Pain and Agony
using a Newton Based Method

Sarah Maria Niebe

November 26, 2009

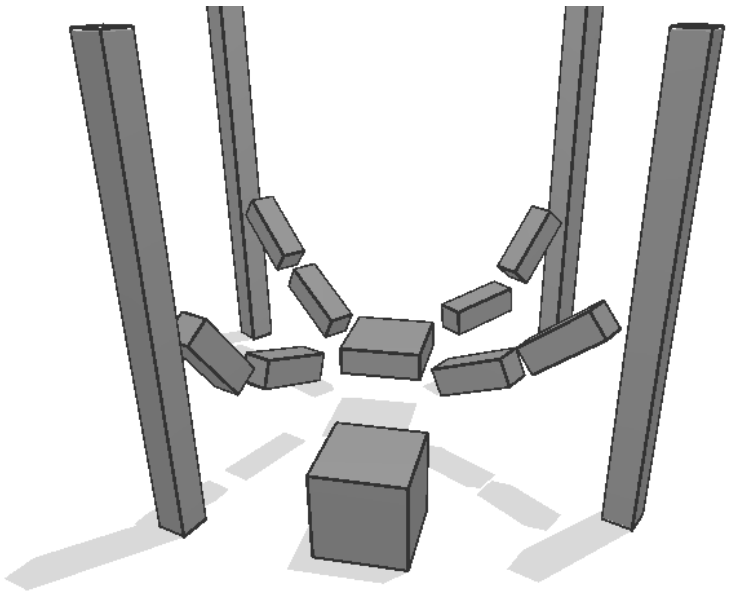


Figure 1: *Screenshot of a simulator using the Fischer-Newton method.*

Department of Computer Science, University of Copenhagen
Universitetsparken 1, DK-2100 Copenhagen East, Denmark
niebe @ diku.dk

Master's Thesis, 30 ECTS

Supervisor: Kenny Erleben

Abstract

The subject of this thesis is contact force determination, specifically for use in interactive rigid body simulation. The aim of the thesis is twofold, to overthrow the projected Gauss-Seidel method and replace it with a more accurate and robust method, and to propose a more accurate friction model than the one used in most interactive simulators.

Part of the thesis is presented in the form of a paper [SNE09]. The paper was accepted for the 2009 workshop on Virtual Reality and Interactive Physical Simulation(VRIPHYS), held in Karlsruhe. Together with RAMBØLL A/S and NVIDIA, the image group at DIKU co-funded the travel expenses for four students, recognizing potential of the work presented in the four accepted papers.

Although the method presented in [SNE09] has potential, it fails to be interactive for more than just small to medium sized simulations. For that reason, I have taken the first steps towards large scale interactive simulations, by posing a model based on proximal mappings. The proximal mapping model is independent of the friction model, making the friction model replaceable by any parameterization of a convex set. The implementation of this proximal mapping model remains to be done, but it shows potential.

Abstract

Emnet for dette speciale er bestemmelse af kontaktkræfter, helt konkret med henblik på anvendelse i interaktive simuleringer af stive legemer. Målet for specialet er todelt, dels at forkaste den projicerede Gauss-Seidel metode og erstatte den med en mere præcis og robust metode, og dels at foreslå en mere præcis friktionsmodel end den som er anvendt i de fleste interaktive simulatorer.

En del af dette speciale er givet i form af en artikel [SNE09]. Artiklen blev optaget til 2009's workshop on Virtual Reality and Interactive Physical Simulation(VRIPHYS), afholdt i Karlsruhe. I samarbejde med RAMBØLL A/S og NVIDIA, har billedgruppen på DIKU dækket rejseomkostninger for fire studerende, en anerkendelse af de fire optagne artiklers potentiale.

Selvom der var potentiale i den præsenterede metode i [SNE09], så er den kun interaktiv for små og mellem størrelses simuleringer. Derfor har jeg taget de første skridt i retning af en interaktiv simulator for store systemer, ved at fremlægge en model som er baseret på *tætteste punkt* funktioner. Denne model er uafhængig af friktionsmodellen, hvilket betyder at friktionsmodellen kan erstattes med en hvilken som parameterisering af en konveks mængde. Tætteste punkt modellen er endnu ikke implementeret, men der er klart potentiale.

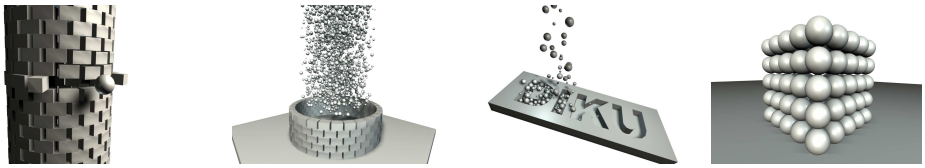
Contents

1 Introduction

Multibody dynamics is the art of simulating the physical interactions of rigid bodies in a virtual world. Depending on the application, the importance of physical correctness may vary greatly. For engineering purposes, the level of physical correctness will take top priority, which is often at the expense of interactivity. On the other hand, when used in a computer game, the physical simulations only need to be plausible whereas interactivity is of utmost importance.

Using the terminology *physical correctness* is problematic for many reasons. Each step from real world observation to the final simulation consists of idealizations and discretizations. The mathematical models are idealized descriptions of empirical observations, the numerical model is a discretization of the mathematical model and so forth. In this context, the rigid body assumption is just another approximation. Still, there exist cases where the rigid body assumption is a fair approach, examples include assembly line simulation [FAS] and robot simulation [Gaz]. Instead of physical correctness, I shall use the terminology *plausibility*.

A multibody dynamics simulator is a complex system of many parts, the focus of this thesis is on the specific problem of determining contact forces. When two rigid bodies are in contact, the resulting interactions can be described as a set of contact forces or impulses acting on the two bodies. Contact forces ensure that my tea mug stays firmly on the table, rather than sinking into it or sliding along it. When a brick is thrown at a wall,



(a) (b) (c) (d)

Figure 2: *Renderings of typical rigid body test setups, testing collisions and stacking. Borrowed from the OpenTissue [OT] library.*

contact forces determine what happens to both brick and wall. The accuracy of the computed contact forces affects both plausibility and stability of the final simulation [Erl07]. For this reason, good contact force determination methods are wanted in physical simulation, whether it is used in engineering systems or computer games.

1.1 Motivation and Goals

Taking a master's degree is a specialization process, a fine tuning of skills. Early on, I had the idea that my specialization would be in imaging and robotics, but it never really caught on. Somehow, I found my self taking courses on physical simulation, writing projects about physical simulation and overall focusing on this specific area. When I try to explain what it is that fascinates me, it is the multitude of applications.

I have formulated a set of goals for this thesis, of which I would like to list three:

1. To describe the contact force determination problem.
2. To model the contact force determination problem.
3. To implement a solution method for the contact force determination problem.

As different applications pose different requirements to the simulation method used, I have tried to specify what application I wish to aim at. As mentioned, there is a multitude of applications, robotics, engineering, computer games, biomechanics just to mention a few. Having taken courses in optimization, I lean towards high accuracy. However, I prefer not having to wait minutes, hours or even days for a simulation to be complete, so I also lean towards real time or at least interactivity. An application demanding both accuracy and interactivity could be a virtual prototyping system. By using an interactive simulator to construct a roughly sketched prototype, an engineer could try out several ideas before committing to a design. For a virtual prototyping system to be usable, it has to have a high level of accuracy, it has to be able to handle large-scale simulations of several thousand bodies, it has to be ro-

bust enough to handle known simulation issues, and it has to be interactive. Using this imagined application, I will have something to evaluate against.

1.2 Work Process

Prior to this thesis, I have completed two projects which served as preparation for the work presented here. The first project was a strictly theoretic treatment of the Fischer reformulation of a linear complementarity problem, examining its properties and applicability to the generalized Newton method [Nie09b]. The second project was more implementation oriented, an implementation of the generalized Newton method was presented. However, the method was restricted to solving linear complementarity problems. The following sections should reveal why this is not satisfactory for solving the contact force problem. Still, the results of this project, [Nie09a], held some promise of benefit from using the Fischer reformulation.

I was presented with the opportunity to write part of my thesis in the form of a paper, co-authored by my fellow student Morten Silcowitz and my supervisor Kenny Erleben. By doing so, I have gained the experience of writing a research paper and having it reviewed by experts in the field. The paper was accepted, which meant that we – the authors – went to give an oral presentation at the VRIPHYS 2009 workshop, in Karlsruhe early November 2009. The combined experience of writing and submitting a paper, attending a workshop, international networking and being part of a research community has given me the taste for more. Had anyone asked me six months ago, what were my plans after submitting my thesis, the answer would have been a nice job in the private sector. Now however, I intend to apply for a PhD, hoping I can continue to contribute to the research community.

Besides the VRIPHYS paper, working on this thesis has resulted in the co-authoring and submission of two additional papers. These are included as Appendices B and C. Decision of acceptance is still pending.

1.3 Background

Multibody dynamics was introduced to the Computer Graphics community in the late 1980's, primarily focused on impulse–based and penalty–based methods [Hah88, MW88]. There are three paradigms in multibody dynamics. In addition to the impulse–based and penalty–based methods, the third paradigm is constraint–based methods.

Baraff [Bar89] presents an early constraint–based simulator, using an acceleration–based formulation of the equations of motion. Baraff states the contact force problem as a quadratic problem. The contact force problem is solved using a heuristic solution method, rather than a direct solution method. Friction is not included in this model, as friction is likely to cause problems for the selected solution method. In [Bar91] the lack of friction is rectified, as Baraff introduces two different approaches for including friction in the contact force problem.

Baraff claims that the main problem is static friction, so the first friction model uses dynamic friction to model static friction. This is done by using a threshold value on the relative tangential velocity, to distinguish between dynamic and static friction. Unfortunately, this introduces creeping in the contacts. Although the visual effect of this creeping can be removed by lowering the threshold value, the added energy in the system is unwanted.

The second model proposed uses a decoupling of the two tangential friction directions, thus linearizing the static friction constraint. For one–dimensional friction, this model is equivalent to Coulomb's friction model. When the dimensionality is expanded, this model will most likely overestimate the friction force.

Alart and Curnier [AC91] use a variational approach to the contact force problem, modeling constraints as inclusions. The system is then formulated as a projection scheme, and solved by a generalized Newton method.

In [Bar94], Baraff continues to work on the constraint–based acceleration formulation, using the linearized friction model proposed in [Bar91]. The novel contribution is in using a pivotal method for solving the resulting complementarity problem, rather than reformulating the problem as a quadratic programming problem as done in his previous works on the subject. There

are claims of interactivity for two–dimensional simulations, where three–dimensional simulations are still offline. As Mirtich and Canny [MC95] point out, the constraint–based method using an acceleration–based formulation can not handle collisions. Contact and collisions are dealt with separately, and by two different methods. Instead, they propose a method based on the early work of [Hah88]. Contacts are modeled as a series of micro-collisions. Although the method in [Hah88] is often mentioned in literature along side other impulse–based methods, Mirtich and Canny claim to be the fathers of the method as well as its name.

When solving a computational problem, a guarantee of solution existence is preferred. The method presented in [Bar94] had no such guarantee. Stewart and Trinkle [ST96] presented a novel approach to modeling contact force problems. Their model guarantees solution existence but not uniqueness of the solution. They use the constraint–based paradigm, but in contrast to previous literature, the formulation is velocity–based and not acceleration–based. The model is based on an inner polyhedral friction cone approximation, the accuracy of the approximation is controlled by the amount of facets used. The model is in the form of a linear complementarity problem. In [AP97], the Stewart and Trinkle model is extended to include frictionless joints.

Chatterjee [Cha99] lashes out at the use of complementarity conditions on a velocity level, claiming there is no physical basis for such formulations. However, as Chatterjee also mentions, the formulations have the benefit of internal mathematical consistency.

A survey of solution methods, for models similar to the one presented in [AC91], is presented in [JAJ98]. The methods are iterative methods, categorized in three different groups: Newton methods, Successive-Over-Relaxation(SOR) methods and quasi-Newton methods. Their studies indicate that the SOR methods are favorable. Due to the nonlinear nature of the equation system, there is no proof of convergence properties only empirical observations.

The algorithm presented in [MS04] is clearly aimed at computer animation, interactivity and accuracy are not part of their considerations. They use a partial solution of the contact force problem combined with a freez-

ing technique to reduce the size of the contact problem, the result is an inaccurate simulation with a plausible visual result. Even when using these optimization tricks, the simulation of a thousand cubes falling in to a funnel takes 9.5 hours. Kaufman et al. [KEP05] present a decoupled projection scheme, computing friction and normal forces as two sets of projections onto convex sets. The method in [KEP05] is also aimed at large scale computer animations. Although no frame rates are reported, the method is claimed to be linear in the total number of contact points. In [Erl07] a constraint–based velocity formulation for large scale simulations is presented, using a shock propagation method for handling penetrations. The method is compared with other large scale simulation methods, not including the method from [KEP05]. The method is not included in the comparison because Erleben claims it is not shown to work for dense structured stacks. Realizing that the decoupling of normal and friction forces can cause instability, Kaufman et al. present a revised and coupled projection scheme in [KSJP08]. The method is claimed interactive enough for use in haptic simulations.

1.4 Outline

Some topics are beyond the scope of this thesis, and are therefore assumed known by the reader. The following is a list of such topics and references to literature where the topics are covered more extensively:

- Rigid body simulation [ESHD05]
- Analytical mechanics [Lan70]
- Numerical optimization [NW99]
- Nonsmooth analysis [Cla90]
- Complementarity theory [Mur88, Bil95, NZ95]

For a short overview of the thesis, the following is a list of compact section summaries.

Section 1 A short introduction to multibody dynamics, motivating the work presented in this thesis. An overview of the relatively short history of the field is presented.

Section 2 An introduction of the math needed to understand the remainder of the thesis.

Section 3 The general equations of motions are outlined, along with three different models of the contact force determination problem.

Section 4 This section contains a summary, elaborations and conclusions of the paper submitted to the 2009 VRIPHYS workshop. It is recommended to read the article in Appendix A before reading this section.

Section 5 The model outlined in Section 3.3.3 is more thoroughly derived, and a method for solving such a model is proposed.

Section 6 A gathering of loose ends, trying to make conclusive statements based on the presented work. Possible routes of future work is laid out.

Appendix A The reviewer comments and the accepted VRIPHYS paper are included in this appendix.

Appendix B A paper I have co-authored and submitted to the WSCG 2010 conference on Computer Graphics, Visualization and Computer Vision. It is a survey of heuristics for improving convergence of the projected Gauss-Seidel. Acceptance decision is expected in early December 2009.

Appendix C A paper I have co-authored and submitted to the GRAPP 2010 International Conference on Computer Graphics Theory and Applications. The paper presents an extension of the projected Gauss–Seidel method, using a subspace minimization approach. Acceptance decision is expected in late January 2010.

The notation is kept simple, and should be clear from context. However, having read several articles on the same subject I can conclude that notation is neither uniform nor as simple as the authors seem to think. Therefore, I include this short list of notation.

- $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{\gamma}$ bold lowercase letters are vectors.
- $\boldsymbol{A}, \boldsymbol{M}, \boldsymbol{\Lambda}$ bold uppercase letters are matrices.
- $\boldsymbol{x}_i$ subscripting is used as indexing.

- $\boldsymbol{A}_{ij}$ the (i, j)'th component of the matrix.
- $\boldsymbol{x}^k$ superscripting is used for iteration count.
- $\text{co}\{S\}$ the convex hull of the set S.
- $||\cdot||$ the Euclidean norm.
- $||\cdot||_C$ the C norm, such that $||\boldsymbol{x}||_C = \sqrt{\boldsymbol{x}^T \boldsymbol{C} \boldsymbol{x}}$.
- $\boldsymbol{y} = \boldsymbol{x}$, $\boldsymbol{y} \geq \boldsymbol{x}$ equality and inequality between vectors are element wise.
- $\boldsymbol{x}^+, \boldsymbol{x}^-$ vectors of same dimension as $\boldsymbol{x}$, constructed such that $\boldsymbol{x}^+ \geq 0$, $\boldsymbol{x}^- \geq 0$ and $\boldsymbol{x}^+ - \boldsymbol{x}^- = \boldsymbol{x}$.
- $\mathbb{R}^n_+$ The nonnegative subset of $\mathbb{R}^n$.

2 Mathematical Prerequisites

This section covers some of the mathematical concepts I will use in following sections. This has been given its own section, so the sections where the concepts are used can be focused on application rather than general theory.

2.1 Clarke's Generalized Jacobian

The method chosen to solve the contact force problem is a Newton–based method. The Newton method is characterized by the solution of the Newton system,

$$\nabla f(\boldsymbol{x})\Delta x = -f(\boldsymbol{x}), \tag{1}$$

where $\nabla f(\boldsymbol{x})$ is the gradient of $f(\boldsymbol{x})$. For this to be a well-defined problem, $\nabla f(\boldsymbol{x})$ has to be defined everywhere $f(\boldsymbol{x})$ is defined. If $f(\boldsymbol{x})$ is not everywhere differentiable, an alternative to $\nabla f(\boldsymbol{x})$ is required. Clarke [Cla90] gives the following definition,

Definition 2.1 (Clarke's generalized Jacobian) *Let Ω_f be the set of points where f is not differentiable, then Clarke's generalized Jacobian of f at $\boldsymbol{x} \in \Omega_f$, $\delta f(\boldsymbol{x})$, is defined as,*

$$\delta f(\boldsymbol{x}) = co\{\lim_{\boldsymbol{x}' \to \boldsymbol{x}} \nabla f(\boldsymbol{x}') \,|\, \boldsymbol{x}' \notin \Omega_f\}. \tag{2}$$

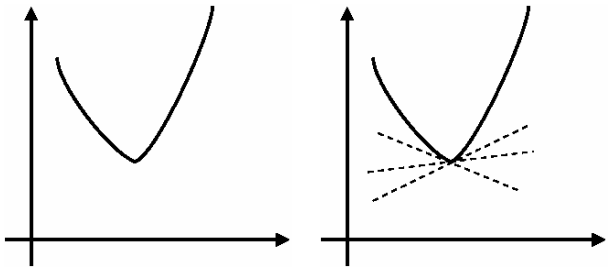
Clarke's generalized Jacobian is a generalization of the subdifferential. The subdifferential is a concept from convex analysis and is the set of all lines going through the non-differentiable point while remaining below – hence the name subdifferential – or on the curve as illustrated by Figure 3. When $f(\boldsymbol{x})$ is everywhere differentiable, $\delta f(\boldsymbol{x})$ is equivalent to $\nabla f(\boldsymbol{x})$. Using (2), the generalized Newton system is,

$$\boldsymbol{H}_{f(\boldsymbol{x}_k)} f(\boldsymbol{x})\Delta \boldsymbol{x} = -f(\boldsymbol{x}), \quad \boldsymbol{H}_{f(\boldsymbol{x}_k)} \in \delta f(\boldsymbol{x}), \tag{3}$$

thus making the Newton method applicable for a wider set of functions.

2.2 Complementarity Problems

The chosen simulation paradigm uses complementarity problems to model the contact force problem. In a single dimension, the complementarity prob-



(a) (b)

Figure 3: *Visualization of a subdifferential. Figure (a) shows a function that is not everywhere differentiable. Figure (b) shows (a subset of) the subdifferential at the non-differentiable point.*

lem(CP) is to determine x and y such that,

$$x \geq 0, \quad y \geq 0, \quad xy = 0. \tag{4}$$

In the n–dimensional case, the nonlinear complementarity problem(NCP) is to determine $\boldsymbol{x} \in \mathbb{R}^n$ such that,

$$\boldsymbol{x} \geq 0, \quad f(\boldsymbol{x}) \geq 0, \quad \boldsymbol{x}^T f(\boldsymbol{x}) = 0, \tag{5}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$. When f is an affine mapping, $f(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$, the NCP (5) is reduced to a linear complementarity problem(LCP). Determine $\boldsymbol{x}$ such that,

$$\boldsymbol{x} \geq 0, \quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} \geq 0, \quad \boldsymbol{x}^T(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}) = 0. \tag{6}$$

Solving LCPs is in general NP hard, however it can be shown that some LCPs can be solved in polynomial time. Consider a LCP where the matrix $\boldsymbol{A}$ can be written as the product $\boldsymbol{R}\boldsymbol{D}\boldsymbol{R}^T$, such that $\boldsymbol{R}$ is an orthogonal matrix and $\boldsymbol{D}$ is a diagonal matrix. By substitution, (6) can be written as,

$$\boldsymbol{x} \geq 0, \quad (\boldsymbol{R}\boldsymbol{D}\boldsymbol{R}^T)\boldsymbol{x} + \boldsymbol{b} \geq 0, \quad \boldsymbol{x}^T((\boldsymbol{R}\boldsymbol{D}\boldsymbol{R}^T)\boldsymbol{x} + \boldsymbol{b}) = 0. \tag{7}$$

Define $\boldsymbol{z} = \boldsymbol{R}^T\boldsymbol{x}$ so that,

$$\boldsymbol{x} \geq 0, \quad (\boldsymbol{R}\boldsymbol{D})\boldsymbol{z} + \boldsymbol{b} \geq 0, \quad \boldsymbol{x}^T((\boldsymbol{R}\boldsymbol{D})\boldsymbol{z} + \boldsymbol{b}) = 0. \tag{8}$$

Pre-multiply the two inequalities by $\boldsymbol{R}^T$ such that,

$$\boldsymbol{R}^T\boldsymbol{x} \geq \boldsymbol{R}^T 0, \quad \boldsymbol{R}^T(\boldsymbol{R}\boldsymbol{D}\boldsymbol{z} + \boldsymbol{b}) \geq \boldsymbol{R}^T 0, \quad (\boldsymbol{R}^T\boldsymbol{x})^T(\boldsymbol{R}^T(\boldsymbol{R}\boldsymbol{D}\boldsymbol{z} + \boldsymbol{b})) = 0. \tag{9}$$

Substitute for $\boldsymbol{z}$ and utilize that $\boldsymbol{R}$ is an orthogonal matrix such that $\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{R}^{-1}\boldsymbol{R} = \boldsymbol{I}$,

$$\boldsymbol{z} \geq 0, \quad \boldsymbol{D}\boldsymbol{z} + \boldsymbol{R}^T\boldsymbol{b} \geq 0, \quad \boldsymbol{z}^T(\boldsymbol{D}\boldsymbol{z} + \boldsymbol{R}^T\boldsymbol{b}) = 0. \tag{10}$$

Note that $\boldsymbol{R}^T\boldsymbol{b}$ is simply an orthogonal transformation of $\boldsymbol{b}$. Let $\boldsymbol{c} = \boldsymbol{R}^T\boldsymbol{b}$,

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} \begin{bmatrix} D_{11} & 0 & \cdots & 0 \\ 0 & D_{22} & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & D_{nn} \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} \geq 0. \tag{11}$$

Now the entire LCP has been orthogonally transformed, resulting in n one–dimensional problems rather than one n–dimensional problem. The problem is reduced to finding the n z_i's that solves,

$$z_i(z_i D_{ii} + c_i) = 0. \tag{12}$$

If $\boldsymbol{A}$ is symmetric, then an eigenvalue decomposition of $\boldsymbol{A}$ yields an orthogonal and a diagonal matrix. Eigenvalue decomposition can be performed in $\mathcal{O}(n^3)$ [TB97], and the solution of the transformed problem amounts to a linear run through the n equations setting,

$$z_i = \begin{cases} 0 & \text{if } c_i \geq 0 \\ -\frac{c_i}{D_{ii}} & \text{if } c_i < 0 \end{cases} \tag{13}$$

Thus, a subclass of LCPs can be solved in polynomial (cubic) time. When $\boldsymbol{A}$ is symmetric and positive definite, the eigenvalue decomposition ensures that $D_{ii} > 0 \, \forall i$, so for this class of matrices at least, solving LCPs can be done in polynomial time.

The mixed linear complementarity problem(MLCP) is a generalization of the NCP, determine $\boldsymbol{x}$ such that,

$$(\boldsymbol{x} - \boldsymbol{l})^T f(\boldsymbol{x})^+ = 0 \quad \wedge \quad (\boldsymbol{u} - \boldsymbol{x})^T f(\boldsymbol{x})^- = 0, \tag{14}$$

where $\boldsymbol{l}$ and $\boldsymbol{u}$ represents lower and upper bounds of $\boldsymbol{x}$ respectively.

2.3 Proximal Mapping

In Section 3.3.3, I present a contact force model, using a proximal mapping notation. The proximal mapping of a point to a convex set K is the closest point in K,

$$\text{prox}_K^C(\boldsymbol{x}) = \min_{\boldsymbol{z}\in K} ||\boldsymbol{x}-\boldsymbol{z}||_C, \quad \boldsymbol{x}\in\mathbb{R}^n, \tag{15}$$

for some scalar $r > 0$. When C is omitted, the Euclidean norm is used. The proximal mapping

$$\boldsymbol{x} = \text{prox}_{\mathbb{R}^n_+}^C(\boldsymbol{x} - r(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b})),$$

is equivalent to the LCP,

$$\boldsymbol{x}\geq 0, \quad \boldsymbol{A}\boldsymbol{x}+\boldsymbol{b}\geq 0, \quad \boldsymbol{x}^T(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b}) = 0,$$

when $\boldsymbol{A}$ is symmetric. To realize this, first note that the norm function is nonnegative and that any minimizer of $||\boldsymbol{x}-\boldsymbol{z}||$ is also the minimizer of $\frac{1}{2}||\boldsymbol{x}-\boldsymbol{z}||^2$. Using this, let $K = \mathbb{R}^n_+$,

$$\begin{aligned} \boldsymbol{x}^* = \text{prox}_K^C(\boldsymbol{x}) &= \min_{\boldsymbol{x}\in K} ||\boldsymbol{x}-\boldsymbol{z}||_C, && (16a)\\ &= \min_{\boldsymbol{x}\in K} \frac{1}{2}||\boldsymbol{x}-\boldsymbol{z}||_C^2. && (16b)\end{aligned}$$

Define $\boldsymbol{z} = \boldsymbol{x} - r(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b})$ and $C = \boldsymbol{A}^{-1}$,

$$\begin{aligned} \boldsymbol{x}^* &= \min_{\boldsymbol{x}\in K} \frac{1}{2}||\boldsymbol{x}-(\boldsymbol{x}-r(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b}))||_C^2, && (17a)\\ &= \min_{\boldsymbol{x}\in K} \frac{1}{2} r^2(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b})^T\boldsymbol{C}(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b}), && (17b)\\ &= \min_{\boldsymbol{x}\in K} \frac{1}{2}(\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{C}\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{C}\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{C}\boldsymbol{b} + \boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{C}\boldsymbol{b}), && (17c)\\ &= \min_{\boldsymbol{x}\in K} \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x}. && (17d)\end{aligned}$$

The solution of the resulting quadratic problem(QP) (17d) is given by the KKT optimality conditions,

$$\begin{aligned} x^T\boldsymbol{A} + \boldsymbol{b}^T &= \boldsymbol{\lambda}, && (18a)\\ \boldsymbol{x} &\geq 0, && (18b)\\ \boldsymbol{\lambda} &\geq 0, && (18c)\\ \boldsymbol{x}^T\boldsymbol{\lambda} &= 0, && (18d)\end{aligned}$$

where $\boldsymbol{\lambda}$ is a vector of Lagrange multipliers. So, the solution to the LCP given by the KKT conditions (18) is equivalent to the solution of the QP (17d). Having already shown that the solution to $\text{prox}_K^C(\boldsymbol{x} - r(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}))$ is equivalent to the solution of QP (17d), it follows that the solution to the proximal mapping is also equivalent to the solution of the LCP.

3 Contact Models

In rigid body simulation, contact forces are applied to prevent rigid bodies from penetrating each other. The fidelity of the simulation is highly affected by the accuracy of the computed contact forces. To achieve physical plausibility, frictional forces are essential. The formulation of a frictional contact force problem includes the modeling of both normal force constraints and friction force constraints. Henceforth, the frictional contact force problem will simply be referred to as the contact force problem. Before deriving the contact model, it should be noted that the simulation system is constraint–based using a velocity–based position update [Erl07].

As this is a thesis on contact problems, and not collision detection, the following models are given under the assumption that contact determination has already been performed.

3.1 Equations of Motion

The main purpose of rigid body simulation is to determine the position of a body at time $t + 1$, given some physical interaction at time t. To do this, a way of model of the physical interactions is needed. There are two main approaches to such a model, the vectorial and the variational approach [Lan70]. The vectorial approach is best known as Newton's laws of motion, describing motion as a relation between momentum and force. Euler and Lagrange introduced the variational approach, using kinetic and potential energy to model the physical interactions of a system.

Newton's model describes the effect experienced by a single body, where the Euler-Lagrange model describes the effect of an entire system simultaneously. When using Newtonian mechanics, the choice of an appropriate coordinate system is of great importance. The Lagrange equations used in the Euler-Lagrange approach are invariant under coordinate changes, so the choice of coordinate system is irrelevant.

The Lagrangian equation for a system of bodies is defined as,

$$\mathcal{L}(\boldsymbol{q}, \boldsymbol{v}) = \frac{1}{2}\boldsymbol{v}^T \boldsymbol{M}(\boldsymbol{q})\boldsymbol{v} - \boldsymbol{V}(\boldsymbol{q}), \tag{19}$$

where $\boldsymbol{v}$ is the vector of generalized velocities and $\boldsymbol{q}$ is the generalized position

vector. The total kinetic energy for all bodies in the system is $\frac{1}{2}\boldsymbol{v}^T\boldsymbol{M}(\boldsymbol{q})\boldsymbol{v}$, $\boldsymbol{V}(\boldsymbol{q})$ is the generalized potential energy. The matrix $\boldsymbol{M}(\boldsymbol{q})$ is the generalized mass matrix. The action integral of (19) is,

$$\mathcal{A} = \int_a^b \mathcal{L}\, dt. \tag{20}$$

Hamilton's principle states that

> *the motion of an arbitrary mechanical system occurs in such a way that the definite integral $\mathcal{A}$ becomes stationary for arbitrary possible variations of the configurations of the system, provided the initial and final configurations of the system are prescribed* [Lan70].

In short, this means that,

$$\delta\mathcal{A} = 0. \tag{21}$$

The necessary and sufficient condition for (20) to be stationary, is the differential equation,

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{q}} - \frac{d}{dt}\frac{\partial\mathcal{L}}{\partial\dot{\boldsymbol{q}}} = 0. \tag{22}$$

This is known as the Euler-Lagrange equation, where $\frac{\partial\mathcal{L}}{\partial\boldsymbol{q}} = -\nabla\boldsymbol{V}(\boldsymbol{q})$ and $\frac{\partial\mathcal{L}}{\partial\dot{\boldsymbol{q}}} = \boldsymbol{M}(\boldsymbol{q})\frac{d\boldsymbol{v}}{dt}$. For free moving non-colliding objects, the Newton-Euler equations are,

$$\boldsymbol{M}(\boldsymbol{q})\frac{d\boldsymbol{v}}{dt} = \nabla\boldsymbol{V}(\boldsymbol{q}) + k(\boldsymbol{q},\boldsymbol{v}), \tag{23a}$$

$$\frac{d\boldsymbol{q}}{dt} = \boldsymbol{v}. \tag{23b}$$

They are also known as the equations of motion. The linear mapping $k(\boldsymbol{q},\boldsymbol{v})$ is the non-quadratic contribution of kinetic energy, describing the gyroscopic forces.

When objects collide, their interactions are modeled using contact forces. The contact forces consist of both normal forces and frictional forces, and are subject to a set of constraints. Without specifying the individual constraints, I define the set of constraint functions,

$$\boldsymbol{c}(\boldsymbol{q}) = [c_1(\boldsymbol{q}), c_2(\boldsymbol{q}), \ldots, c_n(\boldsymbol{q})], \tag{24}$$

the Jacobian, $\boldsymbol{J}_c$, of (24) is,

$$\boldsymbol{J}_c = \begin{bmatrix} \frac{\partial c_1}{\partial q_1} & \frac{\partial c_2}{\partial q_1} & \cdots & \frac{\partial c_n}{\partial q_1} \\ \frac{\partial c_1}{\partial q_2} & \frac{\partial c_2}{\partial q_2} & \cdots & \frac{\partial c_n}{\partial q_2} \\ \vdots & \vdots & & \vdots \\ \frac{\partial c_1}{\partial q_m} & \frac{\partial c_2}{\partial q_m} & \cdots & \frac{\partial c_n}{\partial q_m} \end{bmatrix}. \tag{25}$$

The constraints are added to (20) by applying the Lagrange multiplier method such that,

$$\boldsymbol{M}(\boldsymbol{q})\frac{d\boldsymbol{v}}{dt} = \nabla \boldsymbol{V}(\boldsymbol{q}) + k(\boldsymbol{q}, \boldsymbol{v}) + \boldsymbol{J}_c^T \boldsymbol{\lambda}, \tag{26}$$

$$\frac{d\boldsymbol{q}}{dt} = \boldsymbol{v}, \tag{27}$$

where $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_n]$ are the Lagrange multipliers.

3.2 Modeling Friction

The contact models derived in the following sections all use Coulomb's friction law as a basis for constraining the frictional forces. The models use different approximations to Coulomb's friction law. Coulomb's friction law for isotropic planar friction can be stated as a cone,

$$\mathrm{F}_C = \{\boldsymbol{c}_t + c_n\boldsymbol{n} \;\mid\; ||\boldsymbol{c}_t|| \leq \mu c_n,\, \boldsymbol{c}_t \perp \boldsymbol{n}\}. \tag{28}$$

In the case of dynamic friction, the resulting friction force will lie on the boundary of the friction cone and oppositely directed to the relative contact velocity. If the friction is static, any friction force $\boldsymbol{c}_t \in \mathrm{F}_C$ is valid. Figure 4 is a geometric visualization of Coulomb's friction cone.

3.3 Contact Forces

For the models presented in this thesis, a contact force is defined by both a normal component and a frictional component. The magnitude of the normal component is a scalar, λ_n. The dimension of the frictional component is determined by the friction model used.

Without loss of generality, I shall only consider a single contact point.

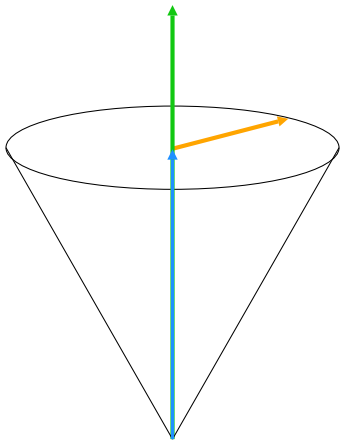


Figure 4: *Visualization of an isotropic planar friction cone, as defined by Coulomb's friction law (28). The green vector is the contact normal* $\boldsymbol{n}$*, the blue vector is the normal force* $c_n\boldsymbol{n}$ *and the orange vector is a valid friction force with maximal magnitude. The cone is the set of all valid friction forces.*

3.3.1 LCP - The Classical Approach

Stewart and Trinkle(ST) [ST96] derived a LCP formulation of the contact force problem, which is cited, used and improved upon in large parts of literature on complementarity formulations of the contact problem [AP97, KSJP08]. Their model can approximate Coulomb's friction model with great accuracy. However, to achieve great accuracy, large systems of variables are required. Solving large systems is infeasible if interactivity is desired. Still, the ST model serves as a nice basis for the derivation of the model used in the two popular open source simulators, Open Dynamics Engine(ODE) and Bullet. This model is derived in Section 3.3.2.

The basic idea of the ST model is a linearization of Coulomb's friction law, the polyhedral friction cone approximation,

$$\mathrm{F}_{ST} = \{\boldsymbol{J}_D^T\boldsymbol{\lambda}_D + \boldsymbol{J}_n^T\lambda_n \;\mid\; \mu\lambda_n - \boldsymbol{e}^T\boldsymbol{\lambda}_D \geq 0,\, \boldsymbol{\lambda}_D \geq 0\}. \tag{29}$$

The vector $\boldsymbol{e}$ is a vector of ones, it has the same dimension as $\boldsymbol{\lambda}_D$. The rows of matrix the $\boldsymbol{J}_D$ are direction vectors $\boldsymbol{d}_j$ spanning a subspace of (28). For every i there is a j such that,

$$\boldsymbol{d}_i = -\boldsymbol{d}_j. \tag{30}$$

The direction vectors $\boldsymbol{d}_j$ are not required to be of unit length when modeling anisotropic friction. Figure 5 illustrates a polyhedral friction cone approximation using 8 facets. The accuracy of the model can be controlled by the number of direction vectors used. However, increasing the number of facets of the polyhedral, increases the size of the resulting contact force problem. There is a clear tradeoff between friction accuracy and problem size, which ultimately will be a tradeoff between friction accuracy and speed. Different friction force magnitude vectors $\boldsymbol{\lambda}_D$ may yield the same friction force $\boldsymbol{J}_D^T\boldsymbol{\lambda}_D$. To ensure unique solution of $\boldsymbol{\lambda}_D$, provided that $\boldsymbol{J}_D\boldsymbol{v} \neq 0$, extra conditions are needed.

Consider the equations of motion, where normal and frictional contributions are separated, and all external forces are agglomerated in $\boldsymbol{F}$,

$$\boldsymbol{M}(\boldsymbol{q})\frac{d\boldsymbol{v}}{dt} = \underbrace{\boldsymbol{J}_n^T\lambda_n + \boldsymbol{J}_D^T\boldsymbol{\lambda}_D}_{\text{total contact force}} + \boldsymbol{F}, \tag{31a}$$

$$\frac{d\boldsymbol{q}}{dt} = \boldsymbol{v}. \tag{31b}$$

The normal force ensures that no penetrations occur, this is modeled by the non-penetration constraints,

$$\boldsymbol{J}_n\boldsymbol{v} \geq 0, \tag{32a}$$

$$\lambda_n \geq 0, \tag{32b}$$

$$(\boldsymbol{J}_n\boldsymbol{v})\lambda_n = 0. \tag{32c}$$

The friction force is required to lie within or on the boundary of the friction cone approximation,

$$\mu\lambda_n - \boldsymbol{e}^T\boldsymbol{\lambda}_D \geq 0, \tag{33}$$

where $\boldsymbol{\lambda}_D \geq 0$. We let the scalar β be a measure of the most negative contact velocity, such that,

$$\beta\boldsymbol{e} + \boldsymbol{J}_D\boldsymbol{v} \geq 0, \tag{34a}$$

$$\beta \geq 0. \tag{34b}$$

The relative tangential contact velocity is coupled with the friction force by the following complementarity conditions,

$$(\beta\boldsymbol{e} + \boldsymbol{J}_D\boldsymbol{v})^T\boldsymbol{\lambda}_D = 0, \tag{35a}$$

$$(\mu\lambda_n + \boldsymbol{e}^T\boldsymbol{\lambda}_D)\beta = 0. \tag{35b}$$

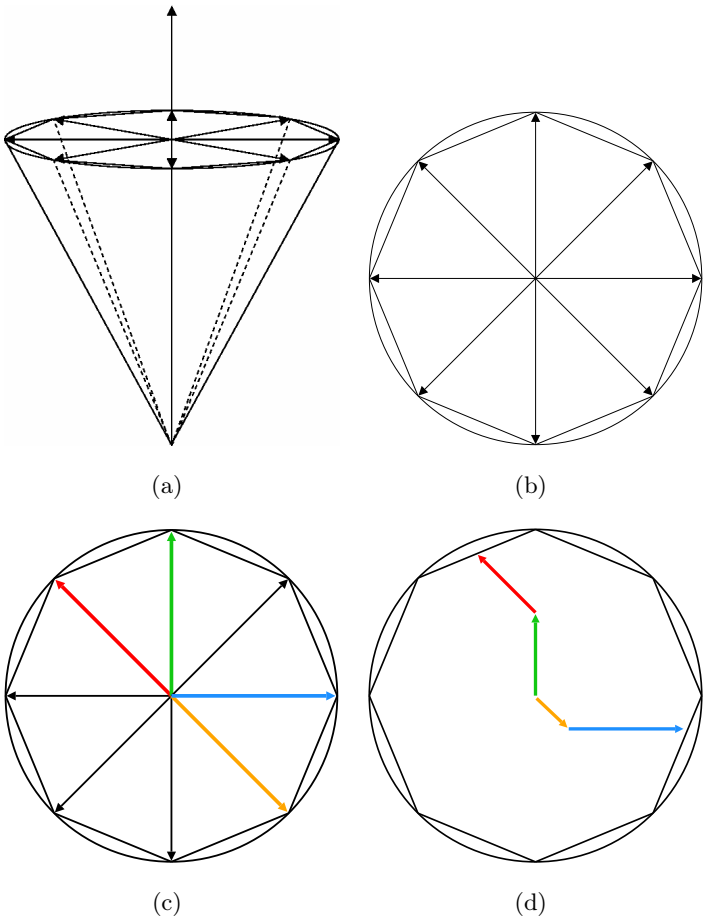


Figure 5: *A geometric illustration of the polyhedral friction cone approximation, presented by Stewart and Trinkle [ST96] (b) Tilted side view (b) Top view. The open end radius is given by $\mu\lambda_n$. To geometrically verify that equation (29) does indeed describe the inscribed polygon of the friction cone, consider the following example (c) Four rows of $\boldsymbol{J}_D$ are explicitly colored (d) Two valid combinations, given by $\boldsymbol{\lambda}_D^1 = [\frac{1}{2}\ \frac{1}{2}\, 0\, 0\, 0\, 0\, 0\, 0]^T$ and $\boldsymbol{\lambda}_D^2 = [0\, 0\, 0\, 0\, 0\, 0\, \frac{1}{4}\ \frac{3}{4}]^T$*

These last two complementarity conditions ensure that the resulting friction force satisfies the principle of maximum dissipation. The complete contact force determination problem is then to compute $\boldsymbol{q}, \boldsymbol{v}, \boldsymbol{\lambda}_D, \lambda_n$and β such that,

$$\begin{aligned}
\boldsymbol{M}(\boldsymbol{q})\frac{d\boldsymbol{v}}{dt} &= \boldsymbol{J}_n^T\lambda_n + \boldsymbol{J}_D^T\boldsymbol{\lambda}_D + \boldsymbol{F}, && \text{(36a)}\\
\frac{d\boldsymbol{q}}{dt} &= \boldsymbol{v}, && \text{(36b)}\\
\beta\boldsymbol{e} + \boldsymbol{J}_D\boldsymbol{v} &\geq 0, && \text{(36c)}\\
\boldsymbol{J}_n\boldsymbol{v} &\geq 0, && \text{(36d)}\\
\mu\lambda_n - \boldsymbol{e}^T\boldsymbol{\lambda}_D &\geq 0, && \text{(36e)}\\
\boldsymbol{\lambda}_D, \lambda_n, \beta &\geq 0, && \text{(36f)}\\
(\beta\boldsymbol{e} + \boldsymbol{J}_D\boldsymbol{v})^T\boldsymbol{\lambda}_D &= 0, && \text{(36g)}\\
(\boldsymbol{J}_n\boldsymbol{v})\lambda_n &= 0, && \text{(36h)}\\
(\mu\lambda_n - \boldsymbol{e}^T\boldsymbol{\lambda}_D)\beta &= 0. && \text{(36i)}
\end{aligned}$$

By discretization of $\frac{d\boldsymbol{v}}{dt}$ in (31), we can isolate $\boldsymbol{v}^{l+1}$,

$$\boldsymbol{v}^{l+1} = \boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{J}_n^T\lambda_n' + \boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{J}_D^T\boldsymbol{\lambda}_D' + \boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{F}', \tag{37}$$

where $\boldsymbol{F}'$, λ_n' and $\boldsymbol{\lambda}_n'$ are abstractions of the discretization variables. Using (31) and introducing the shorthand $\boldsymbol{W} = \boldsymbol{M}(\boldsymbol{q})^{-1}$, the contact force problem can be stated in matrix-vector form,

$$\underbrace{\begin{bmatrix} \boldsymbol{J}_n\boldsymbol{W}\boldsymbol{J}_n^T & \boldsymbol{J}_n\boldsymbol{W}\boldsymbol{J}_D^T & 0 \\ \boldsymbol{J}_D\boldsymbol{W}\boldsymbol{J}_n^T & \boldsymbol{J}_D\boldsymbol{W}\boldsymbol{J}_D^T & \boldsymbol{e} \\ \mu & -\boldsymbol{e}^T & 0 \end{bmatrix}}_{\boldsymbol{A}_{\text{LCP}}} \underbrace{\begin{bmatrix} \lambda_n\prime \\ \boldsymbol{\lambda}_D\prime \\ \beta \end{bmatrix}}_{\boldsymbol{x}_{\text{LCP}}} + \underbrace{\begin{bmatrix} \boldsymbol{J}_n\boldsymbol{W}\boldsymbol{F}\prime \\ \boldsymbol{J}_D\boldsymbol{W}\boldsymbol{F}\prime \\ 0 \end{bmatrix}}_{\boldsymbol{b}_{\text{LCP}}} \geq 0, \tag{38a}$$

$$\boldsymbol{x}_{\text{LCP}} \geq 0, \tag{38b}$$

$$\boldsymbol{x}_{\text{LCP}}^T(\boldsymbol{A}_{\text{LCP}}\boldsymbol{x}_{\text{LCP}} + \boldsymbol{b}_{\text{LCP}}) = 0. \tag{38c}$$

For a single contact point, the dimension of the LCP is $\boldsymbol{A}_{\text{LCP}} \in \mathbb{R}^{(2+k)\times(2+k)}$ where k is the number of facets in the friction polyhedral approximation. Even with a simple four-sided pyramid approximation to the friction cone, a contact force problem for n contact points would be $\boldsymbol{A}_{\text{LCP}} \in \mathbb{R}^{(6n)\times(6n)}$.

3.3.2 NCP - The Interactive Approach

As mentioned, the LCP presented in (37) and (38) is not feasible for real time or interactive simulations. Instead, an alternative friction model has been developed which reduces the problem size significantly. As in (37), the generalized velocities are,

$$\boldsymbol{v}^{l+1} = \boldsymbol{W}\boldsymbol{J}_n^T\lambda_n' + \boldsymbol{W}\boldsymbol{J}_t^T\boldsymbol{\lambda}_t' + \boldsymbol{W}\boldsymbol{F}', \tag{39}$$

but instead of $\boldsymbol{J}_D$, we now use $\boldsymbol{J}_t$ to span the frictional plane. By defining $\boldsymbol{\lambda} = [\lambda_n' \;\; \boldsymbol{\lambda}_t'^T]^T$ and the contact Jacobian $\boldsymbol{J} = [\boldsymbol{J}_n \;\; \boldsymbol{J}_t]^T$ we can restate (39) as,

$$\boldsymbol{v}^{l+1} = \boldsymbol{W}\boldsymbol{J}^T\boldsymbol{\lambda} + \boldsymbol{W}\boldsymbol{F}', \tag{40}$$

where $\boldsymbol{v}^{l+1}$ is the vector of generalized velocities. By pre-multiplying the generalized velocities with the contact Jacobian, $\boldsymbol{J}$, we obtain the relative contact velocity vector $\boldsymbol{y}$,

$$\boldsymbol{y} = \boldsymbol{J}\boldsymbol{v}^{l+1} = \underbrace{\boldsymbol{J}\boldsymbol{W}\boldsymbol{J}^T}_{A_{\text{NCP}}}\boldsymbol{\lambda} + \underbrace{\boldsymbol{J}\boldsymbol{W}\boldsymbol{F}'}_{b_{\text{NCP}}}. \tag{41}$$

For simplicity, let us first consider Coulomb's friction law in the one–dimensional case. The relative velocity in the one–dimensional case can fall in to one of two groups,

- $y \neq 0$: dynamic friction.
- $y = 0$: static friction.

Let us for convention decide that $y < 0$ is left sliding, meaning that $y > 0$ is right sliding. In case of left sliding, the friction force will be on the right edge of the corresponding friction cone, and vice versa for right sliding. In case of static friction, the friction can be anywhere in the friction cone including the boundary. This can be written as [Bar94],

$$y < 0 \;\; \Rightarrow \;\; \lambda_t = \mu\lambda_n, \tag{42a}$$

$$y > 0 \;\; \Rightarrow \;\; \lambda_t = -\mu\lambda_n, \tag{42b}$$

$$y = 0 \;\; \Rightarrow \;\; -\mu\lambda_n \leq \lambda_t \leq \mu\lambda_n. \tag{42c}$$

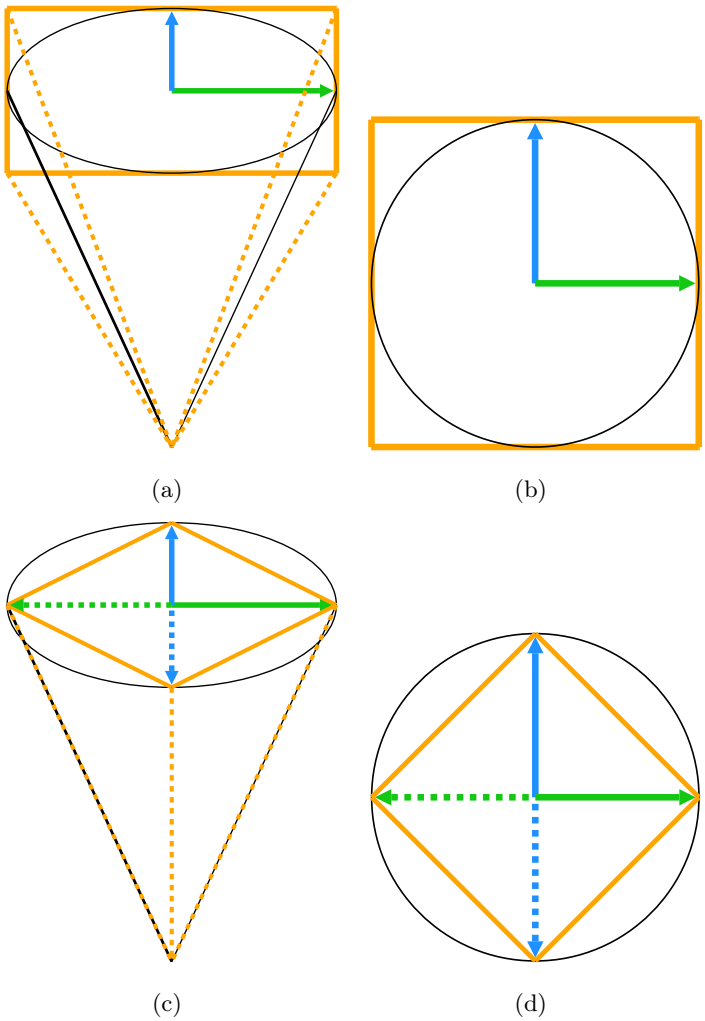


Figure 6: *A geometric illustration of the two presented friction cone approximations. (a) and (b) show the outer pyramid, the result of the NCP friction model where the two friction directions (blue and green vectors) are decoupled. (c) and (d) show the LCP using the same amount of facets, resulting in an inner pyramid.*

We span the frictional plane by two orthogonal direction vectors of unit length, using the one–dimensional friction model to bound the frictional contribution in the two directions. The matrix $\boldsymbol{J}_t$ is constructed from these two direction vectors. To formalize the bounds, we define the upper and lower bounds,

$$\begin{aligned} u_n(\boldsymbol{\lambda}) &= \infty, && (43a)\\ l_n(\boldsymbol{\lambda}) &= 0, && (43b)\\ u_t(\boldsymbol{\lambda}) &= \mu\lambda_n, && (43c)\\ l_t(\boldsymbol{\lambda}) &= -\mu\lambda_n. && (43d) \end{aligned}$$

For the full contact problem for a single contact point, the relative contact velocity is a three–dimensional vector with one normal contribution and two frictional contributions. To couple the normal and frictional contributions we first split $\boldsymbol{y}$ into its positive and negative components such that,

$$\boldsymbol{y} = \boldsymbol{y}^+ - \boldsymbol{y}^-, \tag{44}$$

where

$$\boldsymbol{y}^+ \geq 0, \quad \boldsymbol{y}^- \geq 0, \quad \left(\boldsymbol{y}^+\right)^T \left(\boldsymbol{y}^-\right) = 0. \tag{45}$$

Using the bounds and the splitting of $\boldsymbol{y}$, the contact force problem can be stated as the following NCP,

$$\begin{aligned} \boldsymbol{y}^+ - \boldsymbol{y}^- &= \boldsymbol{A}_{\mathrm{NCP}}\boldsymbol{\lambda} + \boldsymbol{b}_{\mathrm{NCP}}, && (46a)\\ \boldsymbol{y}^+ &\geq 0, && (46b)\\ \boldsymbol{y}^- &\geq 0, && (46c)\\ \boldsymbol{u}(\boldsymbol{\lambda}) - \boldsymbol{\lambda} &\geq 0, && (46d)\\ \boldsymbol{\lambda} - \boldsymbol{l}(\boldsymbol{\lambda}) &\geq 0, && (46e)\\ (\boldsymbol{u}(\boldsymbol{\lambda}) - \boldsymbol{\lambda})^T \boldsymbol{y}^- &= 0, && (46f)\\ (\boldsymbol{\lambda} - \boldsymbol{l}(\boldsymbol{\lambda}))^T \boldsymbol{y}^+ &= 0, && (46g)\\ \left(\boldsymbol{y}^+\right)^T \left(\boldsymbol{y}^-\right) &= 0. && (46h) \end{aligned}$$

By now, the auxiliary variable β is eliminated and the friction plane is spanned using only two orthogonal direction vectors. This friction model

reduces the memory footprint drastically, making it more applicable for interactive simulators. For a single contact point the problem to solve is only $\boldsymbol{A}_{\text{NCP}} \in \mathbb{R}^{3\times3}$, a quarter of the size of a four sided LCP model. The main problem with this model is the decoupling of the two frictional directions. This decoupling amounts to solving two one–dimensional friction problems and tends to overshoot the resulting friction force. Figure 6 is a visual comparison of the two pyramidal friction cone approximations given by the LCP and NCP formulations respectively.

3.3.3 Proximal Mapping - The Elegant Approach

Solution methods for solving NCPs are often based on some form of reformulation of the NCP. Often these reformulations are based on NCP functions, merit functions, smoothing functions, interior points methods and projection methods [FK98]. In [SNE09] we use the NCP function known in literature as the Fischer function, for reformulating the contact force problem. The reformulated problem is solved using a Newton–based method. The reformulation we use in [PNE10] is based on the NCP function called the minimum function, which we solve using the projected Gauss–Seidel method.

The contact model presented here is a variation of the projection method, inspired by the augmented Lagrangian approach in [AC91] and the staggered approach in [KSJP08].

Let us restate the non-penetration constraints (32),

$$\lambda_n \geq 0, \quad v_n \geq 0, \quad \lambda_n v_n = 0, \tag{47}$$

where $v_n = \boldsymbol{J}_n v$ is the relative contact velocity in the normal direction. This is equivalent to the proximal mapping,

$$\lambda_n = \text{prox}_{\mathbb{R}_+}(\lambda_n - r_n v_n), \quad r_n > 0. \tag{48}$$

The proof of this equivalency claim is given in Section 5. To determine the frictional magnitudes $\boldsymbol{\lambda}_D$ we use Coulomb's friction law, which states that when the relative tangential contact velocity, $||\boldsymbol{v}_D|| = ||\boldsymbol{J}_D^T \boldsymbol{v}|| = 0$ then any $\boldsymbol{\lambda}_D \in \text{F}_C$ is valid. When $||\boldsymbol{v}_D|| \neq 0$, the resulting friction force lies on the boundary of F_C, and is oppositely directed,

$$\boldsymbol{\lambda}_D = -\mu\lambda_n \frac{\boldsymbol{v}_D}{||\boldsymbol{v}_D||}. \tag{49}$$

This is equivalent to the proximal mapping,

$$\boldsymbol{\lambda}_D = \text{prox}_{\text{F}_C}(\boldsymbol{\lambda}_D - r_D \boldsymbol{v}_D), \quad r_D > 0. \tag{50}$$

The full contact force problem – for a single contact point – can be stated as,

$$\begin{aligned} \boldsymbol{M}(\boldsymbol{q})\frac{d\boldsymbol{v}}{dt} - \boldsymbol{J}_n^T \lambda_n - \boldsymbol{J}_D^T \boldsymbol{\lambda}_D - \boldsymbol{F} &= 0, & \text{(51a)} \\ \lambda_n - \text{prox}_{\mathbb{R}_+}(\lambda_n - r_n v_n) &= 0, & \text{(51b)} \\ \boldsymbol{\lambda}_D - \text{prox}_{\text{F}_C}(\boldsymbol{\lambda}_D - r_D \boldsymbol{v}_D) &= 0. & \text{(51c)} \end{aligned}$$

The parameterization, $\boldsymbol{J}_D$, of the frictional plane is the same as the one used in the LCP formulation, although any parameterization of a convex set could be used in its place. A more detailed derivation of this model is presented in Section 5.

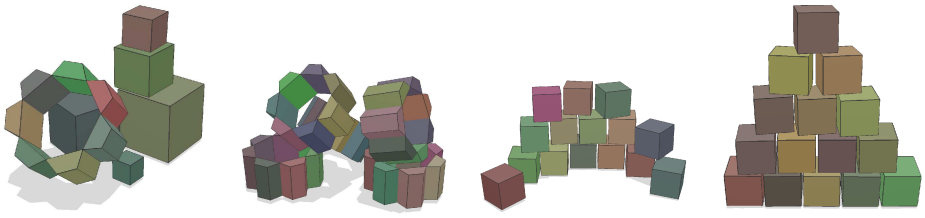
4 The Fischer–Newton Method – VRIPHYS 2009

I submit part of this thesis in the form of a paper [SNE09]. The paper covers large parts of the goals set for the thesis, albeit in a highly condensed version. The paper was submitted, in collaboration with Morten Silcowitz and Kenny Erleben, to the Virtual Reality and Interactive Physical Simulation(VRIPHYS) 2009 workshop. The subject of the paper is a contact force problem solver, based on a reformulation of the complementarity problem describing the contact force determination problem.

The reformulation described in the paper is not entirely novel, Billups [Bil95] used a similar approach to reformulate a mixed linear complementarity problem with fixed bounds. However, the problem we reformulate is a nonlinear complementarity problem with variable bounds. Also, novelty lies in the application to the contact force problem and the convergence analysis of this. To support this statement, I quote all three reviewers:

> reviewer#0: *This paper presents a novel technique for computing contact forces.*
>
> reviewer#4: *The approach seems to be a fresh look at the problem...*
>
> reviewer#5: *the paper brings some original work...*

The paper was accepted with minor revisions. The paper and the reviewer comments are included as Appendix A.



(a) (b) (c) (d)

Figure 7: *Example setups from the Fischer–Newton simulator.*

4.1 Summary

The motivation for the article, was to improve the level of accuracy of contact handling in interactive physical simulation. This is attempted by replacing the widely used Projected Gauss–Seidel(PGS) method by a Newton–based method. Such a method is presented and compared to the PGS method.

Two existing models for the contact force problem are restated, the LCP formulation of [ST96] where the friction cone is approximated by a polyhedral, and a NCP formulation also using a friction cone approximation. Based on the NCP formulation, a third model is posed where a reformulation of the NCP facilitates the use of a Newton–based solver. The function used for the reformulation is the Fischer function, first used in [Fis92] to solve optimization problems.

4.2 Reviewers Comments

The review was a double-blind review, in the sense that both the submitting authors and the reviewers were anonymous. Our paper was reviewed by three members of the review board, and although they did not agree completely on the individual questions, all three of them gave the overall recommendation score 7, out of a possible 9. The paper has been rated R, which in this context means that it *presents innovative research results*. As the overall recommendation score indicates, the review comments are generally positive. The few negative comments are mostly directed at the presentation – language, citations, level of detail – and not at the novel contributions.

4.3 Elaborations

Because of restrictions on the number of pages allowed for the submission of this paper, it was not possible to address all areas in detail. The following paragraphs are elaborations on two sub-methods, which were only briefly outlined in the paper.

The Quasi Fletcher–Reeves Method In its purest form, the Newton method requires an exact solution of a matrix equation. This is an operation with a time complexity of $O(n^3)$, to be performed at each iteration of the

Algorithm 1 Conjugate Gradient [NW99]

1: Given x_0
2: Initialize variables $k \Leftarrow 0$, $\boldsymbol{r}_0 = \boldsymbol{A}\boldsymbol{x}_0 - \boldsymbol{b}$, $\Delta\boldsymbol{x}_0 \Leftarrow -\boldsymbol{r}_0$,
3: **while** $\boldsymbol{r}_k \neq \boldsymbol{0}$ **do**
4: $\quad \alpha \Leftarrow \frac{\boldsymbol{r}^T\boldsymbol{r}}{\Delta\boldsymbol{x}_k^T\boldsymbol{A}\Delta\boldsymbol{x}_k}$
5: $\quad \boldsymbol{x}_{k+1} \Leftarrow \boldsymbol{x}_k + \alpha\Delta\boldsymbol{x}_k$
6: $\quad \boldsymbol{r}_{k+1} \Leftarrow \boldsymbol{r}_k + \alpha\boldsymbol{A}\Delta\boldsymbol{x}_k$
7: $\quad \beta \Leftarrow \frac{\boldsymbol{r}_k^T\boldsymbol{r}_k}{\boldsymbol{r}_{k+1}^T\boldsymbol{r}_{k+1}}$
8: $\quad \Delta\boldsymbol{x}_{k+1} \Leftarrow -\boldsymbol{r}_{k+1} + \beta\Delta\boldsymbol{x}_k$
9: $\quad k \Leftarrow k + 1$
10: **end while**

Newton method. Even for relatively small matrices, this is intractably expensive. An alternative is an inexact solution of the equation, which on one side invalidates all quadratic convergence claims, but on the other side might improve performance to a point where it can be called interactive. Since the intended usage of the method is for interactive simulation, we chose to use an inexact iterative solver to solve the Newton system. Algorithm 1 shows pseudocode for the Conjugated Gradient(CG) method. The CG method is an iterative method for solving linear systems of equations like $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. When $\boldsymbol{A}$ is a positive definite, symmetric $n \times n$ matrix, the CG method is known to terminate at the solution in n iterations. The system we wish to solve is the generalized Newton system,

$$\boldsymbol{H}_{\Phi(\boldsymbol{x}_k)}\Delta\boldsymbol{x}_k = -\boldsymbol{\Phi}(\boldsymbol{x}_k), \tag{52}$$

in which the generalized Jacobian $\boldsymbol{H}_{\Phi(\boldsymbol{x}_k)}$ is not guaranteed to be neither symmetric nor positive definite. One could argue that we should use a method that can handle non-symmetric systems, such as the Generalized Minimal Residual(GMRES) method or a bi–conjugate gradients(bi–CG) method. The GMRES method is expensive both work wise and storage wise, and its usability is critically dependent on choosing the optimal restarting heuristics. The bi–CG method is essentially a parallel method, using both $\boldsymbol{A}$ and $\boldsymbol{A}^T$ to compute a search direction. When $\boldsymbol{A}$ is symmetric, this amounts to performing two identical CG iterations per bi–CG iteration. So instead

Algorithm 2 Quasi Fletcher–Reeves

1: Given $\boldsymbol{x}_0$
2: Set error tolerance level ϵ and maximum iteration count k_{max}
3: Initialize variables $k \Leftarrow 0$, $\boldsymbol{r}_0 = \boldsymbol{\Phi}(\boldsymbol{x}_0)$, $\Delta\boldsymbol{x}_0 = \boldsymbol{r}_0$
4: **while** $\boldsymbol{r}_k^T \boldsymbol{r}_k < \epsilon \wedge k < k_{max}$ **do**
5: $\quad \Delta\hat{\boldsymbol{x}}_k \Leftarrow \boldsymbol{H}_{\Phi(x_k)} \Delta\boldsymbol{x}_k$ {Use equation (53)}
6: $\quad \alpha \Leftarrow \frac{\boldsymbol{r}_k^T \boldsymbol{r}_k}{\Delta\boldsymbol{x}_k^T \Delta\hat{\boldsymbol{x}}_k}$
7: $\quad \boldsymbol{x}_{k+1} \Leftarrow \boldsymbol{x}_k + \alpha\Delta\boldsymbol{x}_k$
8: $\quad \boldsymbol{r}_{k+1} \Leftarrow \boldsymbol{r}_k + \alpha\Delta\hat{\boldsymbol{x}}_k$
9: $\quad \beta \Leftarrow \frac{\boldsymbol{r}_k^T \boldsymbol{r}_k}{\boldsymbol{r}_{k+1}^T \boldsymbol{r}_{k+1}}$
10: $\quad \Delta\boldsymbol{x}_{k+1} \Leftarrow -\boldsymbol{r}_{k+1} + \beta\Delta\boldsymbol{x}_k$
11: $\quad k \Leftarrow k + 1$
12: **end while**

of putting a lot of work into methods that will most likely require more time and storage – two sparse commodities in interactive applications – we just assume that the CG method will work *well enough* for the generalized Newton method to converge. Some minor changes has been made to the CG method, in part to ensure termination and in part to improve performance.

The resulting CG–based method is named the Quasi Fletcher–Reeves(QFR) method, reflecting the use of a difference approximation and the application of the CG method to a nonlinear problem. Algorithm 2 shows pseudocode for the QFR method. The two main changes are the altered termination criteria and the finite difference approximation of the product $\boldsymbol{H}_{\boldsymbol{\Phi}(\boldsymbol{x}_k)}\Delta\boldsymbol{x}$ [NW99],

$$\boldsymbol{H}_{\Phi(\boldsymbol{x}_k)}\Delta\boldsymbol{x} \approx \frac{\Phi(\boldsymbol{x} + h\boldsymbol{p}) - \Phi(\boldsymbol{x})}{h}, \qquad h > 0. \tag{53}$$

Because $\boldsymbol{H}_{\boldsymbol{\Phi}(\boldsymbol{x}_k)}$ is not necessarily symmetric and positive definite, there is no guarantee that the CG method will converge to the correct solution. Therefore, an error tolerance level is introduced along with an upper limit on the iterations allowed. In the actual implementation, extra book keeping ensures that if the maximum iteration count is reached before the error tolerance level, then the solution with the smallest residual achieved so far is returned.

The reason for using the finite difference approximation (53), is to avoid

an explicit evaluation of $\boldsymbol{H}_{\boldsymbol{\Phi}(\boldsymbol{x}_k)}$. This is further motivated by the fact that $\boldsymbol{\Phi}(\boldsymbol{x})$ is known to be not everywhere (Fréchet) differentiable [Nie09b], thus further complicating the evaluation of its Jacobian. The overall results of the Fischer–Newton method show that the QFR method produces valid results, or at least results of such a quality that the Fischer–Newton method is able to converge to a usable solution. It would be interesting to analyze the convergence of the QFR method and the effect of substituting it with a more correct method for solving nonlinear equation systems. However, the focus of the was the underlying models and not the numerical specifics of an implementation.

Line Search Using the search direction, $\Delta\boldsymbol{x}$, given by the solution of the generalized Newton system (52), the next iterate of the generalized Newton method is computed. However, even with an exact solution to the Newton system, global convergence is not guaranteed. To achieve global convergence, we apply a line search method. Following the lines of [DLFK96], we use a backtracking line search method with an Armijo condition. The line search is performed on the natural merit function,

$$\boldsymbol{\Psi}(\boldsymbol{x}) = \frac{1}{2}||\boldsymbol{\Phi}(\boldsymbol{x})||^2. \tag{54}$$

The solution of the Newton system for a smooth system, is always a descent direction for the natural merit function, as long as $\boldsymbol{\Phi}(\boldsymbol{x}) \neq 0$. The same applies for the generalized Newton system for a nonsmooth system when $\boldsymbol{H}_{\boldsymbol{\Phi}(\boldsymbol{x}_k)} \in \partial\boldsymbol{\Phi}(\boldsymbol{x}_k)$. To realize this, we use the rules of standard nonsmooth calculus [Cla90] to get,

$$\nabla\boldsymbol{\Psi}(\boldsymbol{x}_k) = \boldsymbol{V}^T\boldsymbol{\Phi}(\boldsymbol{x}_k), \quad \forall\boldsymbol{V} \in \partial\Phi(\boldsymbol{x}_k). \tag{55}$$

Since $\Delta\boldsymbol{x}_k$ is a descent direction of $\boldsymbol{\Psi}$ if and only if $\nabla\boldsymbol{\Psi}(\boldsymbol{x}_k)^T\Delta\boldsymbol{x}_k < 0$, we check by substitution that this holds for the $\Delta\boldsymbol{x}_k$ found by solving

$\boldsymbol{H}_{\Phi(\boldsymbol{x}_k)}\Delta\boldsymbol{x}_k = -\boldsymbol{\Phi}(\boldsymbol{x}_k)$,

$$\begin{aligned}
\nabla\boldsymbol{\Psi}(\boldsymbol{x}_k)^T\Delta\boldsymbol{x}_k &= (\boldsymbol{V}^T\boldsymbol{\Phi}(\boldsymbol{x}_k))^T\Delta\boldsymbol{x}_k && \text{(56a)}\\
&= (\boldsymbol{H}^T_{\Phi(\boldsymbol{x}_k)}\boldsymbol{\Phi}(\boldsymbol{x}_k))^T\Delta\boldsymbol{x}_k && \text{(56b)}\\
&= (\boldsymbol{\Phi}(\boldsymbol{x}_k)^T\boldsymbol{H}_{\Phi(\boldsymbol{x}_k)})\Delta\boldsymbol{x}_k && \text{(56c)}\\
&= \boldsymbol{\Phi}(\boldsymbol{x}_k)^T(-\boldsymbol{\Phi}(\boldsymbol{x}_k)) && \text{(56d)}\\
&= -||\boldsymbol{\Phi}(\boldsymbol{x}_k)||^2 && \text{(56e)}\\
&< 0. && \text{(56f)}
\end{aligned}$$

The substitution performed in (56b) is valid since $\boldsymbol{H}_{\Phi(\boldsymbol{x}_k)} \in \partial\boldsymbol{\Phi}_k$. The strict inequality in (56f) comes from the precondition that $\boldsymbol{\Phi}(\boldsymbol{x}_k) \neq 0$. This should justify performing the line search on the natural merit function.

As noted, we use a backtracking line search method with an Armijo condition,

$$\boldsymbol{\Psi}(\boldsymbol{x} + \alpha\Delta\boldsymbol{x}) \leq \boldsymbol{\Psi}(\boldsymbol{x}) + \alpha c\nabla\boldsymbol{\Psi}(\boldsymbol{x})^T\Delta\boldsymbol{x}, \quad c \in (0;1) \tag{57}$$

where $\alpha \in (0;1]$ is the step size we wish to determine. As this is a backtracking method, we start with the largest α possible and gradually reduce it until the Armijo condition (57) is satisfied. A wide variety of heuristics for this reduction is available, we simply chose the decreasing sequence $\alpha = 2^{-k}$ where $k = \{0, 1, 2, \ldots\}$. Using (56), the Armijo condition in (57) can be reduced to,

$$\boldsymbol{\Psi}(\boldsymbol{x} + \alpha\Delta\boldsymbol{x}) \leq (1 - 2\alpha c)\boldsymbol{\Psi}(\boldsymbol{x}), \quad c \in (0;1) \tag{58}$$

Tests revealed a clear improvement in convergence when using the line search, especially when friction forces were included.

4.4 Conclusion

The Fischer–Newton method shows some improvement in accuracy compared to the Projected Gauss–Seidel method(PGS), but not for all possible configurations. The greatest improvement over the PGS method is for simulations of simple two box setups with large mass ratios. However, as the complexity of the scenes grows, the gained accuracy from using the Fischer–Newton method decreases. As well as losing accuracy when the scene complexity

grows, the Fischer–Newton method ceases to operate at interactive speed. Thus, the Fischer–Newton method can only be called interactive for small to medium sized setups. Further, the results show that when large friction forces are present in the setup, the generalized Newton system suffers from over–determinacy.

At this point, two main routes for future work presents them selves:

The method: Assuming a sound model, improve the method. This could include a better solver for the generalized Newton system, more accurate approximations of the generalized Jacobian and more strict convergence/termination criteria.

The model: Assuming a sound method, improve the model. The friction model used, is an approximation to Coulomb's friction, maybe the approximation could be improved. The reformulation used, may be too hard to solve, other reformulations could be examined.

The convergence plots in [SNE09] are divided in normal and frictional contributions. A general trend is that the normal contribution shows a relatively smooth convergence curve, where the frictional contribution is generally more jagged. This could imply that the model for friction is flawed, or that the method is not suited for solving problems constructed by this model. Either way, I feel convinced that the main problem with the Fischer–Newton method is strongly connected to the friction model used. In [PNE10](Appendix B), the same friction model is used in a different reformulation of the contact force problem, and solved using a different method. Even with these changes, the same conclusion is reached, that the friction model is far from optimal.

The improved handling of large mass ratios and general improvement in accuracy [Nie09a], makes the generalized Newton method interesting. Still, optimization is needed so that interactivity is feasible for larger setups.

For now, I will turn my attention to the contact force model with the intent of improving the modeling of frictional forces.

5 Proximal Mapping

The contact model roughly outlined in Section 3.3.3 is an alternative to the Fischer function model used in [SNE09]. However, in contrast to the Fischer function model, the proximal mapping approach has been applied to frictional contact problems in literature since [AC91]. The method is based on projections onto closed convex sets. This section is divided in two parts. The first part is an elaboration on the contact force determination problem stated as a proximal mapping problem. The second part focuses on how to solve a proximal mapping problem, in an effort to lay the foundation for an implementation. As with the previous model derivations, the model derived in this section is for a single contact point.

5.1 Mathematical Model

A contact force is the sum of its normal contribution and its frictional contribution. Thus, for a single contact point the contact force is,

$$F_{contact} = \boldsymbol{J}_n^T \lambda_n + \boldsymbol{J}_D^T \boldsymbol{\lambda}_D. \tag{59}$$

Although the two force contributions are not strictly independent of each other, the model often used in many simulators is a decoupling of the two. The trend is to first compute the normal contribution, and then use the computed value as a constant when computing the frictional contribution [KSJP08]. As this is the approach used in both [SNE09] and [PNE10], this section on contact forces will also be split in to a normal and a frictional part.

5.1.1 Normal Forces

The normal force acts orthogonally to the contact plane, it is non-attractive so its magnitude is non-negative, $\lambda_n \geq 0$. The relative contact velocity in normal direction is constrained the same way, such that $v_n \geq 0$. Together they enforce a non-penetration constraint, so if the contact velocity is greater than zero then the normal force is zero and vice versa. This can be stated as the one-dimensional complementarity problem,

$$\lambda_n \geq 0, \quad v_n \geq 0, \quad \lambda_n v_n = 0. \tag{60}$$

Theorem 5.1 *The one–dimensional complementarity problem,*

$$a \geq 0, \quad b \geq 0, \quad ab = 0,$$

is equivalent to the proximal mapping,

$$a = \text{prox}_{\mathbb{R}^+}(a - rb), \quad \forall r > 0,$$

where $\mathbb{R}^+$ *is the set of positive real numbers including* 0*.*

Proof To prove this, let us consider the three possible solutions to the one–dimensional complementarity problem,

1. If $a = 0$, then $0 = \text{prox}_{\mathbb{R}^+}(-rb)$. Since $-rb < 0$, the closest point to $-rb \in \mathbb{R}^+$ is 0.
2. If $b = 0$, then $a = \text{prox}_{\mathbb{R}^+}(a)$. Since $a > 0$, the closest point to $a \in \mathbb{R}^+$ is a.
3. If $a, b = 0$, then $0 = \text{prox}_{\mathbb{R}^+}(0)$. The closest point to $0 \in \mathbb{R}^+$ is 0.

Figure 8 visualizes the proof. □

Using Theorem 5.1, we can write the non-penetration constraint (60) as,

$$\lambda_n = \text{prox}_{\mathbb{R}^+}(\lambda_n - r_n v_n), \quad r_n > 0. \tag{61}$$

The relative contact velocity in the normal direction is $v_n = \boldsymbol{J}_n \boldsymbol{v}$.

5.1.2 Frictional Forces

The work done by the friction force $\boldsymbol{\lambda}_D$ is defined as $W_\lambda = \boldsymbol{\lambda}_D^T \boldsymbol{v}_D$. The principle of maximum dissipation states that the work done by $\boldsymbol{\lambda}_D$ must be equal to or lesser than the work done by any other friction force $\boldsymbol{\gamma}_D \in F_C(\mu\lambda_n)$. This can be expressed as,

$$\boldsymbol{\lambda}_D^T \boldsymbol{v}_D \leq \boldsymbol{\gamma}_D^T \boldsymbol{v}_D, \quad \forall \boldsymbol{\gamma}_D \in F_C(\mu\lambda_n) \tag{62}$$

which is equivalent to the variational inequality problem $\text{VI}(\boldsymbol{v}_D, F_C(\mu\lambda_n))$.

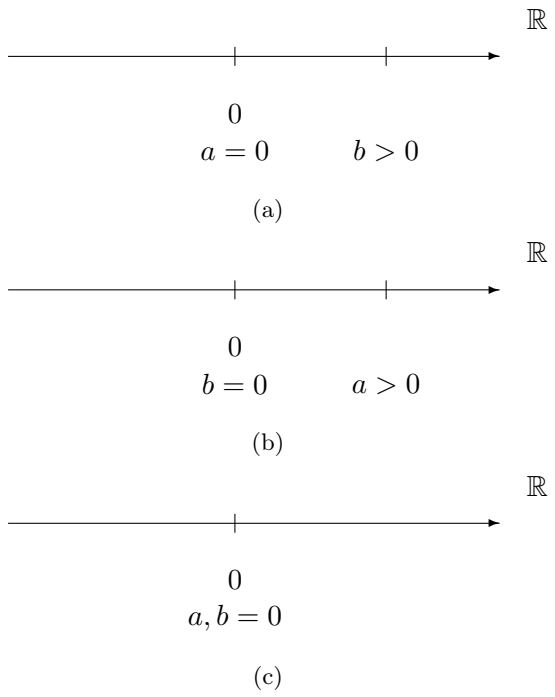


Figure 8: *A geometric interpretation of the three possible solutions to the one–dimensional linear complementarity problem* $a \geq 0,\ b \geq 0,\ ab = 0$.

Definition 5.1 (Variational inequality problem) *The variational inequality problem VI*(f, K) *is to determine* $\boldsymbol{x}^* \in K \subset \mathbb{R}^n$ *such that,*

$$f(\boldsymbol{x}^*)^T(\boldsymbol{z} - \boldsymbol{x}^*) \geq 0, \quad \forall \boldsymbol{z} \in K \tag{63}$$

where f *is a given continuous function* $f : K \to \mathbb{R}^n$ *and* K *is a given closed convex set.*

A variational inequality problem can be stated as an equivalent proximal mapping. To prove this, we first need to establish some properties of the proximal mapping.

Proposition 5.2 *If* K *is a closed convex subset of* $\mathbb{R}^n$ *and* $\boldsymbol{x} \in \mathbb{R}^n$*, then there exists a unique* $\boldsymbol{x}^* \in K$ *such that,*

$$||\boldsymbol{x} - \boldsymbol{x}^*|| \leq ||\boldsymbol{x} - \boldsymbol{z}||, \quad \forall \boldsymbol{z} \in K$$

and $\boldsymbol{x}^*$ *is the unique solution to the optimization problem,*

$$\boldsymbol{x}^* = \text{prox}_K(\boldsymbol{x}) = \min_{\boldsymbol{z} \in K} ||\boldsymbol{x} - \boldsymbol{z}||$$

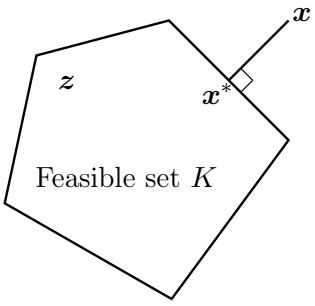


Figure 9: *The orthogonal projection,* $\boldsymbol{x}^* = \text{prox}_K(\boldsymbol{x})$*, of* $\boldsymbol{x}$ *onto the convex feasible set* K *is the minimizer of* $\min_{\boldsymbol{z}\in K} ||\boldsymbol{x} - \boldsymbol{z}||$.

Proof Assume that $\boldsymbol{x}^* \neq \min_{\boldsymbol{z}\in K} ||\boldsymbol{x} - \boldsymbol{z}||$ then there exists a $\boldsymbol{z}' \in K$ such that,

$$||\boldsymbol{x} - \boldsymbol{x}^*|| \geq ||\boldsymbol{x} - \boldsymbol{z}'||$$

which contradicts the premise of Proposition 5.2. A visualization of this is shown in Figure 9. □

Theorem 5.3 *Assume that* K *is a closed convex set, then,*

$$\boldsymbol{x}^* = \text{prox}_K(\boldsymbol{x}),$$

if and only if,

$$(\boldsymbol{x}^* - \boldsymbol{x})^T(\boldsymbol{z} - \boldsymbol{x}^*) \geq 0,$$

for any $\boldsymbol{z} \in K$.

Proof Define the function $g(\boldsymbol{z}) = ||\boldsymbol{x} - \boldsymbol{z}||^2$, and note that the minimizer of $g(\boldsymbol{z})$ over all $\boldsymbol{z} \in K$ is equivalent to $\boldsymbol{x}^* = \text{prox}_K(\boldsymbol{x})$. The sufficient KKT optimality condition for $\boldsymbol{x}^*$ to be a feasible solution to the constrained quadratic optimization problem, $\min_{\boldsymbol{z}\in K} g(\boldsymbol{z})$, is defined as,

$$\nabla g(\boldsymbol{x}^*)^T(\boldsymbol{z} - \boldsymbol{x}^*) \geq 0, \quad \forall \boldsymbol{z} \in K. \tag{64}$$

Since $\nabla g(\boldsymbol{z}) = 2(\boldsymbol{z} - \boldsymbol{x})$, substitution gives,

$$2(\boldsymbol{x}^* - \boldsymbol{x})^T(\boldsymbol{z} - \boldsymbol{x}^*) \geq (\boldsymbol{x}^* - \boldsymbol{x})^T(\boldsymbol{z} - \boldsymbol{x}^*) \geq 0. \quad \square \tag{65}$$

Theorem 5.4 *Assume that K is a closed convex set, then $\boldsymbol{x}^*$ is a solution to VI(f, K) if and only if,*

$$\boldsymbol{x}^* = \text{prox}_K(\boldsymbol{x}^* - rf(\boldsymbol{x}^*)),$$

for any $r > 0$.

Proof Using Theorem 5.3 and substituting $(\boldsymbol{x}^* - rf(\boldsymbol{x}^*))$ for $\boldsymbol{x}$,

$$\begin{aligned}
(\boldsymbol{x}^* - (\boldsymbol{x}^* - rf(\boldsymbol{x}^*)))^T(\boldsymbol{z} - \boldsymbol{x}^*) &\geq 0, \quad \forall \boldsymbol{z} \in K && \text{(66a)}\\
rf(\boldsymbol{x}^*)^T(\boldsymbol{z} - \boldsymbol{x}^*) &\geq 0, \quad \forall \boldsymbol{z} \in K && \text{(66b)}\\
f(\boldsymbol{x}^*)^T(\boldsymbol{z} - \boldsymbol{x}^*) &\geq 0, \quad \forall \boldsymbol{z} \in K && \text{(66c)}
\end{aligned}$$

The last reduction follows from $r > 0$. By Definition 5.1, this concludes the proof. □

It should now be proven that the solution to,

$$\boldsymbol{\lambda}_t^T \boldsymbol{v}_t \leq \boldsymbol{\gamma}_t^T \boldsymbol{v}_t, \quad \forall \boldsymbol{\gamma}_t \in F_C(\mu\lambda_n), \tag{67}$$

is equivalent with,

$$\boldsymbol{\lambda}_t = \text{prox}_{F_C(\mu\lambda)}(\boldsymbol{\lambda}_t - r_t\boldsymbol{v}_t), \quad r_t > 0. \tag{68}$$

The connection between variational inequalities and proximal mappings can also be verified geometrically. Consider the necessary and sufficient conditions for $\boldsymbol{x}^*$ to be a solution to VI(f, K) [NZ95],

$$-f(\boldsymbol{x}^*) \in N_K(\boldsymbol{x}^*), \tag{69}$$

where $N_K(\boldsymbol{x}^*)$ is the normal cone of K at $\boldsymbol{x}^*$,

$$N_K(\boldsymbol{x}) = \{\boldsymbol{y} \in \mathbb{R}^n | \boldsymbol{y}^T(\boldsymbol{z} - \boldsymbol{x}) \leq 0, \quad \forall \boldsymbol{z} \in K\}. \tag{70}$$

Conversely, $\boldsymbol{x}^*$ is the unique solution to $\text{prox}_K(\boldsymbol{x}^* - rf(\boldsymbol{x}^*))$ if and only if $-rf(\boldsymbol{x}^*) \in N_K(\boldsymbol{x}^*)$. Since $r > 0$ is only a scaling factor, the condition reduces to $-f(\boldsymbol{x}^*) \in N_K(\boldsymbol{x}^*)$. As Figure 10 illustrates, for any point $\boldsymbol{x}$ outside of $N_K(\boldsymbol{x}^*)$, there exists a $\tilde{\boldsymbol{x}}$ such that,

$$||\boldsymbol{x} - \tilde{\boldsymbol{x}}|| \leq ||\boldsymbol{x} - \boldsymbol{x}^*||. \tag{71}$$

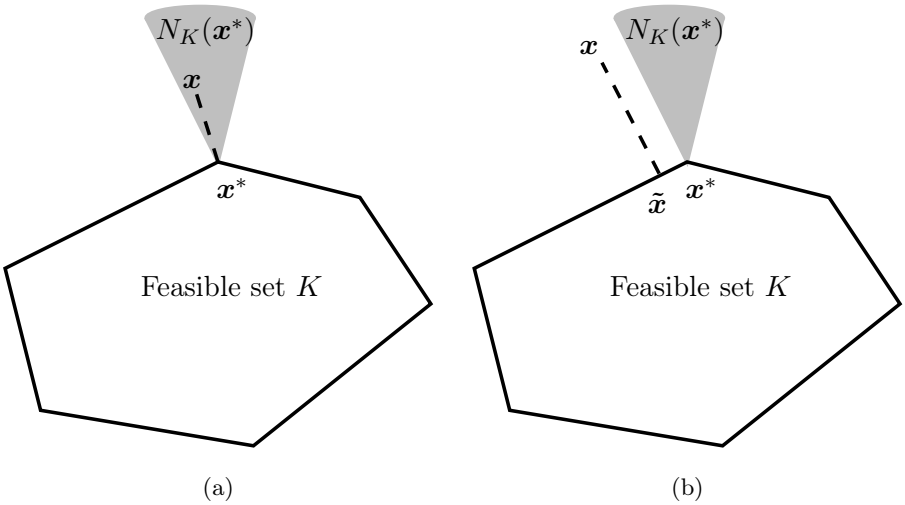


Figure 10: *The normal cone, $N_K(\boldsymbol{x}^*)$, of feasible set K at feasible point $\boldsymbol{x}^*$, illustrated by a gray cone. (a) The closest point in K to any $x \in N_K(\boldsymbol{x}^*)$ is $\boldsymbol{x}^*$, or equivalently $\boldsymbol{x}^* = \text{prox}_K(\boldsymbol{x}), \forall \boldsymbol{x} \in N_K(\boldsymbol{x}^*)$. (b) When $\boldsymbol{x} \notin N_K(x^*)$, then $\boldsymbol{x}^*$ is no longer closest point in K.*

Combining all this, the system to solve is,

$$\begin{aligned} \boldsymbol{M}(\boldsymbol{q})\frac{d\boldsymbol{v}}{dt} - \boldsymbol{F}_{ext} - \boldsymbol{J}_n^T\lambda_n - \boldsymbol{J}_D^T\boldsymbol{\lambda}_D &= 0, && \text{(72a)} \\ \frac{d\boldsymbol{q}}{dt} &= \boldsymbol{v}, && \text{(72b)} \\ \lambda_n - \text{prox}_{\mathbb{R}_+}(\lambda_n - rv_n) &= 0, && \text{(72c)} \\ \boldsymbol{\lambda}_D - \text{prox}_{F_C(\mu\lambda_n)}(\boldsymbol{\lambda}_D - r\boldsymbol{v}_D) &= 0. && \text{(72d)} \end{aligned}$$

Throughout the derivation of the proximal mapping model, the frictional plane has been parameterized by $\boldsymbol{J}_D$, using the Stewart–Trinkle friction model presented in Section 3.3.1. However, the only constraint imposed on friction in this model, is the principle of maximum dissipation (62). This principle is valid for any friction model, therefore we can use any closed convex set to model friction. Thus, using the proximal mapping formulation we can easily apply anisotropic friction.

5.2 Numerical Model

Using inspiration from [AC91, LG03], I propose the following iterative scheme. The superscript k denotes iteration count. Given an initial estimate of λ_n^0

and $\boldsymbol{\lambda}_D^0$, iterate over

$$\begin{aligned} \boldsymbol{M}(\boldsymbol{q})\Delta\boldsymbol{v}^{k+1} &= h\boldsymbol{J}_n^T\lambda_n^k + h\boldsymbol{J}_D^T\boldsymbol{\lambda}_D^k + h\boldsymbol{F}_{ext}, && \text{(73a)} \\ \lambda_n^{k+1} &= \text{prox}_{\mathbb{R}^+}(\lambda_n^k - r\boldsymbol{J}_n(\Delta\boldsymbol{v}^{k+1} + \boldsymbol{v}^k)), && \text{(73b)} \\ \boldsymbol{\lambda}_D^{k+1} &= \text{prox}_{F_C(\mu\lambda_n^{k+1})}(\boldsymbol{\lambda}_D^k - r\boldsymbol{J}_D(\Delta\boldsymbol{v}^{k+1} + \boldsymbol{v}^k)), && \text{(73c)} \end{aligned}$$

until a convergence criteria is met. This could be an error tolerance on the contact force magnitudes, $||\lambda_n^{k+1} - \lambda_n^k|| + ||\boldsymbol{\lambda}_D^{k+1} - \boldsymbol{\lambda}_D^k|| < \epsilon$.Alart and Curnier [AC91] claim that this hierarchical solution of forces (73) is physically sound. Once convergence is reached, the computed values of $\Delta\boldsymbol{v}^{k+1}$, λ_n^{k+1} and $\boldsymbol{\lambda}_D^{k+1}$ are used to update the system according to the time stepping scheme chosen. The Moreau time stepping scheme [Mor88] is often used as an inspiration in papers on complementarity–based velocity formulations.

As mentioned in Section 5.1.2, from a mathematical viewpoint, the proximal mapping formulation is not constrained to use a specific friction model. Using the iterative projection scheme outlined in (73), this freedom of choice is extended so that using a larger parameterization of the friction cone does not affect interactivity in the same degree as with the LCP/NCP formulations. Consider (73a),

$$\underbrace{\boldsymbol{M}(\boldsymbol{q}^k)}_{\boldsymbol{A}_{\text{prox}}}\underbrace{\Delta\boldsymbol{v}^{k+1}}_{\boldsymbol{x}_{\text{prox}}} - \underbrace{(h\boldsymbol{J}_n^T\lambda_n^k + h\boldsymbol{J}_D^T\boldsymbol{\lambda}_D^k + h\boldsymbol{F}_{ext})}_{\boldsymbol{b}_{\text{prox}}} = 0, \tag{74}$$

changing the size of $\boldsymbol{J}_D$ has no impact on the size of the system to solve,

$$\boldsymbol{A}_{\text{prox}}\boldsymbol{x}_{\text{prox}} - \boldsymbol{b}_{\text{prox}} = 0. \tag{75}$$

The $\boldsymbol{b}_{\text{prox}}$ vector will be more expensive to compute, but still it is tempting to assume that greater friction accuracy might be attainable in interactive settings.

Finally, there is the choice of the r-factor in (73b) and (73c). Although any choice of $r > 0$ is equivalently good in the mathematical derivations of the proximal mapping formulation, when it comes to solving the system by a fixed-point iteration scheme, the choice of the optimal r-factor is crucial to obtain fast convergence. I will not discuss the strategies for choosing the optimal r-factors, instead I refer to [FGNU06].

6 Conclusion

Six months ago, I had to formulate a set of goals for the work of this thesis. The goals was thus set before having gained the knowledge I have now. For this reason, the goals reached may appear slightly different from those set. I started out thinking that the Fischer–Newton method might just be a contender to the title *state of the art* method in interactive contact force determination. Four months and a paper later, I faced the limitations of the Fischer–Newton method.

At that point, I had to make a decision. On one hand, I could give up on pursuing interactivity, and instead work on increasing accuracy and stability of the Fischer–Newton method. On the other hand, I could assess why the Fischer–Newton method is infeasible for large scale interactive settings and try to improve it.

The results presented in Appendices A, B and C indicate problems with modeling friction as a nonlinear complementarity problem. The three papers use different reformulations and solution methods, the common factor is the friction model. Based on the conclusions of the three papers, I focused on modeling the contact force problem using a third reformulation. The proximal mapping reformulation, presented in Section 5, holds great promise of both interactivity and improved accuracy. To recap on the goals,

- **To describe the contact force problem:** In Section 3, I describe the contact force problem in a top–down approach, starting with a system of bodies and ending up with the constraints of a single contact point.
- **To formulate a mathematical model of the contact force problem:** Five different models of the contact force problem are presented, three of them in Section 3 and the remaining two in Appendices A and B. Section 5 elaborates on the model outlined in Section 3.3.3.
- **To implement a Newton–based solution method:** The Fischer–Newton method presented in Appendix A, and elaborated on in Section 4, is as the name implies a Newton–based method. The method showed some promise of improved accuracy and stability, when compared to the projected Gauss-Seidel method. However, in many test

cases the Fischer-Newton method only performed at par with the projected Gauss-Seidel, but at a higher cost.

6.1 Pain and Agony

The use of Pain and Agony in the title of this thesis, is a relic from the time where I thought I would be working with the OpenTissue [OT] framework. Pain and Agony are two setups from OpenTissue, simulating two humanoid articulated figures in agonizing situations. I chose these two setups, because the OpenTissue solver based on the projected Gauss–Seidel(PGS) method is notoriously bad at handling them. Ideally, the two setups should settle into rest states, but the inaccuracy of the PGS method introduces energy in the system making it look like the figures are writhing with pain.

In Agony, the figure hangs from a beam with an extremely heavy ball attached at its feet. So for this setup, the difficulty lies in solving the joint constraints accurately when the system includes large mass ratios. As I have worked on contact forces exclusively, I have realized that the Agony setup is uninteresting in this context.

In large parts, the same applies to the Pain setup. However, there does exist some contact force in the setup. In the Pain setup, a figure is hung from four pillars by its hands and feet, placed on its torso is an extremely heavy ball. As with the Agony test, the task is to solve the joint constraints accurately in a system that includes large mass ratios. In addition, there are also large friction forces at play, making the system even harder to solve. Because the Fischer–Newton method is not implemented in the OpenTissue framework, a new Pain demo had to be configured. Figures 11 and 12 shows selected frames of the interactive simulation of the Pain setup. As the figures show, the ball is replaced by a box. This way, the setup is almost guaranteed to have more than a single contact point.

Figure 11 shows some interactions with the simulation, to show that the Fischer–Newton method can solve the joint constraints satisfyingly. The articulated figure is manipulated by pulling its leg or torso, and when it is released it falls back to its resting state. However, when the heavy box is used to interact with the articulated figure, the simulation becomes increas-

ingly instable. Instability tops when the box is placed on the torso of the articulated figure. At this point, all unfixed objects simply fall through the floor. I suspect, this is a result of trying to solve a system with no valid solution. When the box is removed from the system, the remaining system reestablishes it self and falls to rest again.

Although highly instable, at least the simulation is interactive running at a minimum of 25 frames per second.

6.2 Future Work

Obviously, I have to implement a solver that uses the proximal mapping reformulation. The reduction in problem size and the ability to model any convex friction set, makes the proximal mapping model extremely interesting in the context of the prototyping system proposed in Section 1.1. Being able to model anisotropic friction widens the application of the system, as well as increases plausibility. It could be interesting to examine the use of different friction models. Even though the size of the friction plane parameterization no longer affects the size of the contact force problem, there is still the memory footprint to consider. If the friction model doesn't need explicit representation, this could help keep the memory footprint down.

If the simulation blow-ups in the Pain test can be shown to be the result of the underlying nonlinear complementarity problem having no valid solution, this would be a great argument against using the nonlinear complementarity problem reformulation of the contact force problem.

If reality and plausibility are to be improved in interactive simulation, we have to move away from the damped and viscous simulations of the projected Gauss–Seidel. However disappointing the results of the Fischer–Newton were, at least it was able to simulate high frequency movements. In short, the next goal might just be a Prox–Newton method.

Figure 11: *Selected frames from a test run using the Pain setup. The first five frames show interactions from the user, pulling the articulated figure while the heavy box rests on the floor. Once the box is used in the interactions, the simulator shows sign of instability.*

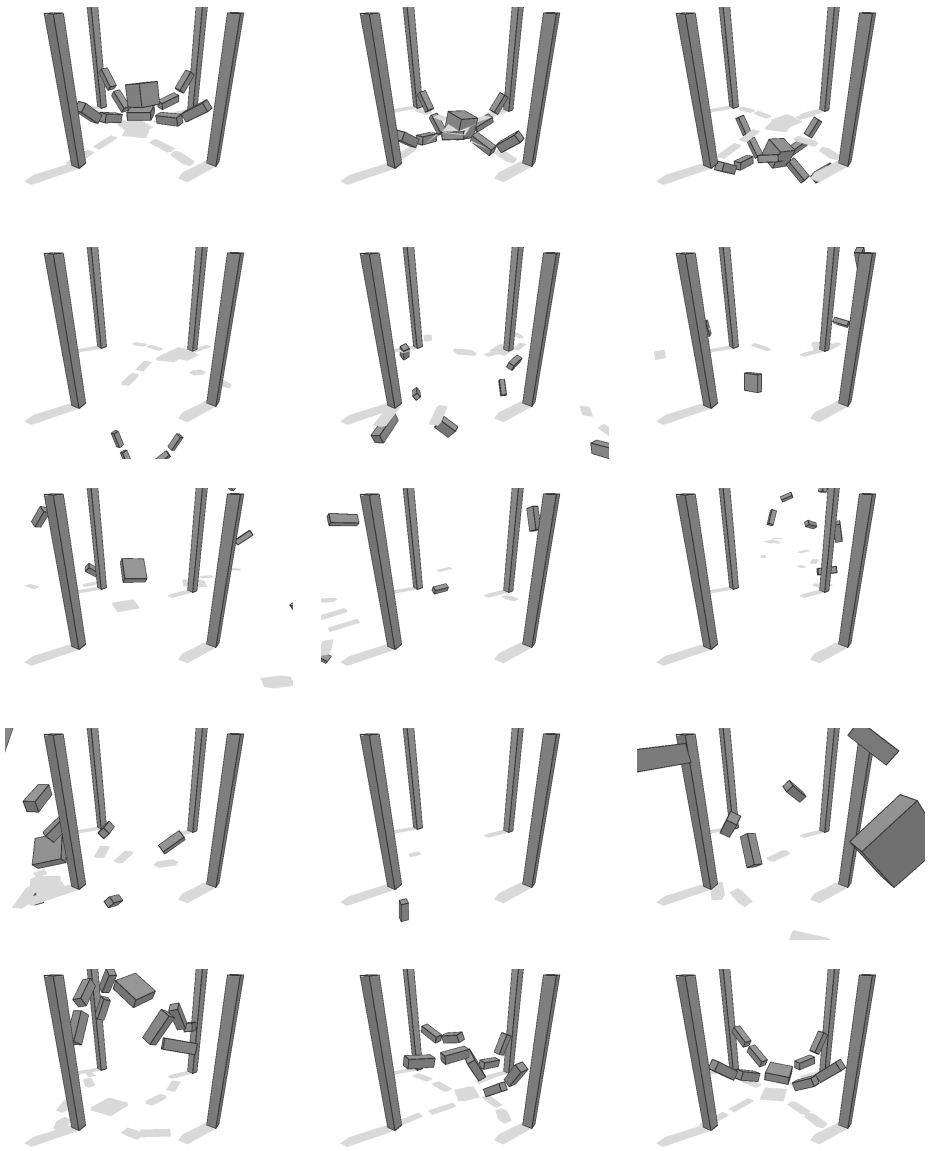


Figure 12: *Selected frames from a test run using the Pain setup. Once the heavy box is placed on top of the articulated figure, the simulation fails completely as both box and articulated figure fall through the floor. This is a clear example of simulation blow-up. As soon as the heavy box leaves the setup, the articulated figure jumps back in position and settles down.*

References

[AC91] Pierre Alart and Alain Curnier. A mixed formulation for frictional contact problems prone to newton like solution methods. *Comp. Methods Appl. Mech. Eng.*, 92(3):353–375, 1991.

[AP97] Mihai Anitescu and Florian A. Potra. Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlinear Dynamics. An International Journal of Nonlinear Dynamics and Chaos in Engineering Systems*, 14(3):231–247, 1997.

[Bar89] David Baraff. Analytical methods for dynamic simulation of non-penetrating rigid bodies. *SIGGRAPH Comput. Graph.*, 23(3):223–232, 1989.

[Bar91] David Baraff. Coping with friction for non-penetrating rigid body simulation. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 31–41. ACM New York, NY, USA, 1991.

[Bar94] David Baraff. Fast contact force computation for nonpenetrating rigid bodies. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 23–34, New York, NY, USA, 1994. ACM.

[Bil95] Stephen Clyde Billups. *Algorithms for complementarity problems and generalized equations.* PhD thesis, University of Wisconsin at Madison, Madison, WI, USA, 1995.

[Cha99] Anindya Chatterjee. On the realism of complementarity conditions in rigid body collisions. *Nonlinear Dynamics*, 20(2):159–168, 1999.

[Cla90] Frank H. Clarke. *Optimization and Nonsmooth Analysis.* Society for Industrial Mathematics, 1990.

[DLFK96] Tecla De Luca, Francisco Facchinei, and Christian Kanzow. A semismooth equation approach to the solution of nonlinear complementarity problems. *Mathematical Programming*, 75(3):407–439, 1996.

[Erl07] Kenny Erleben. Velocity-based shock propagation for multibody dynamics animation. *ACM Trans. Graph.*, 26(2):12, 2007.

[ESHD05] Kenny Erleben, Jon Sporring, Knud Henriksen, and Henrik Dohlmann. *Physics-based Animation*. Charles River Media, 2005.

[FAS] Ford Engine Assembly Line Simulation Tool. http://pdf.directindustry.com/pdf/lanner/ford-engine-assembly-line-simulation-tool/7080-38670-_2.html.

[FGNU06] Martin Foerg, Thomas Geier, Lutz Neumann, and Heinz Ulbrich. r-Factor strategies for the augmented Lagrangian approach in multi-body contact mechanics. In *Proceedings of the III. European Conference on Computational Mechanics, Lisbon*. Springer, 2006.

[Fis92] Andreas Fischer. A special newton-type optimization method. *Optimization*, 24(3):269–284, 1992.

[FK98] Michael C. Ferris and Christian Kanzow. Complementarity and related problems: A survey. *Mathematical Programming Technical Report*, pages 98–17, 1998.

[Gaz] Gazebo. http://playerstage.sourceforge.net/index.php?src=gazebo.

[Hah88] James K. Hahn. Realistic animation of rigid bodies. In *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 299–308, New York, NY, USA, 1988. ACM.

[JAJ98] Franck Jourdan, Pierre Alart, and Michel Jean. A Gauss-Seidel like algorithm to solve frictional contact problems. *Computer methods in applied mechanics and engineering*, 155(1-2):31–47, 1998.

[KEP05] Danny M. Kaufman, Timothy Edmunds, and Dinesh K. Pai. Fast frictional dynamics for rigid bodies. *ACM Trans. Graph.*, 24(3):946–956, 2005.

[KSJP08] Danny M. Kaufman, Shinjiro Sueda, Doug L. James, and Dinesh K. Pai. Staggered projections for frictional contact in multibody systems. *ACM Trans. Graph.*, 27(5):1–11, 2008.

[Lan70] Cornelius Lanczos. *The Variational Principles of Mechanics*. Dover Publications, Inc., 4 edition, 1970.

[LG03] Remco I. Leine and Christoph Glocker. A set-valued force law for spatial Coulomb–Contensou friction. *European Journal of Mechanics/A Solids*, 22(2):193–216, 2003.

[MC95] Brian Mirtich and John Canny. Impulse-based simulation of rigid bodies. In *Proceedings of the 1995 symposium on Interactive 3D graphics*. ACM New York, NY, USA, 1995.

[Mor88] Jean Jacques Moreau. Unilateral contact and dry friction in finite freedom dynamics. *Nonsmooth mechanics and Applications*, pages 1–82, 1988.

[MS04] Victor J. Milenkovic and Harald Schmidl. A fast impulsive contact suite for rigid body simulation. *IEEE Transactions on Visualization and Computer Graphics*, 10(2):189–197, 2004.

[Mur88] Katta G. Murty. *Linear Complementarity, Linear and Nonlinear Programming*. Helderman-Verlag, 1988.

[MW88] Matthew Moore and Jane Wilhelms. Collision detection and response for computer animation. In *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 289–298, New York, NY, USA, 1988. ACM.

[Nie09a] Sarah M. Niebe. Linear complementarity problems solved using newton's method. http://dl.getdropbox.com/u/2801395/niebe_nonlinear.pdf, May 2009.

[Nie09b] Sarah M. Niebe. A newton method for complementarity problems. http://dl.getdropbox.com/u/2801395/niebe_fischer.pdf, March 2009.

[NW99] Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer Series in Operations Research. Springer-Verlag, New York, 1999.

[NZ95] Anna Nagurney and Ding Zhang. *Projected dynamical systems and variational inequalities with applications*. Kluwer Academic Publishers, 1995.

[OT] OpenTissue. http://www.opentissue.org.

[PNE10] Morten Poulsen, Sarah Niebe, and Kenny Erleben. Heuristic convergence rate improvements of the projected gauss–seidel method for frictional contact problems. Submitted for WSCG, 2010.

[SNE09] Morten Silcowitz, Sarah Niebe, and Kenny Erleben. Nonsmooth Newton Method for Fischer Function Reformulation of Contact Force Problems for Interactive Rigid Body Simulation. In *VRIPHYS 09: Sixth Workshop in Virtual Reality Interactions and Physical Simulations*, pages 105–114. Eurographics Association, 2009.

[ST96] David E. Stewart and Jeff C. Trinkle. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *International Journal of Numerical Methods in Engineering*, 39(15):2673–2691, 1996.

[TB97] Lloyd N. Trefethen and David Bau. *Numerical linear algebra*. Society for Industrial Mathematics, 1997.

A VRIPHYS 2009 Reviews and Article

The following is the review comments received for the article *Nonsmooth Newton Method for Fischer Function Reformulation of Contact Force Problems for Interactive Rigid Body Simulation* submitted to the Vriphys 09 workshop. The submitted – and accepted – article is also included.

```
Dear authors,
we are very pleased to inform you of the acceptance of your paper for
the Workshop on Virtual Reality Interaction and Physical Simulation VRIPHYS
09:

-------
paper1006 Nonsmooth Newton Method for Fischer Function Reformulation of
Contact Force Problems for Interactive Rigid Body Simulation
Niebe, Sarah, Erleben, Kenny, Morten Silcowitz
-------

Attached you can find the comments of the referees.  In many cases, the
referees made a great effort to indicate potential improvements of your
paper.  So, please acknowledge their work by addressing their suggestions.
This is particularly true, if additional related work is mentioned in
the reviews or in the case of misspellings or ambiguous statements.  Take
the chance to improve the quality of the final version of *your* paper
by incorporating as many comments as possible.
Final papers are due to:

** September 4, 2009 **

You will soon be receiving an email containing the information needed
for the production of your final VRIPHYS 09 paper.

We are looking forward to your participation in the conference.

Please acknowledge the receipt of this email to srm-vriphys09@eg.org.

Sincerely,
Hartmut Prautzsch
```

Alfred Schmitt
Jan Bender
Matthias Teschner

http://www.vriphys.org/workshops/vriphys09/index.htm

==========
Reviews
==========

Reviewer # 0

**** Overall Recommendation (0 = totally unacceptable ... 9 = excellent)
7

**** Reviewer's Comments
This paper presents a novel technique for computing contact forces. The topic of the paper is relevant for the conference and it is well-written.

**** Paper Classification
R - Research paper (presents innovative research results)
P - Practice and experience (variants, applications, case studies)
S - State-of-the-art report (reviews of recent advances)
R

**** Summary
The paper presents a method to determine contact forces for rigid body simulation. The method is based on a reformulation of a complementarity problem as a root search problem. This is done by using a Fischer function. The root search problem is solved by a generalized Newton method.

**** Originality, Novelty (0 = totally unacceptable ... 9 = excellent)
7

I like the idea of the reformulation, since the Fischer function is easier to handle.

**** Clarity of Presentation (0 = totally unacceptable ... 9 = excellent)
6

Equation 1 is not clear for me. You have an equation of the form F = M u where u is a velocity vector. Newton's second law is F = m a. If you multiply a velocity, you will get an impulse instead of a force. Here I would like to see an explanation for you formulation.

**** Technical Soundness (0 = totally unacceptable ... 9 = excellent)
7

Good.

**** Importance, Utility (0 = totally unacceptable ... 9 = excellent)
6

I think this paper is quite important for the research in the area of rigid body simulations.

**** Suitable for event? (0 = totally unacceptable ... 9 = excellent)
7

Suitable.

**** Completeness of References (0 = totally unacceptable ... 9 = excellent)
5

I missed the paper ''Constraint-based collision and contact handling using impulses'' in the references. It combines a constraint-based approach with an impulse-based method and therefore, solves the problems of the impulse-based approach of Mirtich.

**** Could this approach be implemented by a graduate student? (0 = totally unacceptable ... 9 = excellent)
5

I think a good student could do this.

**** Confidence (1 = very unconfident... 5 = extremely confident)
4

**** Reviewer's Additional Comments For The Authors

Reviewer # 4

**** Overall Recommendation (0 = totally unacceptable ... 9 = excellent)
7

**** Reviewer's Comments
The approach seems to be a fresh look at the problem, and the results are quite impressive, especially for the scenarios with the large mass differences between the bodies.

**** Paper Classification
R - Research paper (presents innovative research results)
P - Practice and experience (variants, applications, case studies)
S - State-of-the-art report (reviews of recent advances)

**** Summary
The authors present an interesting new idea to solve the contact force problem. The idea is to reformulate the problem as a non-linear root search problem using a Fischer function.

**** Originality, Novelty (0 = totally unacceptable ... 9 = excellent)
9

**** Clarity of Presentation (0 = totally unacceptable ... 9 = excellent)
8

**** Technical Soundness (0 = totally unacceptable ... 9 = excellent)
8

**** Importance, Utility (0 = totally unacceptable ... 9 = excellent)

7

**** Suitable for event? (0 = totally unacceptable ... 9 = excellent)
9

**** Completeness of References (0 = totally unacceptable ... 9 = excellent)
8

**** Could this approach be implemented by a graduate student? (0 = totally unacceptable ... 9 = excellent)
6

This stuff is not exactly trivial. (But then, this is true for most physically-based simulation methods.)

**** Confidence (1 = very unconfident... 5 = extremely confident)
3

**** Reviewer's Additional Comments For The Authors

The writing should be improved. For instance, sentences like ''Since contact plane is a two dimensional, ...'' seem a bit odd. In addition, I think the punctuation should be looked over.

Reviewer # 5

**** Overall Recommendation (0 = totally unacceptable ... 9 = excellent)
7

**** Reviewer's Comments
the paper brings some original work and, although rather abundant of details that would better fit in a longer (journal) version of the paper, is well written. The arguments used and the results are convincing, therefore I'm glad to recommend acceptation.

**** Paper Classification
R - Research paper (presents innovative research results)

P - Practice and experience (variants, applications, case studies)
S - State-of-the-art report (reviews of recent advances)
R

**** Summary
The paper proposes a novel formulation of the contect handling problem
to be solved with non linear mninimization algorithms.

**** Originality, Novelty (0 = totally unacceptable ... 9 = excellent)
7

To my knowledge the paper proposes a significant original contribution
to the field

**** Clarity of Presentation (0 = totally unacceptable ... 9 = excellent)
6

The paper is nicely written but in my opinion it could be improved by
shortening the mathamtical derivations where possible (for example removing
the proof of pag 4) and including more intutitive explanations.

**** Technical Soundness (0 = totally unacceptable ... 9 = excellent)
8

The paper include a number of test to show effectiveness of the approach
(and included a demonstrating video with the submission). On the downside,
all the objects have very simple geometry. ALthough the authors pointed
out that it becomes increasily difficult to asses the quality of the simulation
as the contact geometry become more complex, it is also true that it serves
as a stress test for the eficiency of the algorithm (more complex surfaces
imply more contact points)

**** Importance, Utility (0 = totally unacceptable ... 9 = excellent)
8

**** Suitable for event? (0 = totally unacceptable ... 9 = excellent)
8

**** Completeness of References (0 = totally unacceptable ... 9 = excellent)
7

**** Could this approach be implemented by a graduate student? (0 = totally unacceptable ... 9 = excellent)
7

a skilled graduate student would find the details he/she need in the paper. It would not be straighforward.

**** Confidence (1 = very unconfident... 5 = extremely confident)
3

**** Reviewer's Additional Comments For The Authors

Workshop on Virtual Reality Interaction and Physical Simulation VRIPHYS (2009)
H. Prautzsch, A. Schmitt, J. Bender, M. Teschner (Editors)

Nonsmooth Newton Method for Fischer Function Reformulation of Contact Force Problems for Interactive Rigid Body Simulation

M. Silcowitz S. Niebe K. Erleben

Department of Computer Science, University of Copenhagen, Denmark

Abstract

In interactive physical simulation, contact forces are applied to prevent rigid bodies from penetrating each other. Accurate contact force determination is a computationally hard problem. Thus, in practice one trades accuracy for performance. The result is visual artifacts such as viscous or damped contact response. In this paper, we present a new approach to contact force determination. We reformulate the contact force problem as a nonlinear root search problem, using a Fischer function. We solve this problem using a generalized Newton method. Our new Fischer–Newton method shows improved qualities for specific configurations where the most widespread alternative, the Projected Gauss-Seidel method, fails. Experiments show superior convergence properties of the exact Fischer–Newton method.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.5]: Physically based modeling—Computer Graphics [I.3.7]: Animation—Mathematics of Computing [G.1.6]: Nonlinear programming—

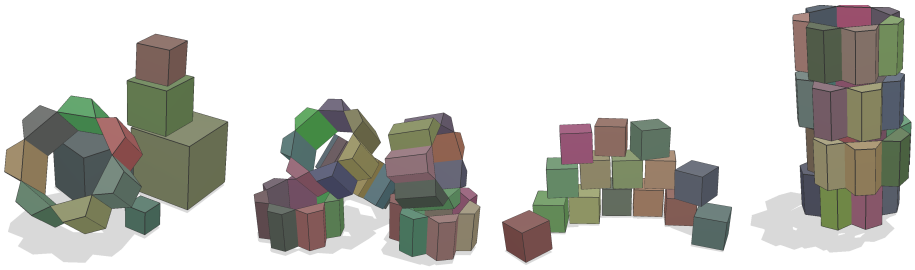
Keywords: Contact Force Problem, Complementarity Formulation, Newton Method, Fischer Function

1. Shortcomings of State-of-Art

Most open source software for interactive real-time rigid body simulation uses the widespread Projected Gauss–Seidel (PGS) method, examples are Bullet and Open Dynamics Engine. However, the PGS method is not always satisfactory, it suffers from two problems: the linear convergence rate [CPS92] and inaccurate friction forces in stacks [KSJP08]. The linear convergence results in viscous motion at contacts and loss of high frequency effects. The viscous appearance results in a time delay in the contact response and reduces plausibility [ODGK03]. This has motivated us to develop a new numerical method, based on a nonsmooth reformulation of the contact force problem. The reformulation transforms a nonlinear complementarity problem (NCP) formulation into a nonsmooth root search problem. Our method is compared to the PGS method for interactive simulation.

Rigid body simulation was introduced to the graphics community in the late 1980's [Hah88, MW88] using penalty based and impulse based approaches to describe the physical interactions. Penalty based simulation is not easily adopted to different simulations without parameter tuning. The impulse based approach was extended and improved [Mir96], however stacking was a problem and it suffered from creeping, these problems has since been rectified [GBF03]. Constraint based simulation has received much attention as an alternative [Bar94] to penalty and impulse based simulation. Constraint based simulation can be classified into two groups: maximal and minimal coordinate methods [Fea98]. The focus of this paper is maximal coordinates methods, which are dominated by complementarity formulations. There are alternatives to complementarity formulations, based on kinetic energy [MS01, MS04] and motion space [RKC03]. However, the former solves a more general problem and is not attractive for performance reasons, and the latter does not include frictional forces.

Complementarity formulations come in two flavors: acceleration based formulations [TTP01] and velocity based formulations [ST96]. Acceleration based formulations can-



(a) (b) (c) (d)

Figure 1: *Contact force computations using our Fischer–Newton method are more efficient for large mass ratios. In (a) we illustrate a random configuration of mixed objects, (b) a pile of non-convex gears, (c) a wall of boxes, and in (d) a pyramid of boxes.*

not handle collisions [AP97, CR98]. Further, acceleration based formulations suffer from indeterminacy and inconsistency [Ste00]. The velocity based formulation suffers from none of these problems and is the formulation we use in this work. [KEP05] presented a velocity based method based on limit surfaces and an approximation of momentum conservation.

The approach we present here is more simple in the sense that we reformulate the frictional problem as a system of nonlinear equations, which we then solve using a Newton method.

2. The Fischer Function Reformulation

The problem of computing contact forces can be formulated as a linear complementarity problem (LCP). However, in interactive physical simulations a different formulation is used, both will be derived shortly. In the following we will compare the two formulations, and thereafter derive our reformulation. We will keep the notation general and simple to enhance readability. Without loss of generality, we will only consider a single contact point to make the differences more visible. The contact force model is the focus of our work and we refer the reader to [Erl07] for details on the time-stepping scheme and the assembly of the matrices used.

First, a derivation of the LCP formulation for computing contact forces. Given the generalized mass matrix, M, the generalized velocity vector, u, and the generalized external force vector, F, we write the discretized Newton-Euler equation,

$$Mu - J_n^T\lambda_n - J_D^T\lambda_D = F, \tag{1}$$

where J_n is the Jacobian corresponding to normal constraints and J_D is the Jacobian corresponding to the tangential contact forces, λ_n and λ_D are the normal and frictional Lagrange multipliers. The actual discretization variables are contained within the Lagrange multipliers, this is a slight abstraction to keep the expression simple. Using (1), we isolate the generalized velocities,

$$u = M^{-1}F + M^{-1}J_n^T\lambda_n + M^{-1}J_D^T\lambda_D. \tag{2}$$

The non-penetration constraints on a velocity-based form can be written as

$$J_n u \geq 0, \tag{3a}$$
$$\lambda_n \geq 0, \tag{3b}$$
$$\lambda_n (J_n u) = 0. \tag{3c}$$

We use a scalar, β, as a measure of the most negative contact velocity along a given positive direction in the contact plane,

$$(\beta e + J_D u) \geq 0, \tag{4a}$$
$$\beta \geq 0, \tag{4b}$$

where e is a vector of ones. The contact velocity along the i^{th} direction in the contact plane is equivalent to the dot product of the i^{th} row of J_D and the generalized velocity, u. A positive span of tangent vectors ensures symmetric behavior. This is not explicit in the equations, but lies in the construction of the J_D-matrix. One can have any number of directional tangent vectors in the contact plane. A friction pyramid approximation corresponds to two orthogonal vectors in the contact plane, t_1 and t_2. In this case the Jacobian matrix J_D has four rows. The first two rows correspond to the directions t_1 and t_2 and the last two rows correspond to $-t_1$ and $-t_2$. We will bound the friction force by the friction polyhedra cone approximation,

$$\mu\lambda_n - e^T\lambda_D \geq 0 \quad \text{and} \quad \lambda_D \geq 0. \tag{5}$$

For the friction pyramid approximation one would have four components of λ_D. Finally contact velocity and frictional force are coupled through the complementarity conditions,

$$\lambda_D^T(\beta e + J_D u) = 0 \quad \text{and} \quad \beta\left(\mu\lambda_n - e^T\lambda_D\right) = 0. \tag{6}$$

Essentially, the first equation will try to pick a friction direction that is closest to the maximum dissipation direction. The last condition ensures that a non-zero friction force is found if slipping occurs. If we have a non-zero relative contact velocity then $\beta > 0$ and the last equation requires $\left(\mu\lambda_n - e^T\lambda_D\right) = 0$. This results in a friction force that lies on the boundary of the friction polyhedra cone approximation. This is in agreement with Coulomb's friction law, in the case of sliding friction. Finally, we can assemble the entire problem by substituting (2) into (3a) and (4a). Rewriting results in the matrix-vector formulation,

$$\underbrace{\begin{bmatrix} J_n M^{-1} J_n^T & J_n M^{-1} J_D^T & 0 \\ J_D M^{-1} J_n^T & J_D M^{-1} J_D^T & e \\ \mu & -e^T & 0 \end{bmatrix}}_{A_{\text{lcp}}} \underbrace{\begin{bmatrix} \lambda_n \\ \lambda_D \\ \beta \end{bmatrix}}_{x_{\text{lcp}}} + \underbrace{\begin{bmatrix} J_n M^{-1} F \\ J_D M^{-1} F \\ 0 \end{bmatrix}}_{b_{\text{lcp}}} \geq 0, \tag{7}$$

and we get the traditional LCP formulation,

$$A_{\text{lcp}} x_{\text{lcp}} + b_{\text{lcp}} \geq 0, \tag{8a}$$
$$x_{\text{lcp}} \geq 0, \tag{8b}$$
$$x_{\text{lcp}}^T (A_{\text{lcp}} x_{\text{lcp}} + b_{\text{lcp}}) = 0. \tag{8c}$$

Now we turn to the nonlinear formulation of the contact force computation. For the nonlinear case we have the Newton-Euler equations,

$$Mu - J_n^T \lambda_n - J_t^T \lambda_t = F, \tag{9}$$

where J_n is the Jacobian corresponding to normal constraints. We do not use the J_D matrix from the LCP formulation instead we use J_t, which is the Jacobian corresponding to the tangential contact forces. Since the contact plane is two dimensional, we choose to span this plane by two orthogonal unit vectors, t_1 and t_2. Any vector in this plane can be written as a linear combination of these two vectors. Thus, J_t has only two rows corresponding to the two directions. From (9) we can obtain the generalized velocities,

$$u = M^{-1}F + M^{-1} J_n^T \lambda_n + M^{-1} J_t^T \lambda_t. \tag{10}$$

Let the Lagrange multipliers $\lambda = \begin{bmatrix} \lambda_n & \lambda_t^T \end{bmatrix}^T$ and contact Jacobian $J = \begin{bmatrix} J_n & J_t \end{bmatrix}^T$, then we write the relative contact velocities as $y = \begin{bmatrix} y_n & y_t^T \end{bmatrix}^T$,

$$y = Ju = \underbrace{JM^{-1}J^T}_{A_{\text{ncp}}}\lambda + \underbrace{JM^{-1}F}_{b_{\text{ncp}}}. \tag{11}$$

To compute the frictional component of the contact force, we need a model of physical friction. We base our model on Coulomb's friction law. In one dimension Coulomb's friction law can be written as,

$$y < 0 \Rightarrow \lambda_t = \mu\lambda_n, \tag{12a}$$
$$y > 0 \Rightarrow \lambda_t = -\mu\lambda_n, \tag{12b}$$
$$y = 0 \Rightarrow -\mu\lambda_n \leq \lambda_t \leq \mu\lambda_n. \tag{12c}$$

This can be proven by algebraic manipulation.

Proof Coulomb's friction law is defined as,

$$\mu\lambda_n - \sqrt{\lambda_t^2} \geq 0, \tag{13a}$$
$$\|y\|\left(\mu\lambda_n - \sqrt{\lambda_t^2}\right) = 0, \tag{13b}$$
$$\|y\|\sqrt{\lambda_t^2} = -y\lambda_t. \tag{13c}$$

The first equation yields a maximum bound on the friction force. The second equation models sticking and slipping friction, while the last ensures that friction is opposing motion in the case of slipping friction. We perform a case-by-case analysis. If $y \neq 0$ then from (13b) we must have,

$$\mu\lambda_n - \sqrt{\lambda_t^2} = 0. \tag{14}$$

If $y > 0$ then from (13c) we must have $\lambda_t < 0$ and if $y < 0$ we have that $\lambda_t > 0$. So putting it together,

$$y < 0 \Rightarrow \lambda_t = \mu\lambda_n \quad \text{and} \quad y > 0 \Rightarrow \lambda_t = -\mu\lambda_n. \tag{15}$$

If $y = 0$ then (13b) and (13c) are trivially fulfilled and from (13a) we have,

$$-\mu\lambda_n \leq \lambda_t \leq \mu\lambda_n. \tag{16}$$

This concludes the proof. □

We split y into positive and negative components,

$$y = y^+ - y^-, \tag{17}$$

where

$$y^+ \geq 0, \quad y^- \geq 0 \quad \text{and} \quad \left(y^+\right)^T \left(y^-\right) = 0. \tag{18}$$

In case of friction we define the bounds $-l_t(\lambda) = u_t(\lambda) = \mu\lambda_n$ and for normal force $l_n(\lambda) = 0$ and $u_n(\lambda) = \infty$. Using the bounds (12), (17) and (18), yields the final NCP formulation,

$$y^+ - y^- = A_{\text{ncp}}\lambda + b_{\text{ncp}}, \tag{19a}$$
$$y^+ \geq 0, \tag{19b}$$
$$y^- \geq 0, \tag{19c}$$
$$u(\lambda) - \lambda \geq 0, \tag{19d}$$
$$\lambda - l(\lambda) \geq 0, \tag{19e}$$
$$\left(y^+\right)^T (\lambda - l(\lambda)) = 0, \tag{19f}$$
$$\left(y^-\right)^T (u(\lambda) - \lambda) = 0, \tag{19g}$$
$$\left(y^+\right)^T \left(y^-\right) = 0. \tag{19h}$$

The advantages are

- There is no need for the auxiliary variable β
- There is no need for a positive span of tangent vectors, we just use two orthogonal directions in the contact plane.
- For two tangent directions J_t is one fourth the size of J_D.

Thus the NCP formulation has a much lower memory footprint. To make the connection to the LCP formulation more clear one could define,

$$J_D u = \begin{bmatrix} J_t u \\ -J_t u \end{bmatrix}, \lambda_D = \begin{bmatrix} \lambda_t^+ \\ \lambda_t^- \end{bmatrix}, \quad \text{and} \quad \beta = \max_j |y_j|. \quad (20)$$

The main problem with the nonlinear formulation as presented, is that t_1 and t_2 are independent. The disadvantage is that it solves the friction problem as two decoupled one-dimensional Coulomb friction models. This results in an outer friction pyramid approximation, where the LCP formulation uses an inner diamond shape approximation. Thus the NCP formulation overestimates the magnitude of the friction force whereas the LCP formulation underestimates the magnitude of the friction force. Using the NCP formulation, the corner directions of the friction pyramid are favored in case of unaligned contact velocities. The LCP formulation favors the tangent direction that best approximates the direction of maximum dissipation. The NCP formulation is attractive for interactive simulations where performance is of greater importance than accuracy. In practice the problem of accuracy can be remedied by choosing t_1 parallel to the sliding direction. Using such a heuristic would make both the LCP and NCP formulations accurate in the case of sliding friction.

We have shown that the formulation (12) is equivalent to the coupled complementarity problems given in (19). We can reformulate those complementarity problems using a Fischer function. The Fischer function, $\phi : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$ is defined as,

$$\phi(a,b) = \sqrt{a^2 + b^2} - a - b. \quad (21)$$

Notice that solutions of the complementarity problem corresponds to the roots of the Fischer function,

$$a \geq 0, \quad b \geq 0, \quad a \cdot b = 0, \quad \text{iff} \quad \phi(a,b) = 0. \quad (22)$$

This allows us to rewrite a complementarity problem into a root search problem, which again allows us to compute solutions using a Newton method. The real valued function $\phi(a,b)$ is trivially extended to a vector function. Using the Fischer function, (19) can be written as,

$$\phi\left(y^+, \lambda - l(\lambda)\right) = 0, \quad (23a)$$

$$\phi\left(y^-, u(\lambda) - \lambda\right) = 0. \quad (23b)$$

Observe, for any arbitrary positive scalar k, (23a) is equivalent to,

$$\phi\left(ky^+, \lambda - l(\lambda)\right) = 0. \quad (24)$$

Further, there always exists a positive scalar k such that,

$$ky^+ = \phi\left(y^- - y^+, u(\lambda) - \lambda\right). \quad (25)$$

This is proven by a case-by-case analysis. If $y^+ = 0$ then (25) reduces to (23b) which trivially holds for any value of k. If $y_i^+ > 0$, then we must have $y_i^- = 0$ and (25) reduces to,

$$ky_i^+ = \underbrace{\sqrt{\left(y_i^+\right)^2 + \left(u(\lambda)_i - \lambda_i\right)^2} - \left(u(\lambda)_i - \lambda_i\right)}_{c} + y_i^+. \quad (26)$$

Using the triangle inequality, we always have $c > 0$ so $k = c/y^+ + 1 > 1$. Thus, for any given $y^+ > 0$ we can always find a $k > 1$ such that (25) holds. We can now substitute (25) into (24) and obtain the Fischer reformulation of our problem,

$$\phi\left(\lambda - l(\lambda), \phi\left(-y, u(\lambda) - \lambda\right)\right) = 0. \quad (27)$$

Notice our formulation differs from the mixed complementarity problem formulation [Bil95]. We are using variable bounds and not fixed bounds. Our derivation do not rely on explicitly enforcing $u(\lambda) \geq l(\lambda)$ at all times, instead this holds implicitly for any solution of the problem.

3. The Nonsmooth Newton Method

Numerical solutions to complementarity problems can be computed using well known iterative methods. In [CPS92], the authors describe matrix splitting methods similar to well known relaxation methods for linear systems: Jacobi, Gauss–Seidel, and successive over–relaxation (SOR). These methods enforce the complementarity constraints by doing a projection following a relaxation. Projected Gauss–Seidel solvers have previously been used [Mor99, Jea99] although in a blocked version. Krylov subspace methods like the Conjugate Projected Gradient (CPG) can be used in a similar manner [RAD05] or together with an active set method [Mur88]. The CPG has quadratic convergence rate for frictionless contacts, but erratic convergence rate when friction is considered [RA04]. Two other types of iterative algorithms are based on Newton and Interior Point (IP) methods. The theoretical convergence rates of the Newton and IP methods are quadratic, which is a clear improvement over relaxation schemes. Complexity can be reduced even further if one considers a multilevel preconditioner for the Newton equation [Ort07]. Direct methods are costly and often limited to LCP formulations. When using incremental matrix factorization, Lemke's and Keller's methods need n iterations resulting in a total complexity of $O(n^3)$ [Lac03]. We choose the Newton method due to the quadratic convergence rate and do not consider IP methods due to the nature of root search problems.

In the following, we will describe our Newton based method, which we have named the Fischer–Newton (FN) method to reflect the use of a Fischer reformulation.

We will now extend our model to support multiple contacts. Let the normal and frictional Lagrange multipliers of the i^{th} contact point $\lambda_i = \begin{bmatrix}\lambda_{n,i} & \lambda_{t_1,i} & \lambda_{t_2,i}\end{bmatrix}^T$. For the i^{th} contact point we have from (27),

$$\phi_i(\lambda) = \phi\left(\lambda_i - l_i(\lambda_i), \phi\left(-y_i, u_i(\lambda_i) - \lambda_i\right)\right) = 0, \qquad (28)$$

where $l_i(\lambda_i) = \begin{bmatrix}0 & -\mu_i\lambda_{n,i} & -\mu_i\lambda_{n,i}\end{bmatrix}^T$ and $u_i(\lambda_i) = \begin{bmatrix}\infty & \mu_i\lambda_{n,i} & \mu_i\lambda_{n,i}\end{bmatrix}^T$ are the normal and frictional lower and upper bounds. The relative contact point velocity y_i is an affine function of the agglomerated Lagrange multiplier vector $\lambda = \begin{bmatrix}\lambda_1^T & \lambda_2^T & \dots & \lambda_N^T\end{bmatrix}^T$. Now we can agglomerate all contact problems (28) into one vector valued problem, given by the vector valued function $\Phi : \mathbb{R}^n \to \mathbb{R}^n$ with $n = 3N$,

$$\Phi(\lambda) = \begin{bmatrix}\phi_1(\lambda) \\ \vdots \\ \phi_N(\lambda)\end{bmatrix}, \qquad (29)$$

which is almost everywhere Frechét differentiable. We define the set Ω_Φ to be the set of all λ for which Φ is not Frechét differentiable. The B-subdifferential of Φ at λ is,

$$\partial_B\Phi = \{H : \mathbb{R}^n \to \mathbb{R}^n | H = \lim J_\Phi(\lambda_i) : \lambda_i \to \lambda, \lambda \notin \Omega_\Phi\}, \qquad (30)$$

where J_Φ is the usual Jacobian. Using the B-subdifferential, the generalized Newton equation is,

$$H_k\Delta\lambda_k = -\Phi(\lambda_k), \qquad (31)$$

where H_k is any nonsingular element in $\partial_B\Phi$ [Bil95]. The Newton update is then

$$\lambda_{k+1} = \lambda_k + \Delta\lambda_k. \qquad (32)$$

The convergence behavior of the numerical method is described by the following Theorem.

Theorem 3.1 (Qi & Sun [QS98, Theorem 2.2]) Suppose that $\Phi(\lambda^*) = 0$ and that all $H \in \partial_B\Phi(\lambda^*)$ are nonsingular. Then the generalized Newton method (31) is Q-superlinearly convergent in a neighborhood of λ^* if Φ is semismooth at λ^*, and quadratically convergent at λ^* if Φ is strongly semismooth at λ^*.

We know from [Bil95, Theorem 3.2.8] that Φ is strongly semismooth everywhere when l and u are fixed bounds. This is easily extended to hold for our version, where the bounds are functions of λ. The only limitation we must impose on $l(\lambda)$ and $u(\lambda)$ is that they are both semismooth functions. This follows since any sum, product or composite of semismooth functions is also semismooth [Bil95]. Figure 2 shows that quadratic convergence is not only a theoretical property but is achievable in practise as well. The test case used for Figure 2 is a small 6×6 problem. It takes 0.0033 seconds on average to compute the solution. For interactive usage – 30 frames per second – using 10 % of the time for contact force solving [YFR06], this restricts the FN method to problems of small dimensions.

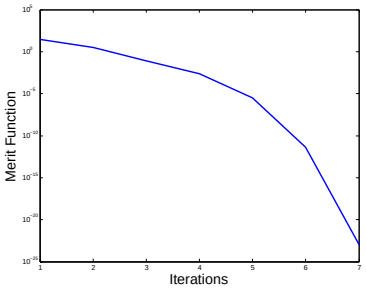


Figure 2: *The Fischer–Newton method on a well behaved problem displaying quadratic convergence. Average time used on solution is 0.0033 seconds for this test case.*

Solving the generalized Newton equation exact is an operation that has a time complexity of $O(n^3)$, a cheaper alternative should improve the overall performance of the FN method.

Quasi Nonlinear Conjugate Gradient Method for Interactive Frame Rates The conjugate gradient (CG) method is the usual choice for iteratively solving equation systems like (31) approximately. It is guaranteed to converge to the solution in at most n iterations. Unfortunately, this only holds for symmetric and positive definite matrices. We cannot guarantee that this holds for H_k. However, it is appealing to use one of the other CG like methods due to the nice properties. The biconjugate gradient method does not require symmetry of the matrix, instead it uses the matrix and its transposed to compute a search direction. For this reason it yields the same result as the CG for symmetric matrices, but at twice the cost. For interactive usage the extra cost is undesired, so a third CG like method is considered. The standard CG method can be considered as the minimization of a quadratic function,

$$\min_\lambda \underbrace{\frac{1}{2}\lambda^T H_k\lambda - \Phi(\lambda)^T\lambda}_{\gamma(\lambda)}, \qquad (33)$$

where the gradient is $\nabla\gamma(\lambda) = H_k\lambda - \Phi(\lambda)$ when H_k is symmetric. The gradient is used in the computation of the new search direction. The nonlinear conjugate gradient (NCG) method makes no such assumption, instead it just minimizes the general function $f(\lambda)$. To do this we need the gradient $\nabla f(\lambda)$ for the computation of new search directions. However, such a gradient is computationally expensive and the aim of interactivity forces us to take the less expensive choice. Instead of using the exact ∇f we use a finite difference approximation,

$$\nabla f(\lambda) \approx \frac{f(\lambda + h) - f(\lambda)}{h}. \qquad (34)$$

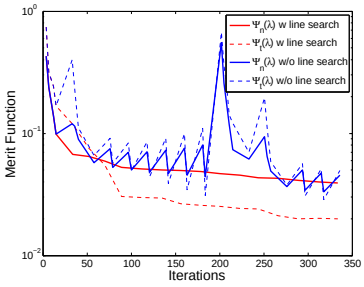


Figure 3: *Motivation for including a line search method. The test case is a pyramid configuration, having 103 contact points resulting in 309 variables. Using line search, a smoother convergence is achieved. The frictional component has the additional benefit of an improved accuracy.*

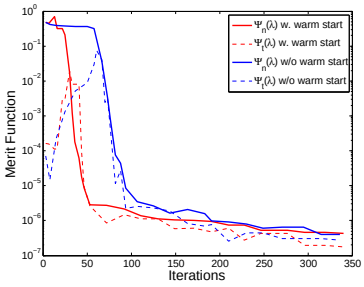


Figure 4: *Two convergence plots for the large mass ratio test case showing how warm starting the Fischer–Newton method provides improved convergence.*

Thus, our method can be perceived as a quasi Fletcher–Reeves [NW99] method where we use finite difference approximations for the gradient and a line-search similar to the exact minimizer used in the Conjugate Gradient method.

Globalization using Line Search The Newton method is not globally convergent from an arbitrary starting point. It has to be initialized within a small enough neighborhood of the solution. To achieve global convergence, a line search method is often used. Experiments indicate that it is beneficial to use a line search method when including friction, see Figure 3. We use a backtracking line search with an Armijo condition to ensure sufficient decrease and that the chosen step length is not too small [NW99]. The line search uses the natural merit function of $\Phi(\lambda)$ as a measure of convergence. The natural merit function is defined as,

$$\Psi(\lambda) = \frac{1}{2}||\Phi(\lambda)||^2. \tag{35}$$

The gradient is,

$$\nabla\Psi(\lambda) = H_k^T\Phi(\lambda). \tag{36}$$

The Armijo condition is used as a termination criteria for the line search method. Applied to $\Psi(\lambda)$, the Armijo condition is,

$$\Psi(\lambda+\Delta\lambda) \leq \Psi(\lambda) + c\alpha\nabla\Psi(\lambda)^T\Delta\lambda, \tag{37}$$

where $c \in (0,1)$. The object of the line search method is to find an α such that (37) is satisfied. To avoid computing $\nabla\Psi(\lambda)$ we insert the definition from (36) into (37) and make a substitution using (31),

$$\Psi(\lambda+\Delta\lambda) \leq (1-2\alpha c)\Psi(\lambda). \tag{38}$$

The step length is used when updating λ_{k+1} such that

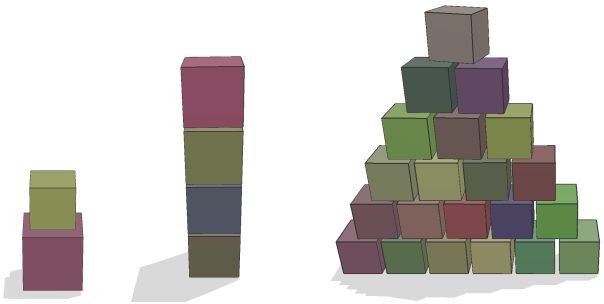
$$\lambda_{k+1} = \lambda_k + \alpha\Delta\lambda_k. \tag{39}$$

Semi-Staggered Warm Starting to Accelerate Convergence The frictional upper and lower bounds depend on the magnitude of the normal forces. Intuition suggests that given a good estimate of the normal forces, the computation of frictional forces will be more feasible. The idea is similar to a staggered approach [KSJP08]. However, instead of continuously iterating, we simply start the FN method with the frictional components disabled and use an absolute threshold termination criteria. Once convergence is detected, the frictional components are enabled and the full contact force problem is solved until convergence or a maximum allowed iteration count is reached. Figure 4 illustrates the added convergence benefit from this semi-staggered warm starting technique.

4. Newton Method in Action

We have constructed three test cases to test different properties of the FN method. All three test cases use simple box geometries to aid the visual detection of physical anomalies, which might be hard to detect if more complex geometries were used.
Further the Jacobian – and subsequently the Newton equation – inherits a lot of the properties of the A_{ncp} matrix. We expect that our method struggles when applied to problems where A_{ncp} is badly conditioned. The three test cases are therefore designed to "stress" the A_{ncp} matrix.

Large Mass Ratios This is a small configuration consisting of only three objects, two boxes and a floor. A heavy box is placed on top of a lighter box. The desired effect is for the boxes to remain still with no additional movement after the initial settlement. This setup results in a A_{ncp} matrix which has an unfortunate distribution of eigenvalues, as seen in Figure 7(a). This gives an ill-conditioned A_{ncp} matrix, making the Newton equation singular. In addition to "stressing" our method, this setup is notoriously difficult for the PGS method to handle. The PGS method has a tendency



(a) (b) (c)

Figure 5: *The three test cases (a) A heavy box placed on top of a light box (b) A stack of equal sized boxes of equal density (c) A dense structured stack of equal sized boxes of equal density.*

to undershoot the magnitudes of the contact forces, resulting in severe penetrations.

Sparse Structure For the sparse structure, a stack of identical boxes are simulated. This setup poses some of the same difficulties as the large mass ratio setup. The bottommost box suffers from the accumulated effect of the remaining boxes on top, acting as one large mass. Furthermore, all contact normals are parallel, which results in an over-determined system.

Dense Structure When the complexity of the physical system increases, people become less competent at noticing dynamic anomalies [ODGK03]. Thus structured stacking is very helpful for visual confirmation of complex configurations. The dense structure covers the above mentioned difficulties. Further this setup tests scalability.

4.1. The Qualities of the Newton Method

Figure 6 shows the fill patterns for the three test cases, visualized by a gray scale image. It confirms that the A_{ncp} is symmetric and that sparsity increases with configuration size. The diagonal entries appear to be larger than off diagonal entries, although not enough for the matrix to be diagonally dominant. Studying the eigenvalue spectrums in Figure 7, reveals multiplicity greater than one for some non-zero eigenvalues and a large number of zero eigenvalues, roughly 60 % for all three cases. The poor eigenvalue properties are a direct result from over-determined systems and large mass ratios. By this analysis, we expect non-unique solutions to exist and to observe convergence to local minima.

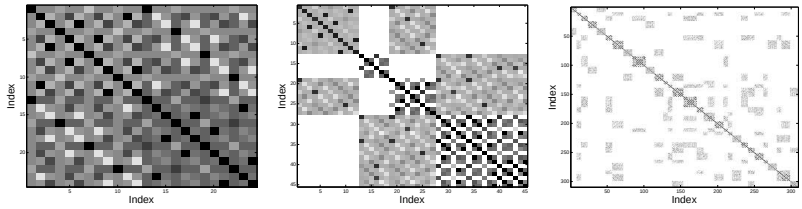
The plots in Figure 8 show the convergence properties of the natural merit function (35) when using the FN and PGS methods. The cost of FN iterations are presented in units of PGS iterations in order to aid the comparison. The convergence is divided into normal and frictional components. The division helps to visualize the effect of the semi-staggered warm start heuristic and whether convergence problem are due to frictional or normal components. Figure 8(a) shows an improvement in both convergence and accuracy in several orders of magnitude, when using the FN method. As the complexity of the scene increases the benefit of using the FN method decreases. In Figure 8(b) the FN method remains superior on both convergence and accuracy for the frictional contribution. When given enough iterations, the PGS method almost achieves the same level of accuracy. Figure 8(c) shows the convergence results for the most complex test case. Both methods converge to local minima for this test case, but in terms of accuracy the minimizer chosen by the FN method is poorer than the one chosen by the PGS method. However, the FN method converged using fewer iterations than PGS.

4.2. Interactive Setting

Figure 10 shows image sequences of a comparison of the handling of large mass ratios, the FN method versus the PGS method. As the images show, the fidelity and consistency break down for the PGS method, where the FN method produces plausible results. In Figure 9 still frames from a larger configuration are shown, the FN method is used to compute contact forces. The FN method is still capable of delivering interactive frame rates for small to medium sized configurations.

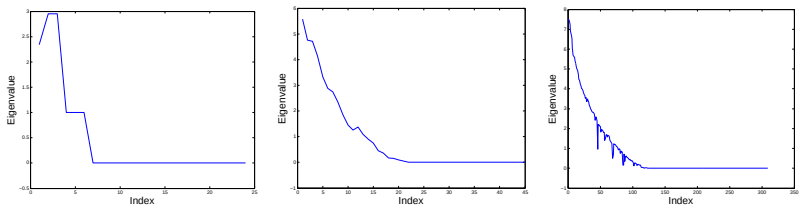
5. Conclusion and Discussion

We have presented a novel Fischer reformulation of a contact force problem for interactive physical simulation. To solve the problem we have developed a novel Fisher–Newton (FN) type method. We have evaluated and compared the new method against the Projected Gauss–Seidel (PGS) method.



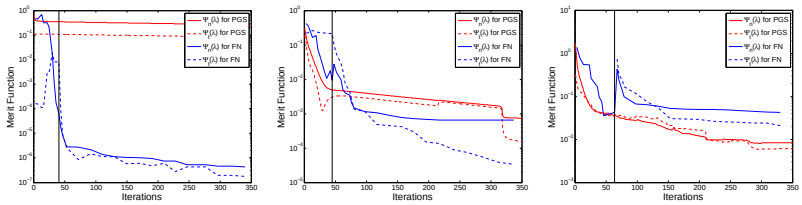
(a) Large Mass Ratio (b) Sparse Structure (c) Dense Structure

Figure 6: *Fill patterns for the A_{ncp} matrices of the three test cases. The color corresponds to magnitude of values, white is zero and black is* $\max(A_{ncp})$*. Observe mostly black diagonal entries and increasing sparsity as configuration size grows.*



(a) Large Mass Ratio (b) Sparse Structure (c) Dense Structure

Figure 7: *Distribution of eigenvalues in A_{ncp} for the three test configurations. Observe a large amount of zero-valued eigenvalues, multiplicity and poor distribution.*



(a) Large Mass Ratio (b) Sparse Structure (c) Dense Structure

Figure 8: *Comparative convergence plots of the Fischer–Newton method versus the Projected Gauss–Seidel method. Notice the improved convergence properties of the Fischer–Newton method for the cases of large mass ratios and sparse structure. The vertical line indicates the point where friction is enabled in the warm start approach.*

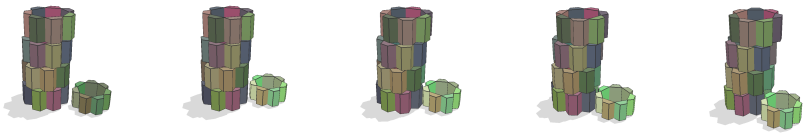
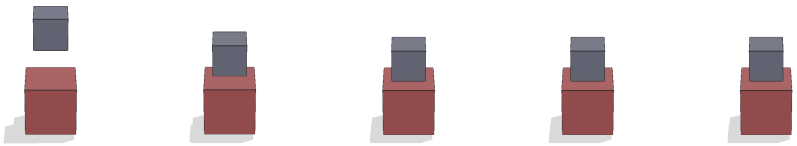


Figure 9: *Still frames from an interactive simulation of a medium sized configuration, with approximately 280 contact points yielding 840 variables. The Fischer–Newton method is implemented in Java and uses 6 milliseconds on average per frame. The program utilizes one core on a 2.1 GHz CPU Duo Core.*



(a) The Fischer–Newton method. Newton iteration limit 10, sub-system iteration limit 15, line-search iteration limit 7

(b) The Projected Gauss–Seidel method. 150 iterations per frame

Figure 10: *Still frames from two comparative movie strips showing the success of the Fischer–Newton method in opposition to the failure of the Projected Gauss–Seidel method. Captures was done every 30 frames. Even with a 150 iterations per frame, the Projected Gauss-Seidel method fails to support the upper heavy box.*

The FN method was able to compute solutions comparable to the PGS method, both in terms of solution accuracy, and in terms of computational effort. Regarding small sized configurations, the FN method demonstrated superior abilities in handling large mass ratios. To a lesser extend, the FN method showed good abilities when applied to smaller stacks of 3-4 objects. The FN method proved to have convergence problems when applied to setup where large friction forces were present. Inclusion of friction results in greater amount of over-determinacy. This over-determinacy is inherited by the Newton equation. We speculate that the increased over-determinacy is the cause of the inability to handle more complex and frictionally dependent problems. One may apply matrix damping to counter the numerical problems coming from over-determinacy. However, it results in a damping of the solution and little is gained in accuracy compared to using the computationally cheaper PGS method. Thus, future work on Newton methods for contact force problems should focus on how to deal efficiently with the over-determinacy in the contact force problem.

References

[AP97] ANITESCU M., POTRA F. A.: Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlinear Dynamics. An International Journal of Nonlinear Dynamics and Chaos in Engineering Systems 14*, 3 (1997), 231–247.

[Bar94] BARAFF D.: Fast contact force computation for nonpenetrating rigid bodies. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1994), ACM, pp. 23–34.

[Bil95] BILLUPS S. C.: *Algorithms for complementarity problems and generalized equations*. PhD thesis, University of Wisconsin at Madison, Madison, WI, USA, 1995.

[CPS92] COTTLE R., PANG J.-S., STONE R. E.: *The Linear Complementarity Problem*. Computer Science and Scientific Computing. Academic Press, February 1992.

[CR98] CHATTERJEE A., RUINA A.: A new algebraic rigid body collision law based on impulse space considerations. *Journal of Applied Mechanics 65*, 4 (1998), 939–951.

[Erl07] ERLEBEN K.: Velocity-based shock propagation for multibody dynamics animation. *ACM Trans. Graph. 26*, 2 (2007), 12.

[Fea98] FEATHERSTONE R.: *Robot Dynamics Algorithms*, second printing ed. Kluwer Academic Publishers, 1998.

[GBF03] GUENDELMAN E., BRIDSON R., FEDKIW R.: Nonconvex rigid bodies with stacking. *ACM Trans. Graph. 22*, 3 (2003), 871–878.

[Hah88] HAHN J. K.: Realistic animation of rigid bodies. In *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1988), ACM, pp. 299–308.

[Jea99] JEAN M.: The non-smooth contact dynamics method. *Computer Methods in Applied Mechanics and Engineering 177*, 3–4 (July 1999), 235–257.

[KEP05] KAUFMAN D. M., EDMUNDS T., PAI D. K.: Fast frictional dynamics for rigid bodies. *ACM Trans. Graph. 24*, 3 (2005), 946–956.

[KSJP08] KAUFMAN D. M., SUEDA S., JAMES D. L., PAI D. K.: Staggered projections for frictional contact in multibody systems. *ACM Trans. Graph. 27*, 5 (2008), 1–11.

[Lac03] LACOURSIERE C.: Splitting methods for dry frictional contact problems in rigid multibody systems: Preliminary performance results. In *The Annual SIGRAD Conference* (November 2003), Ollila M., (Ed.), no. 10 in Linkøping Electronic Conference Proceedings.

[Mir96] MIRTICH B. V.: *Impulse-based dynamic simulation of rigid body systems*. PhD thesis, University of California, Berkeley, 1996.

[Mor99] MOREAU J. J.: Numerical aspects of the sweeping process. *Computer Methods in Applied Mechanics and Engineering 177*, 3–4 (July 1999), 329–349.

[MS01] MILENKOVIC V. J., SCHMIDL H.: Optimization-based animation. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), ACM, pp. 37–46.

[MS04] MILENKOVIC V. J., SCHMIDL H.: A fast impulsive contact suite for rigid body simulation. *IEEE Transactions on Visualization and Computer Graphics 10*, 2 (2004), 189–197.

[Mur88] MURTY K. G.: *Linear Complementarity, Linear and Nonlinear Programming*. Helderman-Verlag, 1988.

[MW88] MOORE M., WILHELMS J.: Collision detection and response for computer animation. In *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1988), ACM, pp. 289–298.

[NW99] NOCEDAL J., WRIGHT S. J.: *Numerical optimization*. Springer Series in Operations Research. Springer-Verlag, New York, 1999.

[ODGK03] O'SULLIVAN C., DINGLIANA J., GIANG T., KAISER M. K.: Evaluating the visual fidelity of physically based animations. *ACM Trans. Graph. 22*, 3 (2003), 527–536.

[Ort07] ORTIZ R.: *Newton/AMG algorithm for solving complementarity problems arising in rigid body dynamics with frictional impacts*. PhD thesis, University of Iowa, July 2007.

[QS98] QI L., SUN D.: Nonsmooth equations and smoothing Newton methods. *Progress in Optimization: Contributions from Australasia, A. Eberhard, B. Glover, R. Hill and D. Ralph eds., Kluwer Academic Publisher, Boston* (1998).

[RA04] RENOUF M., ALART P.: Gradient type algorithms for 2d/3d frictionless/frictional multicontact problems. In *Proceedings of European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS)* (2004).

[RAD05] RENOUF M., ACARY V., DUMONT G.: Comparison of algorithms for collisions, contact and friction in view of real-time applications in multibody dynamics. In *Proceedings (International Conference on Advances in Computational Multibody Dynamics. ECCOMAS Thematic Conference)* (2005).

[RKC03] REDON S., KHEDDAR A., COQUILLART S.: Gauss least constraints principle and rigid body simulations. In *In proceedings of IEEE International Conference on Robotics and Automation* (2003).

[ST96] STEWART D. E., TRINKLE J. C.: An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *International Journal of Numerical Methods in Engineering 39*, 15 (1996), 2673–2691.

[Ste00] STEWART D. E.: Rigid-body dynamics with friction and impact. *SIAM Review 42*, 1 (2000), 3–39.

[TTP01] TRINKLE J. C., TZITZOUTIS J., PANG J.-S.: Dynamic multi-rigid-body systems with concurrent distributed contacts: Theory and examples. *Philosophical Trans. on Mathematical, Physical, and Engineering Sciences 359*, 1789 (December 2001), 2575–2593.

[YFR06] YEH T. Y., FALOUTSOS P., REINMAN G.: Enabling real-time physics simulation in future interactive entertainment. In *Sandbox '06: Proceedings of the 2006 ACM SIGGRAPH symposium on Videogames* (New York, NY, USA, 2006), ACM, pp. 71–81.

B WSCG 2010 Article

I have, in collaboration with Morten Poulsen and Kenny Erleben, submitted a paper for the WSCG 2010 conference on Computer Graphics, Visualization and Computer Vision. The acceptance decision is pending, the answer is expected in early December 2009. The paper is a comparison of heuristics for improving convergence rates of the projected Gauss–Seidel method.

Heuristic Convergence Rate Improvements of the Projected Gauss–Seidel Method for Frictional Contact Problems.

Morten Poulsen
Department of Computer Science,
University of Copenhagen, Denmark
mrtn@diku.dk

Sarah Niebe
Department of Computer Science,
University of Copenhagen, Denmark
niebe@diku.dk

Kenny Erleben
Department of Computer Science,
University of Copenhagen, Denmark
kenny@diku.dk

ABSTRACT

In interactive physical simulation, contact forces are applied to prevent rigid bodies from penetrating and control slipping between bodies. Accurate contact force determination is a computationally hard problem. Thus, in practice one trades accuracy for performance. The result is visual artifacts such as viscous or damped contact response. In this paper, we present heuristics for improving performance for solving contact force problems in interactive rigid body simulation. We formulate the contact force problem as a nonlinear complementarity problem, and discretize the problem using a splitting method and a minimum map reformulation. The resulting model is called the Projected Gauss–Seidel method. Quantitative research results are presented and can be used as a taxonomy for selecting a suitable heuristic when using the Projected Gauss–Seidel method.

Keywords: Nonlinear Complementarity Problem, Contact Forces, Convergence Rate, Projected Gauss–Seidel.

1 SLOW CONVERGENCE RATES

Most open source software for interactive real time rigid body simulation uses the Projected Gauss–Seidel (PGS) method for computing contact forces. This includes the two most popular open source simulators Bullet and Open Dynamics Engine. However, the PGS method is not always satisfactory as it suffers from two major problems: a linear convergence rate [4] and inaccurate friction forces in stacks [10]. The linear convergence rate results in viscous motion at contacts, as well as loss of high frequency effects. The viscous appearance results in a delayed contact response which reduces plausibility [16]. Improving convergence might lead to increased animation quality and higher fidelity. The prospect of improving a state-of-the-art method – the PGS method – has motivated this study of heuristics, aimed at improving convergence.

With this paper, we present a rigorous novel mathematical derivation of the PGS method as well as experience gained from quantitative research results. The results can be used as a taxonomy for selecting a suitable heuristic when using the PGS method.

2 PREVIOUS WORK

Rigid body simulation was introduced to the graphics community in the late 1980's [9, 13], using penalty based and impulse based approaches to describe physical interactions. Penalty based simulation is not easily adopted to different simulations. Mirtich [12] presented an extended and improved impulse based formulation, however stacking was a problem and it suffered from creeping. These problems has since been rectified [8]. Constraint based simulation [2] has received much attention as an alternative to penalty based and impulse based simulation. Constraint based simulation can be divided into two groups: maximal coordinate and minimal coordinate methods [7]. The focus of this paper is maximal coordinate methods, which are dominated by complementarity formulations. Alternatives to complementarity formulations are based on kinetic energy [11] and motion space [17]. However, the former solves a more general problem and is not attractive for performance reasons, the latter does not include frictional forces.

Complementarity formulations are either acceleration based formulations [21] or velocity based formulations [20]. Acceleration based formulations can not handle collisions [1], in addition they suffer from indeterminacy and inconsistency [19]. Velocity based formulations suffer from none of these problems, for this reason we use a velocity based formulation for this paper. The approach we present here is based on a reformulation of the frictional problem as a nonlinear complementarity problem. This results in a slightly inaccurate model with relatively few variables to solve for. This makes it advantageous in interactive simulations from a performance viewpoint.

The state-of-the-art method for solving complementarity formulations in interactive rigid body simulation, is the projected Gauss–Seidel method. To our knowledge, no mathematical derivation of the PGS method for interactive rigid body simulation has been presented

WSCG 2010 conference proceedings, ISBN XX-YYYYYY-Z-Z
WSCG'2010, February 1 – February 4, 2010
Plzen, Czech Republic.

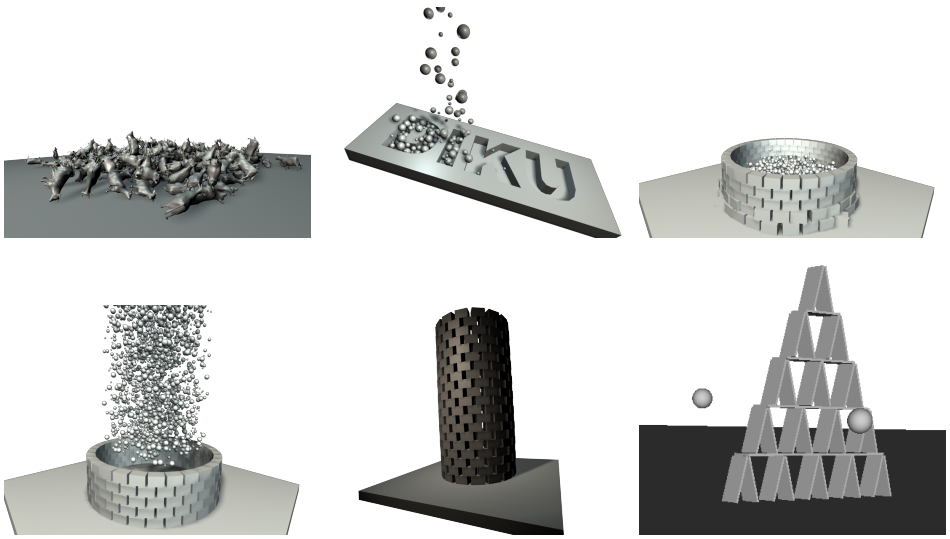


Figure 1: Renderings of selected setups from the test data sets used to examine the performance achieved by using different heuristics for the projected Gauss–Seidel method. The setups varies from 100-10000 interacting rigid bodies in varying degrees of structured configurations.

in the Computer Graphics literature, nor has any studies on its convergence behavior or rates been published.

3 THE NONLINEAR COMPLEMENTARITY PROBLEM FORMULATION

The frictional contact force problem can be stated as a linear complementarity problem (LCP) [20]. However, a slightly different formulation is used in interactive physical simulations, we will derive this formulation. Without loss of generality, we will only consider a single contact point. The focus of this paper is on the contact force model, so the time stepping scheme and matrix layouts are based on the velocity-based formulation in [6]. We have the Newton–Euler equations,

$$\mathbf{M}\mathbf{u} - \mathbf{J}_n^T \lambda_n - \mathbf{J}_t^T \lambda_t = \mathbf{F}, \tag{1}$$

where $\mathbf{J}_n$ is the Jacobian corresponding to normal constraints and $\mathbf{J}_t$ is the Jacobian corresponding to the tangential contact impulses. $\mathbf{M}$ is the generalized mass matrix and $\mathbf{u}$ is the generalized velocity vector. We wish to solve for $\mathbf{u}$ in order to compute a position update. For clarity and readability we have, without loss of generality, abstracted the discretization details within the Lagrange multipliers λ_n, λ_t and generalized external impulses $\mathbf{F}$. Since the contact plane is two dimensional, we span this plane by two orthogonal unit vectors, t_1 and t_2. Any vector in this plane can be written as a linear combination of these two vectors. Thus, $\mathbf{J}_t$ has only two rows corresponding to the two directions. From (1) we can obtain the generalized velocities,

$$\mathbf{u} = \mathbf{M}^{-1}\mathbf{F} + \mathbf{M}^{-1}\mathbf{J}_n^T \lambda_n + \mathbf{M}^{-1}\mathbf{J}_t^T \lambda_t. \tag{2}$$

Let the Lagrange multipliers $\lambda = \begin{bmatrix}\lambda_n & \lambda_t^T\end{bmatrix}^T$ and contact Jacobian $\mathbf{J} = \begin{bmatrix}\mathbf{J}_n & \mathbf{J}_t\end{bmatrix}^T$, then we write the relative contact velocities $\mathbf{y} = \begin{bmatrix}y_n & \mathbf{y}_t^T\end{bmatrix}^T$ such that,

$$\mathbf{y} = \mathbf{J}\mathbf{u} = \underbrace{\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T}_{\mathbf{A}}\lambda + \underbrace{\mathbf{J}\mathbf{M}^{-1}\mathbf{F}}_{\mathbf{b}}. \tag{3}$$

To compute the frictional component of the contact impulse, we need a model of friction. We base our model on Coulomb's friction law. In one dimension, Coulomb's friction law can be written as [2],

$$y < 0 \Rightarrow \lambda_t = \mu\lambda_n, \tag{4a}$$
$$y > 0 \Rightarrow \lambda_t = -\mu\lambda_n, \tag{4b}$$
$$y = 0 \Rightarrow -\mu\lambda_n \leq \lambda_t \leq \mu\lambda_n. \tag{4c}$$

For the full contact problem, we split $\mathbf{y}$ into positive and negative components,

$$\mathbf{y} = \mathbf{y}^+ - \mathbf{y}^-, \tag{5}$$

where

$$\mathbf{y}^+ \geq 0, \quad \mathbf{y}^- \geq 0 \quad \text{and} \quad \left(\mathbf{y}^+\right)^T \left(\mathbf{y}^-\right) = 0. \tag{6}$$

For a frictional contact problem, we define the bounds $-l_t(\lambda) = u_t(\lambda) = \mu\lambda_n$ and for normal impulse $l_n(\lambda) = 0$ and $u_n(\lambda) = \infty$. Combining the bounds with (4), (5)

and (6), we reach the final nonlinear complementarity problem (NCP) formulation,

$$\mathbf{y}^+ - \mathbf{y}^- = \mathbf{A}\lambda + \mathbf{b}, \tag{7a}$$
$$\mathbf{y}^+ \geq 0, \tag{7b}$$
$$\mathbf{y}^- \geq 0, \tag{7c}$$
$$u(\lambda) - \lambda \geq 0, \tag{7d}$$
$$\lambda - l(\lambda) \geq 0, \tag{7e}$$
$$\left(\mathbf{y}^+\right)^T (\lambda - l(\lambda)) = 0, \tag{7f}$$
$$\left(\mathbf{y}^-\right)^T (u(\lambda) - \lambda) = 0, \tag{7g}$$
$$\left(\mathbf{y}^+\right)^T \left(\mathbf{y}^-\right) = 0, \tag{7h}$$

where $l(\lambda) = [l_n(\lambda) \quad \mathbf{l}_t(\lambda)]^T$ and $u(\lambda) = [u_n(\lambda) \quad \mathbf{u}_t(\lambda)]^T$. The advantage of the NCP formulation is a much lower memory footprint than for the LCP formulation. The disadvantage is solving the friction problem as two decoupled one dimensional Coulomb friction models.

4 THE PROJECTED GAUSS–SEIDEL METHOD

The following is a derivation of the PGS method for solving the frictional contact force problem, stated as the NCP (7). Using a minimum map reformulation, the i^{th} component of (7) can be written as

$$(\mathbf{A}\lambda + \mathbf{b})_i = \mathbf{y}_i^+ - \mathbf{y}_i^-, \tag{8a}$$
$$\min(\lambda_i - l_i, \mathbf{y}_i^+) = 0, \tag{8b}$$
$$\min(u_i - \lambda_i, \mathbf{y}_i^-) = 0. \tag{8c}$$

where $l_i = l_i(\lambda)$ and $u_i = u_i(\lambda)$. Note, when $\mathbf{y}_i^- > 0$ we have $\mathbf{y}_i^+ = 0$ which in turn means that $\lambda_i - l_i \geq 0$. In this case, (8b) is equivalent to

$$\min(\lambda_i - l_i, \mathbf{y}_i^+ - \mathbf{y}_i^-) = -(\mathbf{y}^-)_i. \tag{9}$$

If $\mathbf{y}_i^- = 0$ then $\lambda_i - l_i = 0$ and complementarity constraint (8b) is trivially satisfied. Substituting (9) for $\mathbf{y}_i^-$ in (8c) yields,

$$\min(u_i - \lambda_i, \max(l_i - \lambda_i, -(\mathbf{y}^+ - \mathbf{y}^-)_i)) = 0. \tag{10}$$

This is a more compact reformulation than (7) and eliminates the need for auxiliary variables $\mathbf{y}^+$ and $\mathbf{y}^-$. By adding λ_i we get a fixed point formulation

$$\min(u_i, \max(l_i, \lambda_i - (\mathbf{A}\lambda + \mathbf{b})_i)) = \lambda_i. \tag{11}$$

We introduce the splitting $\mathbf{A} = \mathbf{M} - \mathbf{N}$ and an iteration index k. Then we define $\mathbf{c}^k = \mathbf{b} - \mathbf{N}\lambda^k$, $l^k = l(\lambda^k)$ and $u^k = u(\lambda^k)$. Using this we have

$$\min(u_i^k, \max(l_i^k, (\lambda^{k+1} - \mathbf{M}\lambda^{k+1} - \mathbf{c}^k)_i)) = \lambda_i^{k+1}. \tag{12}$$

When $\lim_{k\to\infty} \lambda^k = \lambda^*$ then (12) is equivalent to (7). Next we perform a case-by-case analysis. Three cases are possible,

$$(\lambda^{k+1} - \mathbf{M}\lambda^{k+1} - \mathbf{c}^k)_i < l_i \Rightarrow \lambda_i^{k+1} = l_i, \tag{13a}$$
$$(\lambda^{k+1} - \mathbf{M}\lambda^{k+1} - \mathbf{c}^k)_i > u_i \Rightarrow \lambda_i^{k+1} = u_i, \tag{13b}$$
$$l_i \leq (\lambda^{k+1} - \mathbf{M}\lambda^{k+1} - \mathbf{c}^k)_i \leq u_i \Rightarrow \lambda_i^{k+1} = (\lambda^{k+1} - \mathbf{M}\lambda^{k+1} - \mathbf{c}^k)_i. \tag{13c}$$

Case (13c) reduces to,

$$(\mathbf{M}\lambda^{k+1})_i = -\mathbf{c}_i^k, \tag{14}$$

which for a suitable choice of $\mathbf{M}$ and back substitution of $\mathbf{c}^k$ gives,

$$\lambda_i^{k+1} = (\mathbf{M}^{-1}(\mathbf{N}\lambda^k - \mathbf{b}))_i. \tag{15}$$

Thus, our iterative splitting method becomes,

$$\min(u_i^k, \max(l_i^k, (\mathbf{M}^{-1}(\mathbf{N}\lambda^k - \mathbf{b}))_i)) = \lambda_i^{k+1}. \tag{16}$$

This is termed a projection method. To realize this, let $\lambda' = \mathbf{M}^{-1}(\mathbf{N}\lambda^k - \mathbf{b})$ then,

$$\lambda^{k+1} = \min(\mathbf{u}^k, \max(\mathbf{l}^k, \lambda')), \tag{17}$$

is the $(k+1)^{\text{th}}$ iterate obtained by projecting the vector λ' onto the box given by $\mathbf{l}^k$ and $\mathbf{u}^k$. Valid splittings of $\mathbf{A}$ are

$$\mathbf{M} = \mathbf{D} \quad \wedge \quad \mathbf{N} = -\mathbf{D} - \mathbf{U}, \tag{18a}$$
$$\mathbf{M} = \mathbf{D} + \mathbf{L} \quad \wedge \quad \mathbf{N} = -\mathbf{U}, \tag{18b}$$
$$\mathbf{M} = \mathbf{D} + \omega\mathbf{L} \quad \wedge \quad \mathbf{N} = (1-\omega)\mathbf{D} - \omega\mathbf{U}, \tag{18c}$$

for $\leq \omega \leq 2$. $\mathbf{L}$, $\mathbf{D}$ and $\mathbf{U}$ are strict lower triangular, diagonal, and strict upper triangular parts of $\mathbf{A}$. These choices results in the projected versions of the Jacobi, Gauss–Seidel and Successive Over Relaxation (SOR) methods respectively. When using the Gauss–Seidel splitting (18b), the resulting PGS method (16) can be efficiently implemented by a forward loop over the components and a component wise projection. Pseudocode for this is,

1 : **for** $k = 1$ **to** $k_{\max}$ **do**
2 : **for** $i = 1$ **to** n **do**
3 : $\lambda_i' \leftarrow \frac{-\sum_{j=1}^{i-1} A_{i,j}\lambda_j - \sum_{j=i+1}^{n} A_{i,j}\lambda_j - b_i}{A_{i,i}}$
4 : $\lambda_i \leftarrow \min(u_i, \max(l_i, \lambda_i'))$
5 : **for all** j **dependent on** i **do**
6 : $(l_j, u_j) \leftarrow$ **update**(λ_i)
7 : **next** j
8 : **next** i
9 : **next** k

To our knowledge no known convergence theorems exist for (16) in the case of variable bounds $l(\lambda)$ and $u(\lambda)$. However, for fixed constant bounds the formulation can be algebraically reduced to that of a LCP formulation. In general, LCP formulations can be shown to have linear convergence rate and unique solutions, when A is symmetric positive definite [4]. However, the **A** matrix equivalent of our frictional contact model is positive symmetric semi definite and uniqueness is no longer guaranteed, but existence of solutions are [4].

5 HEURISTICS FOR CONVERGENCE IMPROVEMENTS

If permutations to the ordering of contacts are made prior to (7) then this is equivalent to use a different sorting order in the splitting (16). Adding a permutation Π to the PGS method alters line 2 of the pseudocode,

$2a$: **for** $p = 1$ **to** n **do**
$2b$: $i \leftarrow \Pi(p)$

There exists a sorting which yields substantial improvements in convergence behavior [3]. Knowing the optimal sorting heuristic for a given problem will improve overall performance.

We have researched numerous heuristics, measuring improvements in convergence, speedup and accuracy. In an effort not to obfuscate the interesting and important results, we will only present the most interesting or promising heuristics. We have divided the heuristics into three groups, physics-based, geometry-based or splitting-based. To represent physics-based heuristics, we have chosen an impulse propagation permutation (IPP) heuristic. This is interesting as it is based on the impulse-based formulation of rigid body dynamics. From the group of geometry-based heuristics, a coordinate permutation (CP) heuristic is examined. Inspired by [10], the splitting-based heuristics are represented by both an extreme staggered permutation (ESP) heuristic as well as a greedy staggered permutation (GSP) heuristic.

Impulse Propagation Permutations: The PGS method solves the contact problem sequentially, which is numerically similar to sequential impulse propagation methods [12]. However, the basis of the PGS method is a simultaneous contact model, while a sequential model is used for impulse propagation methods. Impulse propagation methods attempt to model the propagation of collision impulses through rigid bodies, so perhaps a similar physical principle is beneficial with the PGS method. The idea is to apply a permutation of the problem, mimicking impulse propagation. We base the permutation on a sorting of the relative contact velocities **y**. The sorting is performed as a preprocessing step, the same sorting is used throughout the PGS method. The permutation could be incrementally updated, possibly improving the convergence. This increases the time complexity of each Gauss–Seidel iteration by $\mathcal{O}(nlgn)$. Experience showed that this was to costly, so we only performed an ascending and decreasing pre-processing sorting. Results are presented for an ascending ordering permutation heuristic.

Coordinate Permutations: Consider computing contact impulses for a stack of boxes, this problem has a distinctly dominating up/down direction. If one solves for contacts from bottommost boxes before uppermost boxes, then the solution may propagate in a bottom-up fashion similar to [6]. This indicates there might be certain directions, where shocks can be propagated and vanish in a single sweep. We sort the contact points by their position along each coordinate axis, considering both ascending and descending ordering. Different choices for coordinate axes has also been examined. We also considered a symmetric blocked approach. Blocking was introduced to make sure normal impulses where solved prior to the dependent friction impulses. This set of heuristics are represented by a descending ordering by y coordinates.

Staggered Permutations: A simple adaption of the staggered approach consists in a permutation such that the normal and frictional impulses are separated into two disjoint sequences. There are two approaches, either one alternates between normal impulse and friction impulse at each iteration or one could perform a number of iterations on one set of impulses before switching to iterating a number of times on the other set of impulses. The number of iterations between the switches is determined by a given rule. The latter principle is adopted for the GSP heuristic. The GSP heuristic computes an error measure for the normal impulses and frictional impulses, and the set with the largest error measure is the next to be iterated over. In some cases this heuristic would iterate over one set of impulses until convergence, before moving on to the other set.

As in [10], the full problem can also be split into two coupled subproblems, where one alternate between solving normal and friction impulses in each iteration. To realize the benefit of this extreme staggered approach, let us first split (3) into two coupled subproblems by partitioning the matrix-vector equation into two

coupled equations according to the normal and friction sequences such that,

$$\begin{bmatrix} \mathbf{J}_n^T\mathbf{u} \\ \mathbf{J}_t^T\mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{nn}\lambda_n + \mathbf{A}_{nt}\lambda_t + \mathbf{b}_n \\ \mathbf{A}_{tt}\lambda_t + \mathbf{A}_{tn}\lambda_n + \mathbf{b}_t \end{bmatrix}, \quad (19)$$

where $\mathbf{A}_{ab} = \mathbf{J}_a\mathbf{M}^{-1}\mathbf{J}_b^T$, where $a,b \in n,t$. This problem can be decoupled into two subproblems,

$$\mathbf{J}_n^T\mathbf{u} = [\mathbf{A}_{nn}\lambda_n + \mathbf{A}_{nt}\lambda_t + \mathbf{b}_n] \quad (20)$$

and

$$\mathbf{J}_t^T\mathbf{u} = [\mathbf{A}_{tt}\lambda_t + \mathbf{A}_{tn}\lambda_n + \mathbf{b}_n]. \quad (21)$$

Assuming that λ_t is constant in (20) and vice versa for λ_n in (21), we can define the constants $\mathbf{b}'_n = \mathbf{A}_{nt}\lambda_t + \mathbf{b}_n$ and $\mathbf{b}'_t = \mathbf{A}_{tn}\lambda_n + \mathbf{b}_t$, who are updated when we change subproblem. Instead of solving one problem $\mathbf{A} \in \mathbb{R}^{n\times n}$, we solve two subproblems $\mathbf{A}_{nn} \in \mathbb{R}^{\frac{n}{3}\times\frac{n}{3}}$ and $\mathbf{A}_{tt} \in \mathbb{R}^{\frac{2n}{3}\times\frac{2n}{3}}$.

6 EXPERIMENTS AND RESULTS

We define a residual function from (10) as,

$$\mathbf{H}(\lambda) = \min(\mathbf{u}(\lambda) - \lambda, -\min(\lambda - \mathbf{l}(\lambda), \mathbf{y})) \quad (22)$$

The natural merit function of (22) is,

$$\Theta(\lambda) = \frac{1}{2}\mathbf{H}(\lambda)^T\mathbf{H}(\lambda), \quad (23)$$

which we use as an absolute error measure for the PGS method. We use Q-convergence measures for our analysis [14]. A comparison of convergence rates is done by visual inspections of logarithmic plots of $(iteration, \log(\Theta))$, observe Figure 2.

The test data has been generated using a number of setups from [15]. The setups uses the negative z-axis for the direction of gravity and the x-axis and y-axis span the horizontal plane. The only exception is the *diku*-setup, where the negative y-axis is the direction of gravity. For each setup, $\mathbf{A}$ has been examined by plotting nonzero elements and eigenvalues. Many setups have a spectral radius $\rho(\mathbf{M}^{-1}\mathbf{N}) \geq 1$, which is cause for concern when considering convergence proofs for the Gauss–Seidel method for linear equation systems. The $\mathbf{A}$-matrices are positive semi definite, having a large ratio of zero valued eigenvalues, ranging from (50%-90%) – only the *diku*-setup has no zero valued eigenvalues.

To clarify the experimental results, the test data has been partitioned into equivalency classes [6]. For this paper, only one representative data set from each class is presented.

Non-structured: Few contacts are present, $\mathbf{J}$ is small.

Loose structured: Large number of contacts without too much structure. This means that normal and tangent directions will be varied. This results in a large $\mathbf{J}$ with some redundancy.

Dense structured: Large number of contacts with similar normal and tangent directions. This results in a large $\mathbf{J}$ with a high degree of redundancy. This increases the number of zero valued eigenvalues in $\mathbf{A}$.

Setup	Type	# Contacts	# Bodies
diku	Non-structured	105	1001
card	Loose structured	154	43
box stack	Dense structured	68	10

Table 1: Sample setups and their complexity.

All tests were performed on a system with Intel Core 2 Duo P8600 2.4GHz CPU and 3GB RAM, running Windows XP SP 3 after a clean boot. In total, 35 heuristics were examine using 10 data sets. The heuristics were implemented and evaluated using MatLab. For each iteration, the error (23) and computation time were measured. Most of the plots exhibit piecewise linear convergence, and seem to converge towards some local minimum of the error function. The time measurements have been used to compute the time speedup

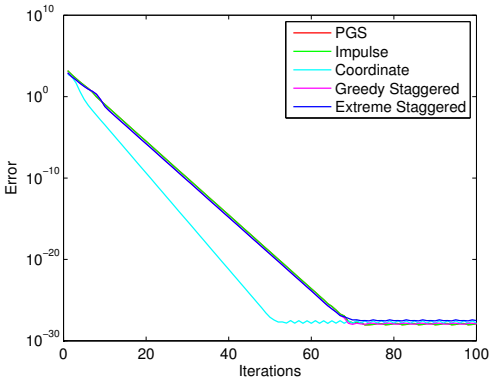
$$Speedup = \frac{t_{PGS}}{t_{heuristic}}. \quad (24)$$

Table 2 and 3 show the speedups that were significantly different from 1.

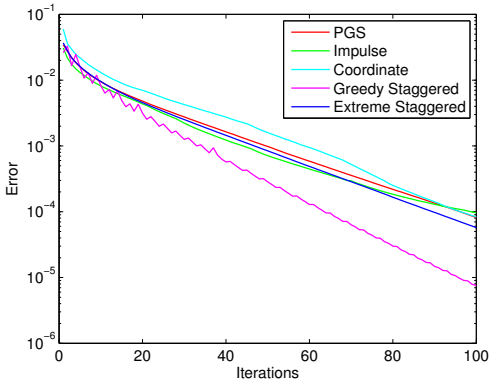
Impulse Propagation Permutation: Inspection of Figure 2 shows that the IPP heuristics performs similar to the pure PGS method. There seems to be no benefit from using this heuristic.

Coordinate Permutation: A number of CP heuristic variations have been investigated sorting by x, y, and z coordinates. The most interesting results arose when sorting descending by y coordinate, which is shown on Figure 2. This particular permutation performs well on dense and non-structured setups, but badly on loose structured setups. While not shown, sorting by descending y coordinates performs better than sorting by the direction of gravity.

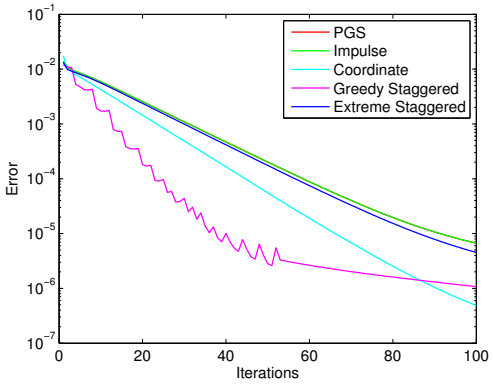
Greedy Staggered Permutation: As Figure 2 shows, the GSP heuristic performs at least as well as the pure PGS method, and often better. When changing from normal iterations to friction iterations (or vice versa), a short burst in the rate of convergence is experienced. This causes the jagged shape of the convergence plots. Another noticeable feature of the GSP heuristic is the speedup. As Table 2 shows, the GPS heuristic is roughly twice as fast as the pure PGS method.



(a) Non-structured setup.



(b) Loose structured setup.



(c) Dense structured setup.

Figure 2: Comparative convergence plots for four different heuristics. Convergence for the PGS method without heuristics is included as a baseline for the comparison. In some cases, PGS is obfuscated by the impulse propagation permutation heuristic.

Setup	Type	Time Speedup
diku	Non-structured	1.7
card	Loose structured	2.0
box stack	Dense structured	2.7

Table 2: Speedup measures of the greedy staggered heuristic compared to the pure projected Gauss–Seidel method. Numbers larger than 1 indicate smaller time usage per iteration.

Extreme Staggered Permutation: The ESP heuristic performs similar to the pure PGS method. However, it uses much less time per iteration than the pure PGS method, see Table 3. The speedup is more pronounced for the larger data sets than for the smaller data sets.

Setup	Type	Time Speedup
diku	Non-structured	1.4
card	Loose structured	2.2
box stack	Dense structured	1.9

Table 3: Speedup measures of the extreme staggered permutation heuristic compared to the pure projected Gauss–Seidel method. Numbers larger than 1 indicate smaller time usage per iteration.

7 CONCLUSION AND DISCUSSION

Based on our test data, using a greedy choice for deciding between performing a normal or friction iteration results in an improved convergence rate in most cases, but consistently lower time usage per iteration. The extreme staggered permutation (ESP) heuristic splits the problem into two sub problems, thereby reducing the time usage per iteration.

Sorting by the y-axis yields improvements in the rate of convergence. We believe this is due to contact points of each sub sequence being spread over the entire problem. It is not surprising that exploiting geometry information can improve convergence rate. However, the problem lies in obtaining this prior knowledge at run time, unless such information is given at design time. Therefore, as a general heuristic coordinate permutation seem to have only little practical usage. Another interesting result is that any of the symmetric coordinate permutations seems to perform worse than the pure projected Gauss–Seidel (PGS) method.

In our opinion, staggered permutations are the most promising heuristics. It is interesting to hypothesize on why this is so. Are their success the result of improved numerical behavior of the method when decoupling normal impulses from friction impulses? Normal impulses control the bounds of the friction impulses, whereas the friction impulses have an indirect, yet significant, effect on normal impulses through the dynamics of multiple contacts. Or is it because the friction model used in the nonlinear complementarity problem (NCP) formulation is too poor a model? Should future

work focus on creating improved contact models for interactive simulation that has a better friction model component? Or should future numerical schemes be based on the splitting idea? The results by [18] suggest that convergence rate for the friction component of the NCP formulation is much slower than for the normal component, This indicates to us that the NCP formulation is a poor friction model. In [10] an accurate linear complementarity problem (LCP) formulation is used to model friction, and they obtain nice results for friction. This again indicates that the current friction model widely used in interactive rigid body simulators is too poor. We will leave these questions open for future work on interactive rigid body simulation.

REFERENCES

[1] Mihai Anitescu and Florian A. Potra. Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlinear Dynamics. An International Journal of Nonlinear Dynamics and Chaos in Engineering Systems*, 14(3):231–247, 1997.

[2] David Baraff. Fast contact force computation for nonpenetrating rigid bodies. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 23–34, New York, NY, USA, 1994. ACM.

[3] Richard L. Burden and J. Douglas Faires. *Numerical Analysis*. Books/Cole Publishing Company, 1997.

[4] Richard Cottle, Jong-Shi Pang, and Richard E. Stone. *The Linear Complementarity Problem*. Computer Science and Scientific Computing. Academic Press, February 1992.

[5] Kenny Erleben. *Stable, Robust, and Versatile Multibody Dynamics Animation*. PhD thesis, Department of Computer Science, University of Copenhagen (DIKU), 2005.

[6] Kenny Erleben. Velocity-based shock propagation for multibody dynamics animation. *ACM Trans. Graph.*, 26(2):12, 2007.

[7] Roy Featherstone. *Robot Dynamics Algorithms*. Kluwer Academic Publishers, second printing edition, 1998.

[8] Eran Guendelman, Robert Bridson, and Ronald Fedkiw. Nonconvex rigid bodies with stacking. *ACM Trans. Graph.*, 22(3):871–878, 2003.

[9] James K. Hahn. Realistic animation of rigid bodies. In *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 299–308, New York, NY, USA, 1988. ACM.

[10] Danny M. Kaufman, Shinjiro Sueda, Doug L. James, and Dinesh K. Pai. Staggered projections for frictional contact in multibody systems. *ACM Trans. Graph.*, 27(5):1–11, 2008.

[11] Victor J. Milenkovic and Harald Schmidl. A fast impulsive contact suite for rigid body simulation. *IEEE Transactions on Visualization and Computer Graphics*, 10(2):189–197, 2004.

[12] Brian Vincent Mirtich. *Impulse-based dynamic simulation of rigid body systems*. PhD thesis, University of California, Berkeley, 1996.

[13] Matthew Moore and Jane Wilhelms. Collision detection and response for computer animation. In *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 289–298, New York, NY, USA, 1988. ACM.

[14] Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer Series in Operations Research. Springer-Verlag, New York, 1999.

[15] OpenTissue. Opensource project, physics-based animation and surgery simulation, www.opentissue.org, 2009.

[16] Carol O'Sullivan, John Dingliana, Thanh Giang, and Mary K. Kaiser. Evaluating the visual fidelity of physically based animations. *ACM Trans. Graph.*, 22(3):527–536, 2003.

[17] Stephane Redon, Abderrahmane Kheddar, and Sabine Coquillart. Gauss least constraints principle and rigid body simulations. In *In proceedings of IEEE International Conference on Robotics and Automation*, 2003.

[18] Morten Silcowitz, Sarah M. Niebe, and Kenny Erleben. Nonsmooth Newton Method for Fischer Function Reformulation of Contact Force Problems for Interactive Rigid Body Simulation. In *Virtual Reality Interactions and Physical Simulations (VRIPhys)*, 2009.

[19] David E. Stewart. Rigid-body dynamics with friction and impact. *SIAM Review*, 42(1):3–39, 2000.

[20] David E. Stewart and Jeff C. Trinkle. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *International Journal of Numerical Methods in Engineering*, 39(15):2673–2691, 1996.

[21] Jeff C. Trinkle, James Tzitzoutis, and Jong-Shi Pang. Dynamic multi-rigid-body systems with concurrent distributed contacts: Theory and examples. *Philosophical Trans. on Mathematical, Physical, and Engineering Sciences*, 359(1789):2575–2593, December 2001.

C GRAPP 2010 Article

I have, in collaboration with Morten Silcowitz and Kenny Erleben, submitted a paper for the GRAPP 2010 International Conference on Computer Graphics Theory and Applications. The acceptance decision is pending, the answer is expected in late January 2010. The paper describes an extension of the projected Gauss–Seidel method, using a subspace minimization approach.

PROJECTED GAUSS–SEIDEL SUBSPACE MINIMIZATION METHOD FOR INTERACTIVE RIGID BODY DYNAMICS

Improving Animation Quality Using A Projected Gauss–Seidel Subspace Minimization Method

Morten Silcowitz, Sarah Niebe and Kenny Erleben
eScience Center, Department of Computer Science, University of Copenhagen, Denmark
mosh@diku.dk, niebe@diku.dk and kenny@diku.dk

Keywords: Contact Force Problem, Complementarity Formulation, Projected Gauss–Seidel, Subspace Minimization.

Abstract: In interactive physical simulation, contact forces are applied to prevent rigid bodies from penetrating and to control slipping between bodies. Accurate contact force determination is a computationally hard problem. Thus, in practice one trades accuracy for performance. This results in visual artifacts such as viscous or damped contact response. In this paper, we present a new approach to contact force determination. We formulate the contact force problem as a nonlinear complementarity problem, and discretize the problem to derive the Projected Gauss–Seidel method. We combine the Projected Gauss–Seidel method with a subspace minimization method. Our new method shows improved qualities and superior convergence properties for specific configurations.

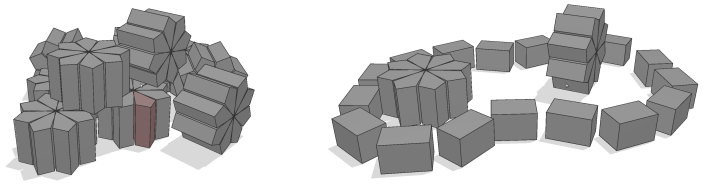
1 INTRODUCTION

Most open source software for interactive real-time rigid body simulation use the widespread Projected Gauss–Seidel (PGS) method, examples are Bullet and Open Dynamics Engine. However, the PGS method is not always satisfactory, it suffers from two problems: linear convergence rate (Cottle et al., 1992) and inaccurate friction forces in stacks (Kaufman et al., 2008). Linear convergence results in viscous motion at contacts and loss of high frequency effects. The viscous appearance results in a time delay in contact responses and reduces plausibility (O'Sullivan et al., 2003). This has motivated us to develop a new numerical method, based on a nonlinear complementarity formulation of the contact force problem. The method combines the PGS method with a subspace minimization solver. The contribution is simple to implement, and existing PGS implementations can easily be extended using our ideas. Our method is compared to the PGS method for interactive simulation.

1.1 PREVIOUS WORK

Rigid body simulation was introduced to the graphics community in the late 1980's (Hahn, 1988; Moore and Wilhelms, 1988), using penalty based and impulse based approaches to describe physical interactions. Penalty based simulation is not easily adopted to different simulations. Mirtich (Mirtich, 1996) presented an extended and improved impulse based formulation, however stacking was a problem and it suffered from creeping. These problems has since been rectified (Guendelman et al., 2003). Constraint based simulation (Baraff, 1994) has received much attention as an alternative to penalty based and impulse based simulation. Constraint based simulation can be divided into two groups: maximal coordinate and minimal coordinate methods (Featherstone, 1998). The focus of this paper is maximal coordinate methods, which are dominated by complementarity formulations. Alternatives to complementarity formulations are based on kinetic energy (Milenkovic and Schmidl, 2004) and motion space (Redon et al., 2003). However, the former solves a more general problem and is not attractive for performance reasons, the latter does not include frictional forces.

Complementarity formulations are either acceleration based formulations (Trinkle et al., 2001) or velocity based formulations (Stewart and Trinkle, 1996). Acceleration based formulations can not handle collisions (Anitescu and Potra, 1997), in addi-



(a) (b)

Figure 1: Various configurations animated using the PGS–SM method. Configurations include composite rigid bodies, contact constraints, and hinge joint constraints with joint limits

tion they suffer from indeterminacy and inconsistency (Stewart, 2000). Velocity based formulations suffer from none of these problems, for this reason we use a velocity based formulation for this paper. The approach we present is based on a reformulating the frictional problem as a nonlinear complementarity problem. The result is a slightly inaccurate model, with relatively few variables to solve for. This makes it advantageous in interactive simulations from a performance viewpoint.

Our work is inspired by (Morales et al., 2008; Arechavaleta et al., 2009). The novel contribution consists of tailoring the ideas to suit the NCP formulation, used for interactive rigid body dynamics. Further, we present optimizations to the subspace method to make the simulation fast enough for interactive usage. Prior work is limited to a linear complementarity problem formulation, missing evaluation of the ideas for general rigid body simulation. Application is limited to grasping, which is dominated by bilateral constraints and static friction. Here we evaluate with focus on both dynamic and static friction, as well as normal constraints.

2 THE NONLINEAR COMPLEMENTARITY PROBLEM FORMULATION

The frictional contact force problem can be stated as a linear complementarity problem (LCP) (Stewart and Trinkle, 1996). However, a different formulation is used in interactive physical simulations, we will derive this formulation. Without loss of generality, we will only consider a single contact point. The focus of this paper is on the contact force model, so the time stepping scheme and matrix layouts are based on the velocity-based formulation in (Erleben, 2007).

We have the Newton–Euler equations,

$$\mathbf{M}\mathbf{v} - \mathbf{J}_n^T\lambda_n - \mathbf{J}_t^T\lambda_t = \mathbf{F}, \tag{1}$$

where $\mathbf{J}_n$ is the Jacobian corresponding to normal constraints and $\mathbf{J}_t$ is the Jacobian corresponding to the tangential contact impulses. $\mathbf{M}$ is the generalized mass matrix and $\mathbf{v}$ is the generalized velocity vector. We wish to solve for $\mathbf{v}$ in order to compute a position update. For readability we have, without loss of generality, abstracted the discretization details within the Lagrange multipliers λ_n, λ_t and generalized external impulses $\mathbf{F}$. Since the contact plane is two-dimensional, we span this plane by two orthogonal unit vectors, t_1 and t_2. Any vector in this plane can be written as a linear combination of these two vectors. Thus, $\mathbf{J}_t$ has only two rows corresponding to the two directions. From (1) we can obtain the generalized velocities,

$$\mathbf{v} = \mathbf{M}^{-1}\mathbf{F} + \mathbf{M}^{-1}\mathbf{J}_n^T\lambda_n + \mathbf{M}^{-1}\mathbf{J}_t^T\lambda_t. \tag{2}$$

Let the Lagrange multipliers $\lambda = \begin{bmatrix}\lambda_n & \lambda_t^T\end{bmatrix}^T$ and contact Jacobian $\mathbf{J} = \begin{bmatrix}\mathbf{J}_n & \mathbf{J}_t\end{bmatrix}^T$, then we write the relative contact velocities $\mathbf{y} = \begin{bmatrix}y_n & \mathbf{y}_t^T\end{bmatrix}^T$ such that,

$$\mathbf{y} = \mathbf{J}\mathbf{v} = \underbrace{\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T}_{\mathbf{A}}\lambda + \underbrace{\mathbf{J}\mathbf{M}^{-1}\mathbf{F}}_{\mathbf{b}}. \tag{3}$$

To compute the frictional component of the contact impulse, we need a model of friction. We base our model on Coulomb's friction law. In one dimension, Coulomb's friction law can be written as (Baraff, 1994),

$$y < 0 \Rightarrow \lambda_t = \mu\lambda_n, \tag{4a}$$
$$y > 0 \Rightarrow \lambda_t = -\mu\lambda_n, \tag{4b}$$
$$y = 0 \Rightarrow -\mu\lambda_n \leq \lambda_t \leq \mu\lambda_n. \tag{4c}$$

For the full contact problem, we split $\mathbf{y}$ into positive and negative components,

$$\mathbf{y} = \mathbf{y}^+ - \mathbf{y}^-, \tag{5}$$

where

$$\mathbf{y}^+ \geq 0, \quad \mathbf{y}^- \geq 0 \quad \text{and} \quad \left(\mathbf{y}^+\right)^T \left(\mathbf{y}^-\right) = 0. \tag{6}$$

For the frictional impulses, we define the bounds $-l_t(\lambda) = u_t(\lambda) = \mu\lambda_n$ and for the normal impulse $l_n(\lambda) = 0$ and $u_n(\lambda) = \infty$. Combining the bounds with (4), (5) and (6), we reach the final nonlinear complementarity problem (NCP) formulation,

$$\mathbf{y}^+ - \mathbf{y}^- = \mathbf{A}\lambda + \mathbf{b}, \tag{7a}$$
$$\mathbf{y}^+ \geq 0, \tag{7b}$$
$$\mathbf{y}^- \geq 0, \tag{7c}$$
$$u(\lambda) - \lambda \geq 0, \tag{7d}$$
$$\lambda - l(\lambda) \geq 0, \tag{7e}$$
$$\left(\mathbf{y}^+\right)^T (\lambda - l(\lambda)) = 0, \tag{7f}$$
$$\left(\mathbf{y}^-\right)^T (u(\lambda) - \lambda) = 0, \tag{7g}$$
$$\left(\mathbf{y}^+\right)^T \left(\mathbf{y}^-\right) = 0, \tag{7h}$$

where $l(\lambda) = [l_n(\lambda) \quad \mathbf{l}_t(\lambda)^T]^T$ and $u(\lambda) = [u_n(\lambda) \quad \mathbf{u}_t(\lambda)^T]^T$. The advantage of the NCP formulation is a much lower memory footprint than for the LCP formulation. The disadvantage is solving the friction problem as two decoupled one-dimensional Coulomb friction models.

3 THE PROJECTED GAUSS–SEIDEL METHOD

The following is a derivation of the PGS method for solving the frictional contact force problem, stated as the NCP (7). Using a minimum map reformulation, the i^{th} component of (7) can be written as

$$(\mathbf{A}\lambda + \mathbf{b})_i = \mathbf{y}_i^+ - \mathbf{y}_i^-, \tag{8a}$$
$$\min(\lambda_i - l_i, \mathbf{y}_i^+) = 0, \tag{8b}$$
$$\min(u_i - \lambda_i, \mathbf{y}_i^-) = 0. \tag{8c}$$

where $l_i = l_i(\lambda)$ and $u_i = u_i(\lambda)$. Note, when $\mathbf{y}_i^- > 0$ we have $\mathbf{y}_i^+ = 0$ which in turn means that $\lambda_i - l_i \geq 0$. In this case, (8b) is equivalent to

$$\min(\lambda_i - l_i, \mathbf{y}_i^+ - \mathbf{y}_i^-) = -(\mathbf{y}^-)_i. \tag{9}$$

If $\mathbf{y}_i^- = 0$ then $\lambda_i - l_i = 0$ and complementarity constraint (8b) is trivially satisfied. Substituting (9) for $\mathbf{y}_i^-$ in (8c) yields,

$$\min(u_i - \lambda_i, \max(l_i - \lambda_i, -(\mathbf{y}^+ - \mathbf{y}^-)_i)) = 0. \tag{10}$$

This is a more compact reformulation than (7) and eliminates the need for auxiliary variables $\mathbf{y}^+$ and $\mathbf{y}^-$. By adding λ_i we get a fixed point formulation

$$\min(u_i, \max(l_i, \lambda_i - (\mathbf{A}\lambda + \mathbf{b})_i)) = \lambda_i. \tag{11}$$

We introduce the splitting $\mathbf{A} = \mathbf{M} - \mathbf{N}$ and an iteration index k. Then we define $\mathbf{c}^k = \mathbf{b} - \mathbf{N}\lambda^k$, $l^k = l(\lambda^k)$ and $u^k = u(\lambda^k)$. Using this we have

$$\min(u_i^k, \max(l_i^k, (\lambda^{k+1} - \mathbf{M}\lambda^{k+1} - \mathbf{c}^k)_i)) = \lambda_i^{k+1}. \tag{12}$$

When $\lim_{k\to\infty} \lambda^k = \lambda^*$ then (12) is equivalent to (7). Next we perform a case-by-case analysis. Three cases are possible,

$$(\lambda^{k+1} - \mathbf{M}\lambda^{k+1} - \mathbf{c}^k)_i < l_i \Rightarrow \lambda_i^{k+1} = l_i, \tag{13a}$$
$$(\lambda^{k+1} - \mathbf{M}\lambda^{k+1} - \mathbf{c}^k)_i > u_i \Rightarrow \lambda_i^{k+1} = u_i, \tag{13b}$$
$$l_i \leq (\lambda^{k+1} - \mathbf{M}\lambda^{k+1} - \mathbf{c}^k)_i \leq u_i \Rightarrow \lambda_i^{k+1} = (\lambda^{k+1} - \mathbf{M}\lambda^{k+1} - \mathbf{c}^k)_i. \tag{13c}$$

Case (13c) reduces to,

$$(\mathbf{M}\lambda^{k+1})_i = -\mathbf{c}_i^k, \tag{14}$$

which for a suitable choice of $\mathbf{M}$ and back substitution of $\mathbf{c}^k$ gives,

$$\lambda_i^{k+1} = (\mathbf{M}^{-1}(\mathbf{N}\lambda^k - \mathbf{b}))_i. \tag{15}$$

Thus, our iterative splitting method becomes,

$$\min(u_i^k, \max(l_i^k, (\mathbf{M}^{-1}(\mathbf{N}\lambda^k - \mathbf{b}))_i)) = \lambda_i^{k+1}. \tag{16}$$

This is termed a projection method. To realize this, let $\lambda' = \mathbf{M}^{-1}(\mathbf{N}\lambda^k - \mathbf{b})$ then,

$$\lambda^{k+1} = \min(\mathbf{u}^k, \max(\mathbf{l}^k, \lambda')), \tag{17}$$

is the $(k+1)^{\text{th}}$ iterate obtained by projecting the vector λ' onto the box given by $\mathbf{l}^k$ and $\mathbf{u}^k$. Using the splitting $\mathbf{M} = \mathbf{D} + \mathbf{L}$ and $\mathbf{N} = -\mathbf{U}$ results in the Projected Gauss–Seidel method. The resulting PGS method (17) can be efficiently implemented as a forward loop over all components and a component wise projection. To our knowledge no known convergence theorems exist for (16) in the case of variable bounds $l(\lambda)$ and $u(\lambda)$. However, for fixed constant bounds the formulation can be algebraically reduced to that of a LCP formulation. In general, LCP formulations can be shown to have linear convergence rate and unique solutions, when A is symmetric positive definite (Cottle et al., 1992). However, the $\mathbf{A}$ matrix equivalent of our frictional contact model is positive symmetric semi definite and uniqueness is no longer guaranteed, but existence of solutions are (Cottle et al., 1992)

4 THE PROJECTED GAUSS–SEIDEL SUBSPACE MINIMIZATION METHOD

We will present a Projected Gauss–Seidel Subspace Minimization (PGS–SM) method, specifically tai-

lored for the nonlinear complementarity problem formulation of the contact force problem. Unlike previous work, which is limited to the linear complementarity problem formulation, our method is more general and further specialized for interactive usage. The PGS–SM method is an iterative method, each iteration consisting of two phases. The first phase estimates a set of active constraints, using the standard PGS method. The second phase solves accurately for the active constraints, potentially further reducing the set of active constraints for the next iteration. In the following we will describe the details of the PGS–SM method.

4.1 DETERMINING INDEX SETS

We define three index sets corresponding to our choice of active constraints in (17)

$$\mathcal{L} \equiv \{i|\mathbf{y}_i > 0\} \tag{18a}$$
$$\mathcal{U} \equiv \{i|\mathbf{y}_i < 0\} \tag{18b}$$
$$\mathcal{A} \equiv \{i|\mathbf{y}_i = 0\} \tag{18c}$$

assuming $\mathbf{l}_i \leq 0 < \mathbf{u}_i$ for all i. The definition in (18) is based on the $\mathbf{y}$-vector. However, one may just as well use the λ-vector, thus having

$$\mathcal{L} \equiv \{i|\lambda_i = \mathbf{l}_i\} \tag{19a}$$
$$\mathcal{U} \equiv \{i|\lambda_i = \mathbf{u}_i\} \tag{19b}$$
$$\mathcal{A} \equiv \{i|\mathbf{l}_i < \lambda_i < \mathbf{u}_i\} \tag{19c}$$

When the PGS method terminates, we know λ is feasible (although not the correct solution). However, $\mathbf{y}$ may be infeasible due to the projection on λ made by the PGS method. This votes in favor of using (19) rather than (18). In our initial test trials, no hybrid or mixed schemes seemed to be worth the effort. Therefore, we use the classification defined in (19) for the PGS–SM method.

4.2 POSING THE REDUCED PROBLEM

Next we use a permutation of the indexes, creating the imaginary partitioning of the system of linear equations (3)

$$\begin{bmatrix}\mathbf{y}_\mathcal{A}\\ \mathbf{y}_\mathcal{L}\\ \mathbf{y}_\mathcal{U}\end{bmatrix} = \begin{bmatrix}\mathbf{A}_{\mathcal{A}\mathcal{A}} & \mathbf{A}_{\mathcal{A}\mathcal{L}} & \mathbf{A}_{\mathcal{A}\mathcal{U}}\\ \mathbf{A}_{\mathcal{L}\mathcal{A}} & \mathbf{A}_{\mathcal{L}\mathcal{L}} & \mathbf{A}_{\mathcal{L}\mathcal{U}}\\ \mathbf{A}_{\mathcal{U}\mathcal{A}} & \mathbf{A}_{\mathcal{U}\mathcal{L}} & \mathbf{A}_{\mathcal{U}\mathcal{U}}\end{bmatrix}\begin{bmatrix}\lambda_\mathcal{A}\\ \lambda_\mathcal{L}\\ \lambda_\mathcal{U}\end{bmatrix} + \begin{bmatrix}\mathbf{b}_\mathcal{A}\\ \mathbf{b}_\mathcal{L}\\ \mathbf{b}_\mathcal{U}\end{bmatrix}. \tag{20}$$

Utilizing that $\forall i \in \mathcal{A} \Rightarrow \mathbf{y}_i = 0$, as well as $\forall i \in \mathcal{L} \Rightarrow \lambda_i = \mathbf{l}_i$ and $\forall i \in \mathcal{U} \Rightarrow \lambda_i = \mathbf{u}_i$, we get

$$\begin{bmatrix}\mathbf{0}\\ \mathbf{y}_\mathcal{L}\\ \mathbf{y}_\mathcal{U}\end{bmatrix} = \begin{bmatrix}\mathbf{A}_{\mathcal{A}\mathcal{A}} & \mathbf{A}_{\mathcal{A}\mathcal{L}} & \mathbf{A}_{\mathcal{A}\mathcal{U}}\\ \mathbf{A}_{\mathcal{L}\mathcal{A}} & \mathbf{A}_{\mathcal{L}\mathcal{L}} & \mathbf{A}_{\mathcal{L}\mathcal{U}}\\ \mathbf{A}_{\mathcal{U}\mathcal{A}} & \mathbf{A}_{\mathcal{U}\mathcal{L}} & \mathbf{A}_{\mathcal{U}\mathcal{U}}\end{bmatrix}\begin{bmatrix}\lambda_\mathcal{A}\\ \mathbf{l}_\mathcal{L}\\ \mathbf{u}_\mathcal{U}\end{bmatrix} + \begin{bmatrix}\mathbf{b}_\mathcal{A}\\ \mathbf{b}_\mathcal{L}\\ \mathbf{b}_\mathcal{U}\end{bmatrix}. \tag{21}$$

To solve this system for $\mathbf{y}_\mathcal{L}$, $\mathbf{y}_\mathcal{U}$ and $\lambda_\mathcal{A}$, we first compute $\lambda_\mathcal{A}$ from

$$\mathbf{A}_{\mathcal{A}\mathcal{A}}\lambda_\mathcal{A} = -\left(\mathbf{b}_\mathcal{A} + \mathbf{A}_{\mathcal{A}\mathcal{L}}\mathbf{l}_\mathcal{L} + \mathbf{A}_{\mathcal{A}\mathcal{U}}\mathbf{u}_\mathcal{U}\right). \tag{22}$$

Observe that $\mathbf{A}_{\mathcal{A}\mathcal{A}}$ is a symmetric principal submatrix of $\mathbf{A}$. Knowing $\lambda_\mathcal{A}$, we can easily compute $\mathbf{y}_\mathcal{L}$ and $\mathbf{y}_\mathcal{U}$,

$$\mathbf{y}_\mathcal{L} \leftarrow \mathbf{A}_{\mathcal{L}\mathcal{A}}\lambda_\mathcal{A} + \mathbf{A}_{\mathcal{L}\mathcal{L}}\mathbf{l}_\mathcal{L} + \mathbf{A}_{\mathcal{L}\mathcal{U}}\mathbf{u}_\mathcal{U} + \mathbf{b}_\mathcal{L}, \tag{23a}$$
$$\mathbf{y}_\mathcal{U} \leftarrow \mathbf{A}_{\mathcal{U}\mathcal{A}}\lambda_\mathcal{A} + \mathbf{A}_{\mathcal{U}\mathcal{L}}\mathbf{l}_\mathcal{L} + \mathbf{A}_{\mathcal{U}\mathcal{U}}\mathbf{u}_\mathcal{U} + \mathbf{b}_\mathcal{U}. \tag{23b}$$

Finally, we verify that $\mathbf{y}_\mathcal{L} < 0$, $\mathbf{y}_\mathcal{U} > 0$ and $\mathbf{l}_\mathcal{A} \leq \lambda_\mathcal{A} \leq \mathbf{u}_\mathcal{A}$. If this holds, we have found a solution. Rather than performing the tests explicitly, it is more simple to perform a projection on the reduced problem

$$\lambda_\mathcal{A} \leftarrow \min(\mathbf{u}_\mathcal{A}, \max(\mathbf{l}_\mathcal{A}, \lambda_\mathcal{A})) \tag{24}$$

We assemble the full solution $\lambda \leftarrow \begin{bmatrix}\lambda_\mathcal{A}^T & \mathbf{l}_\mathcal{L}^T & \mathbf{u}_\mathcal{U}^T\end{bmatrix}^T$, before reestimating the index sets for the next iteration. Observe that the projection on the reduced problem will either leave the active set unchanged or reduce it further.

4.3 THE COMPLETE ALGORITHM

The resulting algorithm can be outlined as

while not convergence do
  $\lambda \leftarrow$ **run PGS for at least** k_{pgs} **iterations**
  if termination criteria is passed then
    return λ
  endif
  for $k = 1$ **to** k_{sm}
    $\mathcal{L} \equiv \{i|\lambda_i = \mathbf{l}_i\}$
    $\mathcal{U} \equiv \{i|\lambda_i = \mathbf{u}_i\}$
    $\mathcal{A} \equiv \{i|\mathbf{l}_i < \lambda_i < \mathbf{u}_i\}$
    solve: $\mathbf{A}_{\mathcal{A}\mathcal{A}}\lambda_\mathcal{A} = -\left(\mathbf{b}_\mathcal{A} + \mathbf{A}_{\mathcal{A}\mathcal{L}}\mathbf{l}_\mathcal{L} + \mathbf{A}_{\mathcal{A}\mathcal{U}}\mathbf{u}_\mathcal{U}\right)$
    $\mathbf{y}_\mathcal{L} \leftarrow \mathbf{A}_{\mathcal{L}\mathcal{A}}\lambda_\mathcal{A} + \mathbf{A}_{\mathcal{L}\mathcal{L}}\mathbf{l}_\mathcal{L} + \mathbf{A}_{\mathcal{L}\mathcal{U}}\mathbf{u}_\mathcal{U} + \mathbf{b}_\mathcal{L}$,
    $\mathbf{y}_\mathcal{U} \leftarrow \mathbf{A}_{\mathcal{U}\mathcal{A}}\lambda_\mathcal{A} + \mathbf{A}_{\mathcal{U}\mathcal{L}}\mathbf{l}_\mathcal{L} + \mathbf{A}_{\mathcal{U}\mathcal{U}}\mathbf{u}_\mathcal{U} + \mathbf{b}_\mathcal{U}$
    update: $(\mathbf{l}, \mathbf{u})$
    $\lambda_\mathcal{A} \leftarrow \min(\mathbf{u}_\mathcal{A}, \max(\mathbf{l}_\mathcal{A}, \lambda_\mathcal{A}))$
    $\lambda \leftarrow \begin{bmatrix}\lambda_\mathcal{A}^T & \mathbf{l}_\mathcal{L}^T & \mathbf{u}_\mathcal{U}^T\end{bmatrix}^T$
    if termination criteria is passed then
      return λ
    endif
  next k
end while

An absolute termination criteria could be applied

$$\psi(\lambda) < \varepsilon_{\text{abs}} \tag{25}$$

for some user specified value ε_{abs}. An alternative termination criteria could be to monitor if the set $\mathcal{A}$ has changed from previous iteration,

$$\mathcal{A}(\lambda^{k+1}) = \mathcal{A}(\lambda^k). \tag{26}$$

A third termination criteria could be testing for stagnation

$$\psi(\lambda^{k+1}) - \psi(\lambda^k) < \psi(\lambda^k)\varepsilon_{\text{rel}}. \quad (27)$$

for some user specified value $\varepsilon_{\text{rel}} > 0$. Other merit-functions could be used in place of ψ. Examples include natural merit functions of the Fischer reformulation (Silcowitz et al., 2009) or the minimum map reformulation (Erleben and Ortiz, 2008). We prefer the Fischer reformulation, as it seems to be more global in the inclusion of boundary information (Billups, 1995). Finally, to ensure interactive performance, one could use an absolute termination criteria on the number of iterations. Using such a criteria, the algorithm may not have iterations enough to reach an accurate solution, we observed this behavior in a few cases. To counter this, a fall back to the best iterate found while iterating could be employed. This would ensure that the PGS–SM method behaves no worse than the PGS method would have done.

5 EXPERIMENTS

We have compared the PGS–SM method to the standard PGS method. For testing the PGS–SM method, we have selected various test cases which we believe to be challenging. The test cases are shown in Figure 2. The test cases include bilateral hinge joints with joint limits, large mass-ratios, inclined plane setups to provoke static friction handling, stacked configurations of different sizes with both box and gear geometries.

Convergence rates for all the test cases are shown in Figure 3. In order to ease comparison, great care is taken to measure the time usage of both methods in units of PGS iterations.

For the tests, we use the iteration limits $k_{\text{pgs}} = 25$ and $k_{\text{sm}} = 5$. Further we use an error tolerance of $\varepsilon_{\text{abs}} = 1e-15$. For the reduced problem we use a non-preconditioned Conjugate Gradient (CG) method with a maximum iteration count equal to the number of variables, and an error tolerance on the residual of $\varepsilon_{\text{residual}} = 1e-15$. The algorithms were implemented in Java using JOGL, and the tests were run on a Lenovo T61 2.0Ghz machine.

As observed in Figure 3, the PGS–SM method behaves rather well for small configurations and configuration with joints. For larger configurations, we obtain convergences similar to the PGS method.

The supplementary video shows interactive simulations of an articulated snake-like figure, comparing the animation quality of the PGS–SM method to the PGS method. All test cases run at interactive frame rates, 25 fps or above. We have observed a different quality in the motion simulated by the PGS–SM method. It is our hypothesis that the PGS–SM method seems to favor static friction over dynamic friction. Our subjective impression is that the PGS–SM method delivers a more plausible animation quality.

The presented algorithm is capable of very accurate computations, compared to the PGS method. However, we have observed problematic instances where simulation blow-up was noticed. The simulation blow-ups appear to occur regardless of how accurate the subspace problem is solved. We observed blow-ups even when using a singular value decomposition pseudo inverse of the reduced problem (22).

In general, if bounds are fixed the problem reduces to a LCP formulation. Applying a simple diagonalization to the LCP, using an eigenvalue decomposition of $\mathbf{A}$, one can easily show that a solution to the problem always exists when $\mathbf{A}$ is positive semi definite. However, when bounds are variable the nonlinear nature of the problem makes it hard to say anything conclusive about existence of a solution. The accuracy of the system is thus clearly affected, when attempting to solve a system that has no solution. The effect can be observed in the behavior of the PGS method. By increasing the number of iterations, the PGS method will converge to a positive merit value. This indicates convergence to a local minimizer of the merit function, and not a global minimizer.

5.1 STABILITY IMPROVEMENTS

Stability can be improved by adding minor changes to the presented algorithm. Such a change could be the use of a relative termination criteria for the PGS method similar to (27), thus forcing the method to iterate long enough to improve the estimate of the active set. In our experience, this can be very beneficial although it counteracts interactive performance.

Another strategy that seems to improve stability, is to add numerical regularization to the sub-problem. The matrix $\mathbf{A}_{\mathcal{A}\mathcal{A}}$ is replaced with $\mathbf{A}'_{\mathcal{A}\mathcal{A}} = \mathbf{A}_{\mathcal{A}\mathcal{A}} + \gamma\mathbf{I}$ for some positive scalar γ. The regularization makes $\mathbf{A}'_{\mathcal{A}\mathcal{A}}$ positive definite, which improves the performance of the CG method. The result of the regularization is observed as a damping in the contact forces. In our opinion, it severely affects the realism of the simulation but works quite robustly. In our experience, it seems that one can get away with only dampening the entries corresponding to friction forces.

Rather than regularizing the $\mathbf{A}$-matrix, one could regularize the bounds. We have experienced positive results when applying a lazy evaluation of the bounds inside the subspace solver loop. Thus, having slightly

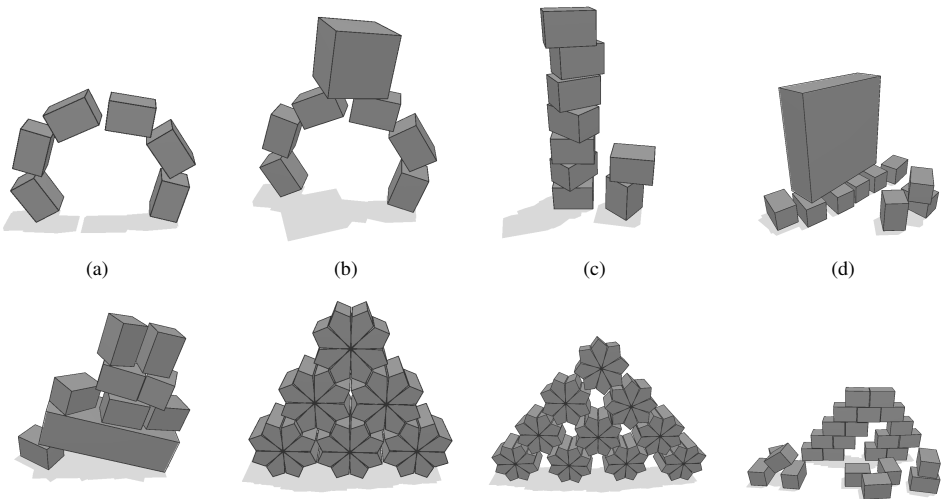


Figure 2: Illustrated test cases used for the PGS–SM method: (a) An arched snake composed of boxes and hinge joints with limits, (b) A heavy box placed upon an arched snake, (c) A large stack of boxes of equal mass, (d) A heavy box resting on lighter smaller boxes, (e) Boxes resting on an inclined surface resulting in static friction forces, (f) A small pyramid of gears, (g) A medium-scale pyramid of gears, (h) A large configuration of boxes stacked in a friction inducing manor.

relaxed bounds appear to add some freedom in reaching proper friction forces. On the downside, it appears to make the solver favor static friction solutions. We leave this idea for future work.

One final variation we will mention, is the staggered approach to the contact force problem. The approach is conceptually similar to (Kaufman et al., 2008). The idea is an iteration-like approach. First solve for normal forces assuming fixed given friction forces, and secondly solve for frictional forces assuming fixed given normal forces. The advantage of the staggered approach is that each normal and friction sub-problem has constant bounds, thus the NCP formulation is trivially reduced to a boxed MLCP, equivalent to a LCP. Given the properties of the **A**-matrix this guarantees solutions exist for the sub-problems. However, whether the sequence of sub-problems will converge in the staggered approach is hard to say. We have not observed any conclusive results on using a staggered approach.

6 CONCLUSION

A Projected Gauss–Seidel subspace minimization (PGS–SM) method has been presented, evaluated and compared to the Projected Gauss–Seidel (PGS) method for interactive rigid body dynamics. The PGS–SM method is stable for small sized configurations with large mass ratios, static friction and bilateral joints subject to limits. For medium and larger sized configurations, the PGS–SM method deteriorates into convergence behavior similar to the PGS method. Still, the PGS–SM method shows qualitatively different appearance in the simulations. For larger configurations, the PGS–SM method may be subject to simulation instability. In our opinion, our investigations indicate a more fundamental problem with the nonlinear complementarity problem (NCP) formulation of the contact force problem. We speculate that existence of solution is vital when accurate computations are performed. The minimum norm nature of the PGS method handles such cases robustly, although not very accurately.

Future work may include investigation into the nature of the NCP formulation, addressing existence of solutions. A more practical viewpoint would be exploring various iterative solvers for the reduced problem, as well as regularization ideas for the NCP formulation. In particular, we find the lazy evaluation of friction bounds appealing.

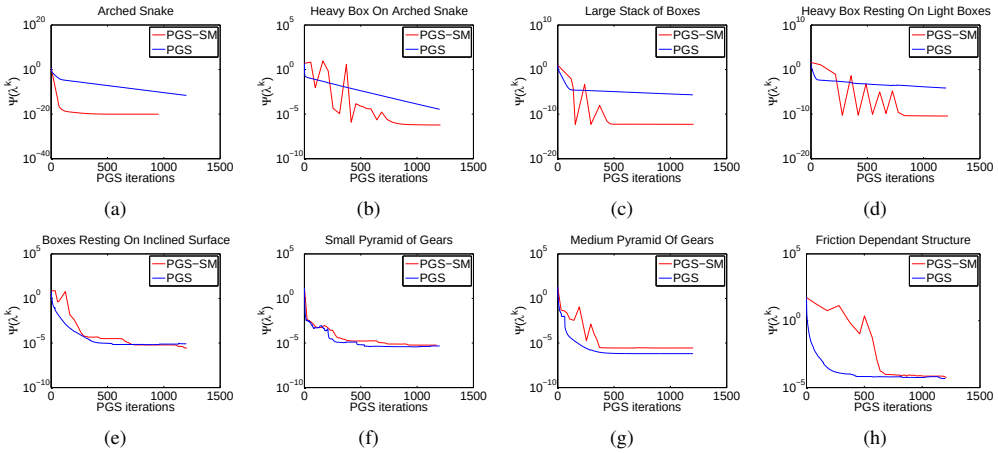


Figure 3: Corresponding convergence plots for the test cases in Figure 2.

REFERENCES

Anitescu, M. and Potra, F. A. (1997). Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlinear Dynamics. An International Journal of Nonlinear Dynamics and Chaos in Engineering Systems*.

Arechavaleta, G., E.Lopez-Damian, and Morales, J. (2009). On the use of iterative lcp solvers for dry frictional contacts in grasping. In *International Conference on Advanced Robotics 2009, ICAR 2009*.

Baraff, D. (1994). Fast contact force computation for nonpenetrating rigid bodies. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*.

Billups, S. C. (1995). *Algorithms for complementarity problems and generalized equations*. PhD thesis, University of Wisconsin at Madison.

Cottle, R., Pang, J.-S., and Stone, R. E. (1992). *The Linear Complementarity Problem*. Academic Press.

Erleben, K. (2007). Velocity-based shock propagation for multibody dynamics animation. *ACM Trans. Graph.*, 26(2).

Erleben, K. and Ortiz, R. (2008). A Non-smooth Newton Method for Multibody Dynamics. In *American Institute of Physics Conference Series*.

Featherstone, R. (1998). *Robot Dynamics Algorithms*. Kluwer Academic Publishers, second printing edition.

Guendelman, E., Bridson, R., and Fedkiw, R. (2003). Nonconvex rigid bodies with stacking. *ACM Trans. Graph.*

Hahn, J. K. (1988). Realistic animation of rigid bodies. In *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques*.

Kaufman, D. M., Sueda, S., James, D. L., and Pai, D. K. (2008). Staggered projections for frictional contact in multibody systems. *ACM Trans. Graph.*, 27(5).

Milenkovic, V. J. and Schmidl, H. (2004). A fast impulsive contact suite for rigid body simulation. *IEEE Transactions on Visualization and Computer Graphics*, 10(2).

Mirtich, B. V. (1996). *Impulse-based dynamic simulation of rigid body systems*. PhD thesis, University of California, Berkeley.

Moore, M. and Wilhelms, J. (1988). Collision detection and response for computer animation. In *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques*.

Morales, J. L., Nocedal, J., and Smelyanskiy, M. (2008). An algorithm for the fast solution of symmetric linear complementarity problems. *Numer. Math.*, 111(2).

O'Sullivan, C., Dingliana, J., Giang, T., and Kaiser, M. K. (2003). Evaluating the visual fidelity of physically based animations. *ACM Trans. Graph.*, 22(3).

Redon, S., Kheddar, A., and Coquillart, S. (2003). Gauss least constraints principle and rigid body simulations. In *In proceedings of IEEE International Conference on Robotics and Automation*.

Silcowitz, M., Niebe, S., and Erleben, K. (2009). Nonsmooth Newton Method for Fischer Function Reformulation of Contact Force Problems for Interactive Rigid Body Simulation. In *VRIPHYS 09: Sixth Workshop in Virtual Reality Interactions and Physical Simulations*, pages 105–114. Eurographics Association.

Stewart, D. E. (2000). Rigid-body dynamics with friction and impact. *SIAM Review*.

Stewart, D. E. and Trinkle, J. C. (1996). An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *International Journal of Numerical Methods in Engineering*.

Trinkle, J. C., Tzitzoutis, J., and Pang, J.-S. (2001). Dynamic multi-rigid-body systems with concurrent distributed contacts: Theory and examples. *Philosophical Trans. on Mathematical, Physical, and Engineering Sciences*.