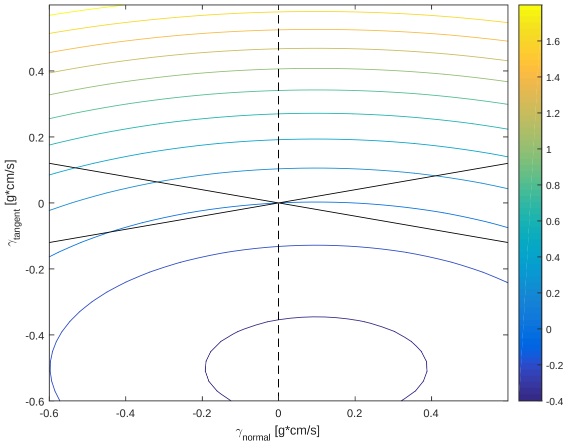


Figure B.7: The relaxed CCQO based on Eq. 3.12.

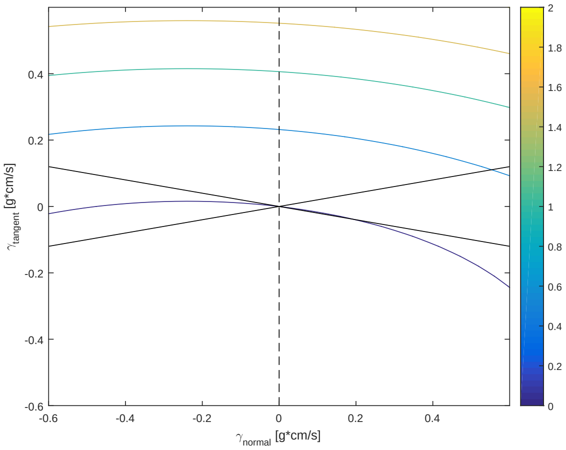


Figure B.8: The anti-relaxed CCQO based on Eq. 5.1.