Proiect cofinanţat din Fondul Social European prin Programul Operaţional Sectorial Dezvoltarea Resurselor Umane 2007-2013
Investeşte în oameni!

Proiect InnoRESEARCH - POSDRU/159/1.5/S/132395
Burse doctorale și postdoctorale în sprijinul inovării și competitivității în cercetare

UNIVERSITATEA **POLITEHNICA** DIN BUCUREŞTI
Facultatea de Automatică şi Calculatoare
Catedra de Calculatoare

Nr. Decizie Senat 243 din 12.05.2016

REZUMAT
TEZĂ DE DOCTORAT

Simularea unificată a corpurilor rigide și flexibile folosind dinamica bazată pe constrângeri

Unified Simulation of Rigid and Flexible Bodies Using Constraint Based Dynamics

Autor: ing. Mihai Frâncu
Conducător de doctorat: prof. dr. ing. Florica Moldoveanu

COMISIA DE DOCTORAT

Preşedinte	Prof. dr. ing. Adina FLOREA	de la	Universiteatea POLITEHNICA Bucureşti
Conducător de doctorat	Prof. dr. ing. Florica MOLDOVEANU	de la	Universiteatea POLITEHNICA Bucureşti
Referent	Prof. dr. ing. Vasile MANTA	de la	Universitatea Tehnică „Gheorghe Asachi” din Iași
Referent	Prof. dr. ing. Dan NEGRUȚ	de la	University of Wisconsin-Madison, SUA
Referent	Prof. dr. ing. Ioan SLUȘANSCHI	de la	Universiteatea POLITEHNICA Bucureşti

Bucureşti 2016

Rezumat

Această teză prezintă un cadru unificat atât teoretic cât și practic de simulare a corpurilor rigide și flexibile. El folosește teoria dinamicii bazată pe constrângeri împreună cu progrese mai recente în ecuații diferențiale algebrice și inegalități variaționale diferențiale. Un efort însemnat este pus în exprimarea tuturor modelelor de material ca o singură problemă neliniară de minimizare constrânsă. Sunt derivate rezolvatoare noi, sunt folosite noi metode de integrare precum Newmark și este propusă o nouă metodă de amortizare.

Un alt pilon al cercetării noastre este dinamica bazată pe poziții (PBD). În această teză demonstrăm ca metoda este fizic corectă, este echivalentă cu integrarea Euler implicită și poate fi reformulată ca o minimizare pornind de la principiile variaționale ale mecanicii. Mai mult, când este folosită regularizarea, constrângerile devin mai slabe și complet echivalente cu forțe elastice integrate implicit. Aceasta este și baza pentru noul nostru rezolvator de element finit ce se bazează pe proiecția pozițiilor și este corect din punct de vedere fizic.

În cele din urmă, tratăm subiectul contactului cu frecare în contextul dinamicii nonnetede. Arătăm că tratatea contactului de către PBD este de fapt o iterație de punct fix a deja doveditei scheme de pășire în timp la nivel de viteză. De asemenea, demonstrăm convergența acestei iterații și includem și un model mai riguros de frecare. Suntem primii care exprimă PBD ca o problemă de minimizare neliniară și convexă cu constrângeri conice. Această formulare include toate tipurile de constrângeri bilaterale și, mai ales, contactul cu frecare.

Abstract

This thesis presents a unified framework at both theoretical and practical level for simulating rigid and flexible bodies. It uses constrained dynamics theory together with more recent advances in differential algebraic equations and differential variational inequalities. A particular effort is put into expressing all the material models as a single nonlinear constrained minimization problem. New solvers are derived, new integration methods like Newmark are used and a new damping method is proposed.

Another pillar of our research is position based dynamics (PBD). In this thesis we prove that the method is physically correct, it is equivalent to implicit Euler integration and can be recast as a minimization starting from the variational principles of mechanics. Moreover, when regularization is employed the constraints become softer and fully equivalent to implicitly integrated elastic forces. This is also the basis for our novel physically correct finite element solver that relies on position projection.

Finally, we treat the subject of contact with friction in the context of nonsmooth dynamics. We show that PBD contact handling is in fact a nonlinear fixed point iteration of the established velocity time stepping scheme. We also prove convergence of this iteration and include a more rigorous friction model. We are the first to express PBD as a convex nonlinear minimization problem with conic constraints. This formulation encompasses all types of bilateral constraints and, more importantly, nonsmooth frictional contact.

Acknowledgments

I would like to thank my advisor for accepting me for this PhD and giving me the opportunity and freedom to work on the topics I had in target. I am also grateful to my girlfriend Cristina for standing by my side during this low-budget academic break from the industry. Also many thanks to professors Dan Negruț and Mihai Anițescu for their willingness to talk to me and help me effectively. Much appreciation goes to Alin Dumitru and Liviu Dinu from Static VFX for their help on rendering and their nice company. I also had very productive chats with Hammad Mazhar, many of which produced actual results, and I thank him for that. There are also other people that have helped me in some way or another during this thesis, even if with just a piece of advice or a nice conversation, and I hope I can mention them all: Kenny Erleben, Victor Asavei, Anca Morar, Lucian Petrescu, Teodor Cioacă, Horea Cărămizaru, Andrei Craifăleanu, Sergiu Crăițoiu, Vasile Brovcenco, Alessandro Tasora, Dario Mangoni.

The work has been partially funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Ministry of European Funds through the Financial Agreement POSDRU/159/1.5/S/132395.

Contents

Chapter 1

Introduction

This thesis describes and extends a number of methods for mechanical simulation in the context of computer graphics. The theory behind these methods belongs to a relatively narrow subfield of mechanics called *constrained dynamics*, i.e. dynamics with added constraints. In particular, we are focusing mainly on position projection methods modeled as mathematical optimization problems.

1.1 Motivation

The author's background is in games and physics for 3D games. This is why in this thesis a strong emphasis is laid upon real time applications. Also, constrained dynamics is often the method of choice of most rigid body physics engines. This is not always true for deformable bodies where different methods are used. This is what lead us to the idea of unification of methods so that a single engine is used for all simulations and also coupling is achieved between rigid and flexible materials.

1.2 Context

This work was elaborated in the context of computer graphics with a focus on virtual reality and interactivity. In contrast to engineering we perform no validation, but rely only on visual inspection and plausibility. Still, we did our utmost best to start from the same physical principles and obey the same physical laws as the other older and accepted methods used in engineering and scientific simulations.

We are focusing only on constraint based methods in order to improve them. We are also comparing them to more traditional elasticity based approaches. For solving the numerical problem we focus on mathematical optimization methods and in particular on iterative solvers and gradient descent approaches. Going deeper, we concentrate only on the dual formulation of the problem in terms of Lagrange multipliers and matrix-free formulations for quick, parallel and low memory footprint implementations.

1.3 Goals

As already hinted, our main goal was to simulate as many different phenomena possible using the framework of constraints. This thesis was intended as a proof of concept and limited itself to coupling rigid and deformable bodies inside the same solver.

Another challenge we set ourselves was to make these methods suitable for real-time and interactive applications. Our intuition was that if constraint methods proved so successful in games and VFX then they are clearly the way to go and should be extended.

The biggest condition for reaching interactive frame rates is that our algorithms are parallel and scale well with the number of computing units. This is why we set ourselves the goal to develop new and competitive solvers that work well on multi-core and many-core (i.e. GPGPU) architectures directly from the mathematical level.

1.4 Thesis outline

- **Chapter 2** does a quick survey of the state of the art and also mentions some historical details.
- **Chapter 3** gives a brief overview of the equations of motion for particles, rigid bodies and continua. Some focus is put on Lagrangian mechanics and the kinematics of rotation.
- **Chapter 4** treats the subject of numerical integration and gives short descriptions of the types of integrators. We then focus on the implicit Euler integrator and present its minimization form and our new Nonlinear Conjugate Gradient solver.
- **Chapter 5** is intended to be an introduction to the practices of modeling different types of objects and materials, so that we can focus solely on the mathematics and numerical methods in the following chapters.
- **Chapter 6** introduces the ideas of constrained dynamics with an emphasis on equality only and holonomic constraints, namely on differential algebraic equations. Our contributions include a new parallel solver, proven equivalence between elasticity and regularized position projection and an accurate position based finite element solver.
- **Chapter 7** gives a brief overview of nonsmooth dynamics, namely the necessity of using impulses to handle impacts and friction. Then we proceed to presenting our new nonlinear and fully implicit approach to nonsmooth dynamics yielding a rigorous formal model of position projection with frictional contact.
- **Chapter 8** describes the actual theoretical foundations and practical details to build our unified simulator. We also added some numerical experiments we performed during our research to analyze some issues in isolation.
- **Chapter 9** gives the final conclusions and summarizes our work. We enumerate once more our contributions and finally list quite a few issues that are still outstanding and represent challenges for the future.

1.5 Publications in connection with this thesis

A good part of the research presented in this thesis was also published in the following articles:

- *Minimum residual methods for cloth simulation* [36]
- *An Improved Jacobi Solver for Particle Simulation* [35]
- *Cloth Simulation Using Soft Constraints* [37]
- *Virtual Try On Systems for Clothes: Issues and Solutions* [38]
- *Simulating Large Scale Coupled Granular Material Simulations using Position Based Dynamics* [66]

Chapter 2

Related work

Now that we clearly defined the context of our thesis we can restrict ourselves to reviewing only a subset of the existing dynamics simulation literature. This chapter will only cover very high level aspects of the state of the art in the topics connected to our research.

2.1 State of the art

Rigid bodies with contact can be simulated with constrained dynamics and other methods, most importantly the penalty method [12]. Other approaches consider the rigid body as a collection of particles and contact forces are computed using the discrete element method (DEM) or in other ways [91, 53]. The particles can move under rigid transformations [47] or be constrained together to form a composite rigid object [31, 62].

Granular matter has been an area of research in computational mechanics for decades. The method of choice is usually the *discrete element method* (DEM) which treats the granules as elastic billiard balls and uses Hertzian contact theory [39]. The DEM method was used in graphics too [16, 2, 47]. Another approach was a continuum based one, considering the granular matter a special kind of fluid [100, 74]. This was followed by a Lagrangian version derived from the *smooth particle hydrodynamics* (SPH) method for simulating fluids [1, 51]. An alternative to DEM is the nonsmooth dynamics approach where the particles are considered fully rigid and this is the path we are following.

Deformable bodies have been traditionally simulated using implicit integrators due to their unconditional stability properties. These have been applied not only to mass-spring systems, but also to simulations using the *finite element method* (FEM) [73]. Recently, the popular Backward Euler integrator has been recast as an optimization problem [23] helping us to gain new insights on a problem that used be solved solely by the Newton method.

Cloth is one example of a deformable body modeled as a mass-spring system [79]. The implicit integration of the equations of motion has become pervasive for cloth since the seminal work of [11]. Its main attraction is its unconditional stability for very stiff equations and large time steps. In games *position based dynamics* (PBD) is usually preferred for simulating cloth [53, 72, 41].

Constrained dynamics was not initially considered for simulating deformable bodies, but this changed with the advent of PBD and constraint regularization [83]. PBD was originally introduced by Jakobsen for games based on molecular dynamics methods and a nonlinear version of the Stewart-Trinkle solver for rigid bodies [53]. Goldenthal later showed that position projection is equivalent to the fully implicit integration of a constrained system [40].

Constraint based methods appeared originally in their acceleration based formulation for rigid body dynamics with joints and contacts [9, 25]. Later on, velocity or impulse based methods gained more popularity [5, 33]. Position based methods are actually a nonlinear version of velocity based ones, in the sense that they can still be expressed as velocity filters, but constraints are enforced at positional level [90]. Part of the inspiration for PBD came from molecular dynamics where methods like SHAKE or RATTLE are widely used [13]. A more detailed study for the application to cloth simulation in computer graphics was done in [40]. Here the method of *fast projection* is developed based on an implicit treatment of constraint directions [50] and a better energy preserving integrator is also derived. Position based methods rely on *projection* for solving differential algebraic equations (DAE), which is ultimately an optimization problem [46]. Another part of inspiration came from *strain limiting* techniques used in elastic cloth simulation [79, 26].

Constraint *regularization* was employed mainly in [57] for making rigid dynamics with contact and friction more tractable numerically. We take the name *soft constraints* from [28] where an older idea is used: regularization under the mask of Constraint Force Mixing (CFM) [88]. Recently constraint regularization has been used for particle based fluid simulation [61]. Another application was intended for the simulation of deformable elastic models using a constraint based FEM formulation [83]. Similar position based approaches can be found in [23] and [18]. The FEM constraint approach is similar in philosophy with *continuum strain limiting* [93, 71].

Iterative methods are currently the preferred way of solving constrained mechanical systems for real-time. Using exact methods can become infeasible when adding contact and friction for more than a few hundred bodies [17]. The fastest and most robust iterative method used in the present is Gauss-Seidel (GS) [27, 33]. GS also knows improvements such as line search with conjugate directions [86] or subspace minimization [87]. Jacobi is another relaxation method, closely resembling GS, but it converges slower and needs modifications to remain stable. Still it is preferred to GS for parallel implementations as it can process each constraint independently from the others [96].

The Conjugate Gradient (CG) method has a good reputation for solving linear systems as it has better convergence than matrix splitting methods like Jacobi or GS [82]. Even though it was used for implicit integration of mass-spring models [11] it has never gained traction in constrained dynamics simulations. There have been attempts at using it [80], but many argued against its applicability for different reasons [33, 95, 69]. Our approach is based on a minimum residual variant of gradient descent algorithms as it guarantees decreasing residual energy and is more stable. After optimizing the conjugate residuals algorithm we arrived at a version of Jacobi with improved convergence. A minimum residual method (GPMINRES) was also used in [49]. The line search Jacobi algorithm offers similar improvements to ours [96, 29], but our addition of a momentum term bears more resemblance to Nesterov's method [67].

2.2 Nonsmooth dynamics

The formulation of contact dates back to Signorini in the context of elasticity. The solution to the problem of contacting elastic bodies was given in the early 60s and contact complementarity conditions were named Signorini-Fichera (or Moreau-Signorini) [99]. The application of this conditions to particles and rigid bodies was done in the 70s and 80s mainly by Moreau and Monteiro-Marques [70]. The work of Moreau was later continued by the likes of Jean, Jourdan, Alart, Curnier and others. In this thesis we are following mostly the work of Anițescu and his collaborators and also variants of it that made it into computer graphics and games.

Chapter 3

Equations of motion

We present the equations of motion in the Newton and Lagrange formulations. We introduce Hamilton's principle and set it as a foundation of later results in this thesis. Rotation kinematics and the Newton-Euler equations of the dynamics of the rigid body are briefly presented and in the end we give a quick guide to continuum mechanics.

3.1 Newton equations of motion

Before mechanics came geometry as a physical science, so it is natural to measure the movement of a particle as the variation of its *position*. The shape drawn by this point at different moments in time is called a *trajectory* and can always be defined as a parametric curve: $\vec{x}(t), t \in \mathbb{R}$. Newton's form of equations of motion give us a law of the variation of momentum when external forces are present. If considering the mass constant and using acceleration we get a second order *ordinary differential equation* (ODE):

$$\dot{\vec{p}} = m\ddot{\vec{x}} = \vec{f}(\vec{x}, \vec{v}, t). \tag{3.1}$$

In the end the first order equations that we will employ throughout this thesis are:

$$\frac{d}{dt}\begin{pmatrix}\mathbf{x}\\ \mathbf{M}\mathbf{v}\end{pmatrix} = \begin{pmatrix}\mathbf{v}\\ \mathbf{f}(\mathbf{x}, \mathbf{v}, t)\end{pmatrix}, \tag{3.2}$$

with $\mathbf{x}(t_0) = \mathbf{x}_0$ and $\mathbf{v}(t_0) = \mathbf{v}_0$ as initial conditions.

3.2 Lagrange equations of motion

The alternative procedure of Lagrange implies identifying from the start the degrees of freedom of the system and is often called a *reduced coordinates* formulation. These reduced coordinates are also known as *generalized coordinates* $\mathbf{q} \in \mathbb{R}^n$, where n is the number of degrees of freedom. They coincide with the aggregated coordinates $\mathbf{x}$ from the previous section when motion is unconstrained and Cartesian coordinates are used. *Generalized velocities* are simply denoted as $\dot{\mathbf{q}}$. Kinetic energy of a particle is $T = mv^2/2$ and we can also define a potential energy V such that the force acting on the particle is a gradient of this potential, i.e. $\vec{f} = -\nabla V$. We introduce the Lagrangian function as the difference between the kinetic and the potential energy: $\mathcal{L} = T - V$.

For a system of particles (and even in general) the kinetic energy has the form $T = \frac{1}{2}\mathbf{v}^T\mathbf{M}\mathbf{v}$, where $\mathbf{M}$ is the positive-definite mass matrix and the potential energy is a function only on positions $V(\mathbf{x})$. Then the Euler-Lagrange equations of motion are:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}}\right) = \frac{\partial \mathcal{L}}{\partial \mathbf{q}}. \tag{3.3}$$

The real power of these equations comes from the fact that they are equivalent to a more general extremum principle known as Hamilton's principle:

$$\text{extremize } S = \int_{t_1}^{t_2} \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}, t) dt. \tag{3.4}$$

3.3 Rotation kinematics

Surprisingly, rotations bring most of the complexity to the dynamics of rigid bodies and give birth to intricate mathematical constructs. There is a broad literature on the representation of rotation[42, 34, 64]. The most common representation comes from linear algebra and is a 3 by 3 orthonormal matrix: $\mathbf{R} \in \mathbb{R}^{3\times 3}, \mathbf{R}^T\mathbf{R} = \mathbf{1}_3$.

The kinematic equation in the rotation matrix representation is:

$$\dot{\mathbf{R}} = \vec{\omega}^{\times}\mathbf{R}, \tag{3.5}$$

where $\vec{\omega}$ is the angular velocity and the $^{\times}$ symbol means the skew-symmetric matrix associated with a vector [10].

However, we will be dealing with this quaternion kinematic equation mostly:

$$\dot{\xi} = \tfrac{1}{2}\vec{\omega} \circ \xi, \tag{3.6}$$

where $\circ$ denotes quaternion multiplication.

3.4 Newton-Euler equations for rigid bodies

In order to be able to write the rigid body equations of motion we need to introduce a few more physical quantities. The most important one is the *angular momentum*, which as the name states is the analog of linear momentum: $\vec{L} = \mathbf{I}\vec{\omega}$, with

$$\mathbf{I} = \sum m_i (r_i^2 \mathbf{1} - \vec{r}_i \vec{r}_i^T). \tag{3.7}$$

The symmetric matrix $\mathbf{I}$ is called the *inertia tensor* and it is the analog of mass for rotations.

The Newton-Euler equations in aggregated Cartesian coordinates are:

$$\tilde{\mathbf{M}}\dot{\mathbf{v}} = \mathbf{f}(\mathbf{x}, \mathbf{v}), \tag{3.8}$$

$$\mathbf{I}\dot{\boldsymbol{\omega}} = \boldsymbol{\tau} - \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega}, \tag{3.9}$$

where $\tilde{\mathbf{M}} = \text{diag}(m_i)$, $\boldsymbol{\tau}$ is the external torque and the relationship between linear and angular velocities (and corresponding generalized positions) is given by the kinematic equations.

3.5 Continuum mechanics

In order to be able to define strain we need to introduce the deformation field and its gradient. Thus for every point in the original undeformed object $\vec{r}$ there corresponds a displaced point $\vec{x} = \phi(\vec{r}) = \vec{r} + \vec{u}(\vec{r})$, where u is the displacement. The gradient $\mathbf{F} = \nabla\phi$ indicates the amount of deformation and also of accumulated elastic energy [85]. We measure this by seeing how far off is $\mathbf{F}$ from being orthonormal and we obtain the Green-Lagrange nonlinear strain:

$$\boldsymbol{\varepsilon}_G = \tfrac{1}{2}(\mathbf{F}^T\mathbf{F} - \mathbf{1}). \tag{3.10}$$

Now that we know about strain, we can move on to the *stress* tensor which is the analog of force. Together with strain they form an analog of Hooke's law:

$$\boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\varepsilon}, \tag{3.11}$$

where $\mathbf{C}$ is rank 4 symmetric tensor defining material properties.

The energy density function is:

$$\Psi(\mathbf{F}(\vec{x})) = \boldsymbol{\varepsilon}^T\mathbf{C}\boldsymbol{\varepsilon}. \tag{3.12}$$

The total energy is the integral of Ψ over the whole volume and the stress is $\boldsymbol{\sigma} = \frac{\partial\Psi}{\partial\boldsymbol{\varepsilon}}$.

Chapter 4

Time discretization

Time discretization means basically numerical integrators for ODEs. They are the heart of a simulator as they are the only way of advancing trajectories in time.

4.1 Numerical integration

We are usually interested in two properties of the integrators: accuracy and stability. Accuracy is generally denoted by the order of the integrator, meaning the power of the time step to which the error is proportional. In this thesis we do not use order higher than two. Stability is a more delicate issue and refers to keeping the error bounded over a long integration time. There are two big classes of integrators: explicit and implicit, and only the latter can guarantee unconditional stability.

4.2 Variational and symplectic integrators

The subject of variational or geometric or symplectic integration is currently under development by mathematicians. The three words have almost similar meaning and originate from properties of analytical mechanics: the variational principles of mechanics [58], the geometric view on mechanics [46] and the symplectic property of Hamiltonian phase space [7, 59]. In a nutshell, variational integration starts from the very basic principles of mechanics and develops a subset of integrators that fit with the nature of dynamical systems. The main goal is to conserve energy and momentum as much as possible irrespective of the integrator order.

4.3 Integration as minimization

Relatively recently the Implicit Euler integrator was recast as a minimization problem [60, 23, 65]:

$$\text{minimize } \tfrac{1}{2}(\mathbf{v}^{l+1})^T \mathbf{M}\mathbf{v}^{l+1} + V(\mathbf{x}^{l+1}). \tag{4.1}$$

At this point we can introduce one of our contributions, which is to apply a nonlinear minimization algorithm directly on (4.1) instead of solving the nonlinear optimality conditions with Newton's method [78] or in a linearly implicit fashion as it was customary [11]. We used a Nonlinear Conjugate Gradient (NCG) algorithm [84] and found that we could ignore the second derivative of the potential in a stable and accurate manner (see [37]).

Chapter 5

Material models

This chapter is in fact about space discretization. If this concept does not apply so much to particles and rigid bodies, this is not the case for deformable bodies where the choice of discretization makes a world of difference between different simulation results.

5.1 Rigid bodies

Rigid bodies are large solid objects (much larger than a particle) that have shape and a distribution of mass. A rigid body has 6 degrees of freedom: 3 for translation and 3 for rotation. In practice we can use somewhere between 6 and 12 parameters to describe the generalized position of a rigid body.

An important type of interaction between rigid bodies is contact, also known as collision or impact. We will give a different meaning to each of these words. By contact we will mean a type of constraint that prevents bodies from inter-penetrating (especially when they are resting on top of each other). Contact is mostly persistent, manifesting friction in the tangential plane (stick), but it can also break (slip). Collision is a word that we will reserve mostly for the phase of collision detection, as it has caught on in time and it refers to testing for intersection between objects and obtaining contact information. Impact refers almost exclusively to high speed collisions that usually result in the objects moving apart immediately after (i.e. elastic or partly elastic impacts).

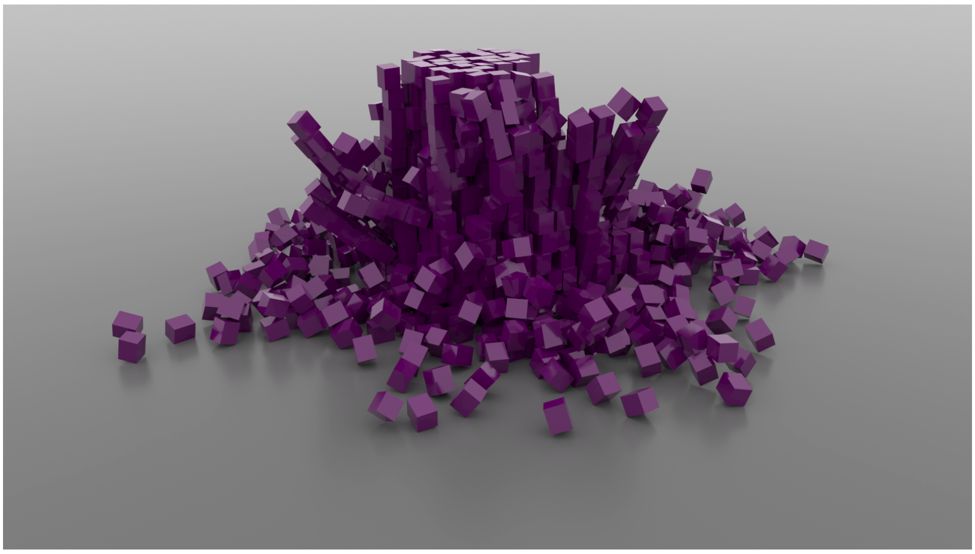


Figure 5.1: Boxes falling on ground solved with position projection.

5.2 Elasticity

In this section we address elastic materials. Bodies made of such materials are called (at least in our context) deformable, flexible or soft bodies. There are many ways to model deformable bodies and they all involve some kind of approximate spatial discretization. Just like in the case of integration in time, the accuracy depends on the size of the discretization resolution.

One of the most popular methods of discretization used in computer graphics consists of particle systems interconnected by linear springs or other forces. The other two popular approaches to modeling elastic bodies are finite differences and finite element.

5.3 Threads

Threads are most often modeled as a chain of particles connected through springs. These springs can be either integrated implicitly or treated as hard links in a constraint approach. More complex models involve articulated bodies [44], Cosserat theory [75], supper-helices [20] and nonsmooth dynamics [21]. In addition, threads need modeling of torsion [56, 44].

5.4 Cloth

Cloth is a generic name we give in computer graphics to thin surfaces or shells or membranes with applications mainly in garment simulation. There is a whole branch of study in mechanical engineering about this type of objects [14], but we are not entering this kind of detail. We focus mainly on the mass-spring approximation and at the end offer an accurate alternative using FEM.

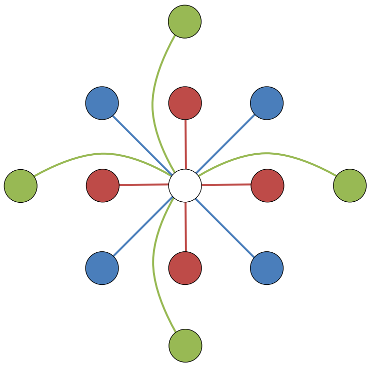


Figure 5.2: Links structure of cloth relative to one particle (white one in the center): stretch links (red), shear links (blue), bend links (green).

As you can see in Figure 5.2 there are mainly 3 types of springs or links inside cloth. The stretch links are structural as they hold the fabric together and also determine its extension and compression properties, i.e. its tensile strength. Shear links have the role of preventing skewing in the plane of the cloth; they are only an approximation of shear stresses that arise in continuous materials. Bending links join more distant neighbors and, as the name says, they prevent bending of the cloth (out of plane deformation). Bending links can be replaced by other type of deformation functions (e.g. dihedral angle [11, 72], curvature [55]) or potentials [19].

Even though cloth was simulated in the past using finite differences, the most popular way of modeling cloth accurately nowadays is the finite element method [97, 94]. The

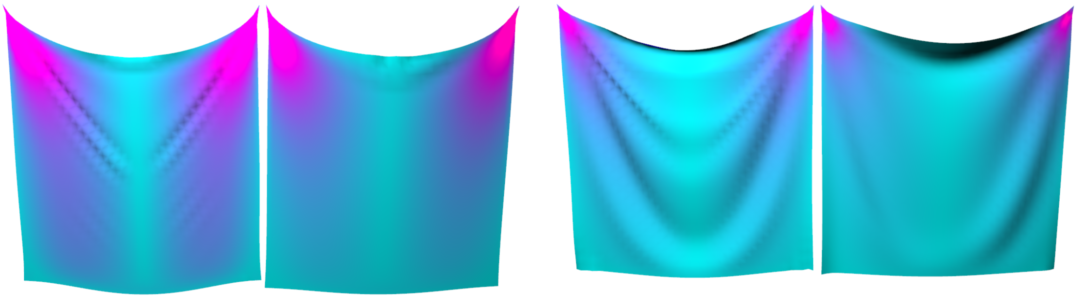


Figure 5.3: Two snapshots of a side by side real-time simulation of two 40×40 cloth pieces with the same Young's modulus E: regularized FEM constraints (left) and soft links (right).

continuum formulation in [11] is actually a particular case of the finite element method. You can find our own constraint based approach in [37] (see also Figure 5.3).

5.5 Deformable bodies

In computer graphics there are many ways in which deformable bodies can be "tricked". We leave aside animation techniques and focus only on dynamics simulation, but even then one can use mass-springs lattices, cloth balloons, shape matching, example based deformation and the list can continue. In engineering on the other hand, where accuracy is important, the state of the art method is FEM.

5.6 Fluids

Fluids make up a special category of study in simulation. In engineering the field is called *computational fluid dynamics* (CFD) and it usually entails very complex and accurate methods. In computer graphics fluid simulation is also very popular and is used for a lot of special effects (VFX) and computer generated imagery (CGI).

The standard SPH method has been extended in the recent years to an implicit version with better incompressibility. Two of these methods, constraint fluids [22] and position based fluids are strongly connected to the formalism of constrained dynamics. We think that the PBF simulations in [62] are the best examples of how fluids can be coupled in the same solver with other position-based methods including the ones presented in this thesis.

5.7 Granular matter

Granular matter is a material with very curious properties as it sometimes behaves like a solid and other times like a fluid [52]. For example, due to static friction piles of grains maintain their shape, but if the angle of repose is reached avalanches can be triggered. For similar reasons sand in an hourglass flows at a steady rate as the pressure does not increase indefinitely with height as in the case of hidrostatic pressure.

Contact can be handled in two ways: the *discrete element method* (DEM) [16], which is basically Hertzian contact theory, and complementarity approaches (including nonsmooth dynamics).

5.8 Collision detection

There are two important types of collision detection: discrete and continuous (CCD). The former is usually implied and it is most used as it is easier to implement. Discrete means that the penetration tests are done at instants of time and continuous implies that the time of impact is identified precisely in between two moments in time. CCD was developed mainly to prevent bullet through paper (or tunneling) artifacts when collisions are missed due to high speeds. This is why CCD is mainly used for self-intersection tests for cloth where it is very hard to tell locally which side is the correct one and there is no notion of body interior.

Collision detection has three stages: broadphase, midphase and narrowphase. Broadphase is the stage when only potential collision candidates are identified based on simple bounding volume tests. These volumes are usually *axis aligned bounding boxes* (AABB) and one of the most popular methods used for broadphase is *sweep and prune* (SAP) [32]. Midphase is optional and it usually means testing two large composite objects against one another. They usually imply using bounding volume hierarchies (BVH) for each body and doing tree traversals in order to do the overlap tests. Finally, narrowphase means doing the exact test between two relatively simple objects (e.g. boxes, spheres, cylinders, capsules, triangles, convex polyhedra).

Chapter 6

Constrained dynamics

Constraints are the stiffness limit of strong elastic forces. The method of choice for solving constraints is that of Lagrange multipliers. We focus mostly on solvers that are more in vein to nonsmooth or impulse based dynamics. We also go a step further by working directly at position level where we present our contributions.

6.1 Constraints

Constraints usually appear in the context of analytical mechanics rather than the Newtonian one, where everything is solved by determining forces. Our procedure increases the number of variables rather than reducing it (redundant coordinates), by adding the so called *Lagrange multipliers*.

Constraints have different classifications. Unilateral constraints involve inequality as they usually mean that a body should stay on one side of a surface at any time, i.e. contact constraint. Bilateral constraints are more popular in physics and they were given more space in the literature as they are easier to deal with than inequalities. Bilateral constraints have two distinctions: *scleronomic* vs. *rheonomic* and *holonomic* vs. *nonholonomic*. Scleronomic means the constraint does not depend on time explicitly while rheonomic signifies the opposite. Holonomic constraints depend only on generalized positions while nonholonomic ones involve velocities or, equivalently, position differentials. In this thesis however we focus mainly on holonomic and scleronomic constraints.

Key in the understanding of constraints is the *principle of virtual work* from Lagrangian mechanics. In the case of equilibrium, the virtual displacements $\delta\mathbf{q}$ must be compatible with the constraints, i.e. $0 = \delta\mathbf{c}(\mathbf{q}) = \nabla\mathbf{c}(\mathbf{q})^T\delta\mathbf{q}$, meaning that they always lie in the tangential plane of the constraint manifold. The principle of virtual work is extended from statics to dynamics by the principle of d'Alembert [58].

6.2 Differential algebraic equations

The most general form of a index 3 DAE encountered in dynamics is:

$$\mathbf{M}\ddot{\mathbf{x}} = \mathbf{f}_{ext} + \mathbf{J}^T\boldsymbol{\lambda}, \tag{6.1}$$

$$\mathbf{c}(\mathbf{x}) = 0, \tag{6.2}$$

where $\mathbf{f}_{ext}$ is the external force. An important property of the DAE is its index - a measure of singularity or perturbation: add one to the number of times it takes to derive the constraint function so that the unknown has the same derivative order as in the ODE [25, 45].

6.2.1 Mechanical engineering

Differential algebraic equations have been developed in the context of mechanisms where many rigid parts are joined by various joints in a complex manner. Popular solvers include DASSL [25], RADAU5 [45] or generalized coordinate partitioning [48] and orthogonalization methods [63]. Index reduction is a penalty approach which converts the DAE into an ODE [15]. This is basically a spring-damper regularization and it is used in modern DAE solvers as a correction technique (now called Baumgarte stabilization) [8].

6.2.2 Molecular dynamics

In computational molecular dynamics (MD) often very strong bonds are replaced by constraints. This is done in order to reduce the stiffness of the problem and speed up the computation. The algorithm SHAKE dates back to the 70s and it is based on a Verlet integrator and nonlinear equation solving [81]. It was later extended to Velocity Verlet and phase space projection: RATTLE [3]. These are both related to the method of projection for solving DAEs [46]:

$$\text{minimize } \|\delta\mathbf{x}\|^2_{\mathbf{M}} \text{ subject to } \mathbf{c}(\mathbf{x}^{l+1}) = 0, \tag{6.3}$$

where $\delta\mathbf{x} = \mathbf{x}^{l+1} - \tilde{\mathbf{x}}$ is the difference between the new unknown position $\mathbf{x}$ and an initial candidate position $\tilde{\mathbf{x}}$ obtained through unconstrained integration.

6.2.3 Computer graphics

In computer graphics constraints were introduced by Baraff and Witkin [9, 98]. Building from Stewart's method and using influences from molecular dynamics and DAE theory, Jakobsen introduced the first position based method in games [53]. Later Müller et al. [72] further developed the method, switched to a Symplectic Euler integrator and gave it its now popular name: *position based dynamics* (PBD). Goldenthal made it more academic and presented it in a different form (decoupled from the Gauss-Seidel solver) called *fast projection* [41, 40].

We tried to understand as much as possible where PBD really comes from and what it really is and came up with two equivalent answers:

- a position-based (index 3) DAE solver using projection or
- a nonlinear velocity time stepping (index 2) DAE solver with stabilization.

6.3 Velocity time stepping

The VTS method is used mainly for contact and friction, but it can also handle bilateral constraints [92, 33]. It is obtained by discretizing the equations of motion (6.1) using a semi-implicit (or symplectic) Euler integrator as in [90]. The velocity based method bears a lot of similarity to the *impulse based method* [68, 43] and therefore Catto called his Gauss-Seidel solver Sequential Impulses [27].

6.4 Nonlinear minimization

We experimented with iterative solvers from the Conjugate Gradient (CG) family [78, 84] to do projection. Most of them worked given enough iterations, but there were stability issues when not using the Minimum Residual (MINRES) type of algorithms. We found out that the problem was not related to the system matrix but rather to the way the residual evolved along the iterations. Our method of choice was in the end Conjugate Residuals (CR) [82].

The implicit Euler integrator has also been shown to have a minimization structure in Section 4.3. The formulation in (4.1) with constraints added gives us an objective function close to the one used in projection:

$$\text{minimize } \tfrac{1}{2h^2}\Delta\mathbf{x}^T\mathbf{M}\Delta\mathbf{x} + V(\mathbf{x}^{l+1}) \text{ subject to } \mathbf{c}(\mathbf{x}^{l+1}) = 0, \tag{6.4}$$

where $\Delta\mathbf{x} = \mathbf{x}^{l+1} - \mathbf{x}^l - h\mathbf{v}^l$. The main takeaway here is that projection is simply just the implicit Euler integration of constraint forces.

Stating position projection as a dual nonlinear optimization problem is a premiere to our knowledge and we think it is a very versatile form:

$$\text{minimize } \tfrac{h^2}{2}\boldsymbol{\lambda}^T\mathbf{A}\boldsymbol{\lambda} + \boldsymbol{\lambda}^T\mathbf{c}(\mathbf{q}^{l+1}), \tag{6.5}$$

where $\mathbf{A} = \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T$. Note that the constraint Jacobian is considered here at the end of the time step in an implicit manner: $\mathbf{J} = \nabla\mathbf{c}(\mathbf{q}^{l+1})$.

6.5 Variational minimization structure

We now make the case that most DAE solvers can be recast as minimization problems. To summarize, projection is a discretized way of expressing Hamilton's principle and expresses as a constrained minimization prolem both position and velocity based methods for solving DAEs. In a nutshell, all the methods can be expressed as a minimization which in turn can be recast to a projection on the constraint manifold or its tangent bundles. We followed the derivation in [54] to show how the Newmark integrator can be recast as a minimization problem.

6.6 Solvers

The two important classes of methods analyzed are relaxation solvers and Krylov subspace methods, better known as the conjugate gradient family of algorithms. At a closer scrutiny the two are quite similar as they both are line searches:

$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \alpha\mathbf{d}_k, \tag{6.6}$$

and at the same time gradient descent methods as the direction $\mathbf{d}$ is usually the gradient or something related [80].

6.6.1 Relaxation

Relaxation solvers are simple iterative methods for solving large sparse linear systems. They are also called *stationary* as the recursion formula does not change or *splitting* methods as they rely on splitting the system matrix in 2 or 3 components. They include the Jacobi, Gauss-Seidel and Successive Over-Relaxation (SOR) methods.

6.6.2 Krylov subspace methods

These methods include the well known Conjugate Gradient (CG) algorithm and its variants [84]. We will now mention briefly the *minimum residual* methods that were initially developed for indefinite symmetric matrices. These methods try to minimize the quadratic norm of the residual $\|\mathbf{r}\|^2$ [78]. The minimum residual counterpart of CG is the Conjugate Residuals (CR) algorithm. This is of importance because we found that the CR algorithm is more stable than CG when applied to constraint projection and for small number of iterations [36].

6.6.3 Accelerated Jacobi

Without giving all the details here, we obtained a version of Jacobi with increased convergence ($\omega \leq 1$):

$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k - \frac{\omega}{A_{ii}}\mathbf{r}_k - \beta_k \delta \boldsymbol{\lambda}_k, \tag{6.7}$$

After looking more closely at the accelerated projected gradient descent (APGD) or Nesterov's method in [67] we came to the conclusion that it can be reduced to the same form as in (6.7). We decided to use the same step size as in Jacobi and adapt the following formulas from [67]:

$$\theta_{k+1} = \tfrac{1}{2}(-\theta_k^2 + \theta_k\sqrt{\theta_k^2 + 4}), \theta_0 = 1 \tag{6.8}$$

$$\beta_{k+1} = \frac{\theta_k(1-\theta_k)}{\theta_k^2 + \theta_{k+1}}. \tag{6.9}$$

6.7 Regularization

Regularization can be expressed as adding a small feedback term to the constraint equation:

$$\tilde{\mathbf{c}}(\mathbf{x}) = \mathbf{c}(\mathbf{x}) + \epsilon\boldsymbol{\lambda} = \mathbf{0}. \tag{6.10}$$

If we denote by $\kappa = 1/\epsilon$ then we can see that this is equivalent to having a potential energy of the form $\frac{\kappa}{2}\|\mathbf{c}(\mathbf{x})\|^2$ which is precisely the elastic spring energy with the stiffness κ [57].

In [37] we showed that implicit Euler integration of elastic media is equivalent to regularized projection. For this we used the minimization formulation in (4.1). The potential term of the constraints V_c can be either a quadratic elastic energy or the Lagrange multiplier potential energy of the constraints. We rewrite equation (4.1) as:

$$\text{minimize } \tfrac{1}{2h^2}\Delta\mathbf{x}^T\mathbf{M}\Delta\mathbf{x} + V_c(\mathbf{x}) + V_{ext}(\mathbf{x}^l), \tag{6.11}$$

where V_c can be either $\frac{\kappa}{2}\|\mathbf{c}(\mathbf{x})\|^2$ or $-\boldsymbol{\lambda}^T\tilde{\mathbf{c}}(\mathbf{x})$.

6.8 Energy dissipation and damping

We have already mentioned that for preserving energy during the simulation it is best to use symplectic integrators (Section 4.2). This is why in [37] we propose a projection scheme based on the Newmark integrator that can be tuned.Another of our contributions is to add damping in a physical and credible manner to the position projection formulation. In general we can do this by using a Rayleigh dissipation function [57]:

$$\varphi(\mathbf{v}) = \tfrac{1}{2}\dot{\mathbf{c}}(\mathbf{q})\mathbf{R}\dot{\mathbf{c}}(\mathbf{q}), \tag{6.12}$$

where $\dot{\mathbf{c}}(\mathbf{q}) = \mathbf{D}^T\mathbf{v}$ and $\mathbf{R}$ is a positive definite matrix (usually diagonal). Using the dissipation potential in (6.12) yields a new regularization formula: $\mathbf{c}(\mathbf{q}) + \mathbf{C}^{-1}\mathbf{R}\dot{\mathbf{c}}(\mathbf{q}) + \mathbf{C}^{-1}\boldsymbol{\lambda} = 0$, which in turns gives a new KKT matrix and a new Schur complement: $\mathbf{A} = h(h\mathbf{1} + \mathbf{C}^{-1}\mathbf{R})\mathbf{D}^T\mathbf{M}^{-1}\mathbf{D} + \mathbf{C}^{-1}$.

6.9 Stability

Given that a part of the stiffness matrix is not implicitly taken into account, it is easy to see why the regularized velocity time stepping scheme might manifest instabilities above certain frequencies. This is mainly due to the linearization error - basically the second order term in the Taylor series expansion of the constraint is missing. For the infinite stiffness case we can still have residual error in the solver that can manifest itself as compliance or possibly unstable linearization error.

6.10 Constraint based FEM

For the particular case of the geometrically linear *finite element method* (FEM) we use the element energy in order to identify the constraint function: $V_e = W\tilde{\boldsymbol{\varepsilon}}^T\mathbf{C}\tilde{\boldsymbol{\varepsilon}}$, where W is the element volume, $\tilde{\boldsymbol{\varepsilon}}$ is the element constant strain in Voigt notation and $\mathbf{C}$ is a stress-strain relation matrix The constraint function is then:

$$\mathbf{c}(\mathbf{x}) = \sqrt{W(\mathbf{x})}\tilde{\boldsymbol{\varepsilon}}(\mathbf{x}). \tag{6.13}$$

Algorithm 1 Pseudo-code for computing the internal forces inside a tetrahedron. Here a block Gauss-Seidel approach is employed.

```
Input: tetrahedron $(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$
Compute shape matrix $\mathbf{D}_s = [\mathbf{x}_1 - \mathbf{x}_0 | \mathbf{x}_2 - \mathbf{x}_0 | \mathbf{x}_3 - \mathbf{x}_0]$
Compute deformation gradient $\mathbf{F} = \mathbf{D}_s\mathbf{D}_m^{-1}$
Compute Green strain $\boldsymbol{\varepsilon}$ from the matrix $\frac{1}{2}(\mathbf{F}^T\mathbf{F} - \mathbf{1}_3)$
Compute strain Jacobian $\mathbf{J}$
Compute local system matrix $\mathbf{A} = h^2\mathbf{J}\mathbf{M}\mathbf{J}^T + \mathbf{C}^{-1}$
Solve $\mathbf{A}\boldsymbol{\lambda} + \boldsymbol{\varepsilon} = 0$
Output: internal forces $\mathbf{f} = \mathbf{J}^T\boldsymbol{\lambda} = (\mathbf{f}_0, \mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3)$
```

6.11 Unilateral constraints

We can consider projection as a QP with both equality and inequality constraints:

$$\begin{aligned} \text{minimize } & \tfrac{1}{2h^2}\delta\mathbf{x}^T\mathbf{M}\delta\mathbf{x} \\ \text{subject to } & \boldsymbol{\Psi}(\mathbf{x}_{k+1}) = 0, \\ & \boldsymbol{\Phi}(\mathbf{x}_{k+1}) \geq 0, \end{aligned} \tag{6.14}$$

where the constraint function $\mathbf{c}$ was split into a bilateral part $\boldsymbol{\Psi}$ and a unilateral one $\boldsymbol{\Phi}$. The most relevant example of a unilateral constraint is contact.

Chapter 7

Nonsmooth dynamics

This chapter is about contact and friction essentially. The name nonsmooth dynamics is given by the fact that both impacts and friction cause discontinuities in the velocity. And this is the context where we are making our own contribution, namely that position based methods can also treat contact and friction in the framework of nonsmooth dynamics.

7.1 Mathematical prelude

$$\mathbf{M}\frac{d\mathbf{v}}{dt} = \mathbf{f}_{ext} + \mathbf{J}^T\boldsymbol{\lambda}, \tag{7.1}$$

$$0 \leq \boldsymbol{\Psi}(\mathbf{q}) \perp \boldsymbol{\lambda} \geq 0. \tag{7.2}$$

This set of equations is called a *differential complementarity problem* (DCP) [90] as it combines a differential equation with complementarity.

The *normal cone* to a given set S at a point $\mathbf{x} \in S$ is the set: $\mathcal{N}_S(\mathbf{x}) = \{\mathbf{p} : \forall \mathbf{y} \in S, (\mathbf{y} - \mathbf{x})^T\mathbf{p} \leq 0\}$. The dual of a cone K is the convex cone [24]: $K^* = \{\mathbf{y} : \forall \mathbf{x} \in K, \mathbf{x}^T\mathbf{y} \geq 0\}$. The negative normal cone is called the *polar cone* or the tangent cone: $K^\circ = -K^*$ with $K = \mathcal{N}_S$.

7.2 Continuous setting

In order to paint a clear picture of contact dynamics we illustrate in Figure 7.1 a particle contact point with a surface. At this point one can identify a normal to the surface, $\mathbf{n}^i$, and any two tangent vectors, $\mathbf{s}^i$ and $\mathbf{t}^i$, so that together they form an orthonormal frame. The most general formulation of constrained dynamics with contact and friction is given in continuous form by a DVI [92]:

$$\mathbf{M}\frac{d\mathbf{v}}{dt} = \sum_{i\in\mathcal{G}_A}(\gamma_n^i\mathbf{D}_n^i + \gamma_s^i\mathbf{D}_s^i + \gamma_t^i\mathbf{D}_t^i) + \sum_{i\in\mathcal{G}_B}(\gamma_B^i\nabla\Psi^i) + \mathbf{f}_{tot}^l, \tag{7.3}$$

$$\frac{d\mathbf{q}}{dt} = \mathbf{v}, \tag{7.4}$$

$$\Psi^i(\mathbf{q}) = 0, i \in \mathcal{G}_B, \tag{7.5}$$

$$0 \leq \Phi^i(\mathbf{q}) \perp \gamma_n^i \geq 0, i \in \mathcal{G}_A, \tag{7.6}$$

$$(\gamma_s^i, \gamma_t^i) = \underset{\sqrt{(\gamma_s^i)^2+(\gamma_t^i)^2}\leq\mu^i\gamma_n^i}{\arg\min} (\mathbf{v})^T(\gamma_s^i\mathbf{D}_s^i + \gamma_t^i\mathbf{D}_t^i). \tag{7.7}$$

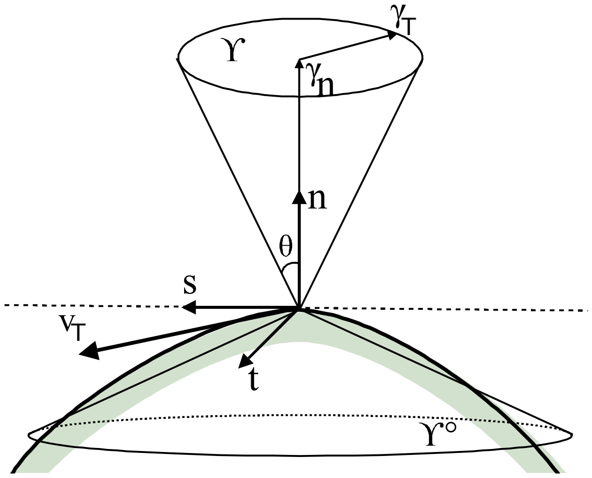


Figure 7.1: Particle contact point with friction cone Υ given by $\theta = \arctan \mu$ and its polar cone Υ° depicted below.

We give here directly a discretized form of the DVI:

$$\begin{aligned}\mathbf{M}(\mathbf{v}^{l+1} - \mathbf{v}^l) &= h \sum_{i \in \mathcal{G}_A} (\gamma_n^i \mathbf{D}_n^i + \gamma_s^i \mathbf{D}_s^i + \gamma_t^i \mathbf{D}_t^i) \\ &\quad + h \sum_{i \in \mathcal{G}_B} (\gamma_B^i \nabla \Psi^i) + h \mathbf{f}_{tot}^l, \end{aligned} \tag{7.8}$$

$$\mathbf{q}^{l+1} = \Lambda(\mathbf{q}^l, \mathbf{v}^{l+1}, h), \tag{7.9}$$

$$\Psi^i(\mathbf{q}^{l+1}) = 0, i \in \mathcal{G}_B, \tag{7.10}$$

$$0 \leq \Phi^i(\mathbf{q}^{l+1}) \perp \gamma_n^i \geq 0, i \in \mathcal{G}_A, \tag{7.11}$$

$$(\gamma_s^i, \gamma_t^i) = \operatorname*{arg\,min}_{\sqrt{(\gamma_s^i)^2 + (\gamma_t^i)^2} \leq \mu^i \gamma_n^i} (\mathbf{v}^{l+1})^T (\gamma_s^i \mathbf{D}_s^i + \gamma_t^i \mathbf{D}_t^i)., \tag{7.12}$$

where Λ is an implicit Euler integration operator for the positions. The novelty in our approach is that we are using a full implicit Euler integrator instead of semi-implicit/symplectic Euler [90, 6] and we keep the non-penetration condition at position level as a nonlinear unilateral constraint.

7.3 Polyhedral friction cone

We can reduce the DVI continuous formulation to a series of solvable mixed LCPs by choosing a set of tangent vectors $\{\mathbf{d}_1 \dots \mathbf{d}_p\}$:

$$\begin{aligned}\mathbf{M}\mathbf{v} - \mathbf{M}\mathbf{v}^l - h\mathbf{f}_{tot}^l - h \sum_{i \in \mathcal{G}_B} (\nabla \Psi_k^i \gamma_B^i) - \\ h \sum_{i \in \mathcal{G}_A} (\gamma_n^i \mathbf{D}_{n,k}^i + \sum_{j=1}^{p} \beta_j^i \mathbf{D}_{j,k}^i) = 0,\end{aligned} \tag{7.13}$$

$$i \in \mathcal{G}_A, 0 \leq \Phi^i(\mathbf{q}_k) + h(\mathbf{D}_{n,k}^i)^T \mathbf{v} \perp \gamma_n^i \geq 0, \tag{7.14}$$

$$0 \leq \lambda^i + (\mathbf{D}_{j,k}^i)^T \mathbf{v} \perp \beta_j^i \geq 0, \tag{7.15}$$

$$0 \leq \mu^i \gamma_n^i - \sum_{j=1}^{p} \beta_j^i \perp \lambda^i \geq 0, \tag{7.16}$$

$$i \in \mathcal{G}_B, \Psi^i(\mathbf{q}_k) + h(\nabla \Psi_k^i)^T \mathbf{v} = 0, \tag{7.17}$$

where $\mathbf{D}_j$ is the generalized coordinates equivalent of $\mathbf{d}_j$, β_j are the friction Lagrange multipliers and λ is a Lagrange multiplier approximating the tangential slip velocity.

7.4 Smooth friction cone

In [4] the DVI is linearized and convexified so that it can be expressed in the end as a quadratic minimization problem with conic constraints. We take this formulation and extend it to the fully implicit and nonlinear case:

$$\begin{aligned} \text{minimize } W(\mathbf{v}) &= \tfrac{1}{2}\mathbf{v}^T\mathbf{M}\mathbf{v} - \hat{\mathbf{f}}^T\mathbf{v} \\ \text{subject to } \Phi^i(\mathbf{q}^{l+1}) &- h\mu^i\|\mathbf{v}_T^i\| \geq 0, i \in \mathcal{G}_A, \\ \Psi^i(\mathbf{q}^{l+1}) &= 0, i \in \mathcal{G}_B, \end{aligned} \tag{7.18}$$

where $\hat{\mathbf{f}} = \mathbf{M}\mathbf{v}^l + h\mathbf{f}_{tot}^l$ and $\|\mathbf{v}_T^i\| = \sqrt{((\mathbf{D}_s^i)^T\mathbf{v})^2 + ((\mathbf{D}_t^i)^T\mathbf{v})^2}$ is the magnitude of the tangential relative velocity at the contact point. Our approach for solving this problem is to derive a new fixed point iteration that is equivalent to a CCP at every kth iteration:

$$\Upsilon_k^\circ \ni -(h\mathbf{D}_k^T\mathbf{v} + \mathbf{b}_k) \perp \boldsymbol{\gamma} \in \Upsilon_k, \tag{7.19}$$

where $\boldsymbol{\gamma} = (\boldsymbol{\gamma}_A, \boldsymbol{\gamma}_B)$ is the Lagrange multipliers vector and $\mathbf{b}_k = (\mathbf{b}_{k,A}, \mathbf{b}_{k,B})$ - the first component corresponding to contacts $\mathbf{b}_{k,A} = (\boldsymbol{\Phi}(\mathbf{q}_k) - h\mathbf{D}_{n,k}^T\mathbf{v}_k\,, 0\,, 0)$ and the second to bilateral constraints $\mathbf{b}_{k,B} = \boldsymbol{\Psi}(\mathbf{q}_k) - h\nabla\boldsymbol{\Psi}_k^T\mathbf{v}_k$. Υ is the direct sum of all friction and bilateral cones, Υ° is the corresponding polar cone and $\mathbf{D}$ is the concatenation of all constraint directions, i.e. $\mathbf{D}_A^i = [\mathbf{D}_n^i|\mathbf{D}_s^i|\mathbf{D}_t^i]$ and $\mathbf{D}_B^i \equiv \nabla\Psi^i$.

7.5 Position projection

It can be shown that the quadratic objective in (7.18) can be reformulated in terms of displacements:

$$\begin{aligned} \text{minimize } W(\delta\mathbf{q}) &= \tfrac{1}{2h^2}\delta\mathbf{q}^T\bar{\mathbf{M}}\delta\mathbf{q}, \\ \text{subject to } &-\mathbf{u} \in \Upsilon^\circ, \end{aligned} \tag{7.20}$$

where $\bar{\mathbf{M}} = \mathbf{L}^T\mathbf{M}\mathbf{L}$ and $\mathbf{u}(\delta\mathbf{q}) = (\mathbf{u}_A, \mathbf{u}_B)$ with $\mathbf{u}_A = h\mathbf{D}_k^T\mathbf{v} + \mathbf{b}_k = (\Phi(\mathbf{q}^{l+1}), h\mathbf{D}_s^T\mathbf{v}^{l+1}, h\mathbf{D}_t^T\mathbf{v}^{l+1})$ and $\mathbf{u}_B = \Psi(\mathbf{q}^{l+1})$. In this form we can easily recognize the projection method for solving differential equations on manifolds (see equation (6.3)). The fixed point iteration in (7.19) extends the projection method to unilateral constraints and friction, given that we use the initial guess for velocity $\mathbf{v}_0 = \tilde{\mathbf{v}}$. If we use $\mathbf{v}_0 = 0$ instead and only one fixed point iteration we obtain the velocity time stepping method with linearized constraints [5].

The dual form problem is obtained following the approach in [67]:

$$\begin{aligned} \text{minimize } &\tfrac{1}{2}\boldsymbol{\gamma}^T\mathbf{A}\boldsymbol{\gamma} + \boldsymbol{\gamma}^T\mathbf{r} \\ \text{subject to } &\boldsymbol{\gamma} \in \Upsilon, \end{aligned} \tag{7.21}$$

where $\mathbf{A} = h^2\mathbf{D}^T\bar{\mathbf{M}}^{-1}\mathbf{D}$ with $\mathbf{D}$ evaluated in $\mathbf{q}^{l+1}$ and $\mathbf{r} = \mathbf{u}$. This is none more than the dual of (7.20).

7.6 Projected iterative solvers

We followed the example of projected relaxation methods and devised our own general projected gradient descent template for solving unilateral constraints similar to (6.6) [35]:

$$\boldsymbol{\lambda}_{k+1} = \mathrm{proj}(\boldsymbol{\lambda}_k + \alpha \mathbf{d}_k + \beta(\boldsymbol{\lambda}_k - \boldsymbol{\lambda}_{k-1})), \tag{7.22}$$

where the projection operator can be as simple as a clamp for nonnegative or box constraints [87] or more complex like a cone projection [92].

7.7 Rigid bodies

Contact only has been tackled in the past either by instantaneously considering it as a bilateral constraint or through a crude complementarity approach. Friction on the other hand has had no solid mathematical framework to rely on and we believe that our nonlinear fixed point iteration is the first (using either an LCP or CCP discretization). You can find our pseudo-code for frictional contact between rigid bodies in Algorithm 2. Note that we identify the two bodies by the indices 1 and 2 and a contact pair is fully determined by a world normal $\mathbf{n}$ and the closest points between the two bodies $\mathbf{a}_1$ and $\mathbf{a}_2$ - each expressed in their respective frame.

Algorithm 2 Pseudo-code for computing the normal and friction forces between 2 rigid bodies in contact. Can be used with either a Jacobi or a Gauss-Seidel approach ($\omega \geq 1, \beta = 0$).

```
Input: contact pair $(\mathbf{n}, \mathbf{a}_1, \mathbf{a}_2)$, $\beta$, old force $\boldsymbol{\gamma}$, and increment $\delta\boldsymbol{\gamma}$
$\mathbf{p}_1 = \mathbf{R}_1\mathbf{a}_1, \mathbf{p}_2 = \mathbf{R}_2\mathbf{a}_2$
Compute normal residual $r_n = \mathbf{n} \cdot (\mathbf{x}_1 + \mathbf{p}_1 - \mathbf{x}_2 - \mathbf{p}_2)$ (gap)
Compute normal diagonal term $d_n$ of matrix $\mathbf{A}$
$\gamma_n = \mathrm{clamp}(\gamma_n - \frac{\omega}{h^2 d_n} r_n - \beta\delta\gamma_n, 0, \infty)$, $\gamma_T = 0$
Compute relative velocity $\mathbf{v}_{12} = (\mathbf{v}_1 + \boldsymbol{\omega}_1 \times \mathbf{p}_1) - (\mathbf{v}_2 + \boldsymbol{\omega}_2 \times \mathbf{p}_2)$
Compute tangential relative velocity $\mathbf{v}_T = \mathbf{v}_{12} - (\mathbf{n} \cdot \mathbf{v}_{12})\mathbf{n}$
if $\mathbf{v}_T \neq 0$ then
    Compute tangential residual $r_T = \|\mathbf{v}_T\|$ (slip speed)
    Compute tangential direction $\boldsymbol{\tau} = \mathbf{v}_T / v_T$
    Compute tangential diagonal term $d_T$
    $(\gamma_n, \gamma_T) = \mathrm{project}(\gamma_n, \gamma_T - \frac{\omega}{h^2 d_T} r_T - \beta\delta\gamma_T)$
end if
Output: contact force $\boldsymbol{\gamma} = (\gamma_n, \gamma_T)$, i.e. $\mathbf{f} = \gamma_n\mathbf{n} + \gamma_T\boldsymbol{\tau}$
```

7.8 Friction models

For an overview of friction models see [77] or the discussion in [57]. It is important to note that some of these approximations are using an estimate of the normal impulse component - the case of box or cylinder friction. This estimate can be a constant [27], coming from a previous frame or from a previous solver iteration [76]. The case that running the normal contact solver and friction solver one after the other is actually a staggered approach is made in [17]. The most popular friction model in computer graphics and games is the box LCP or the square pyramid one [33, 96].

Chapter 8

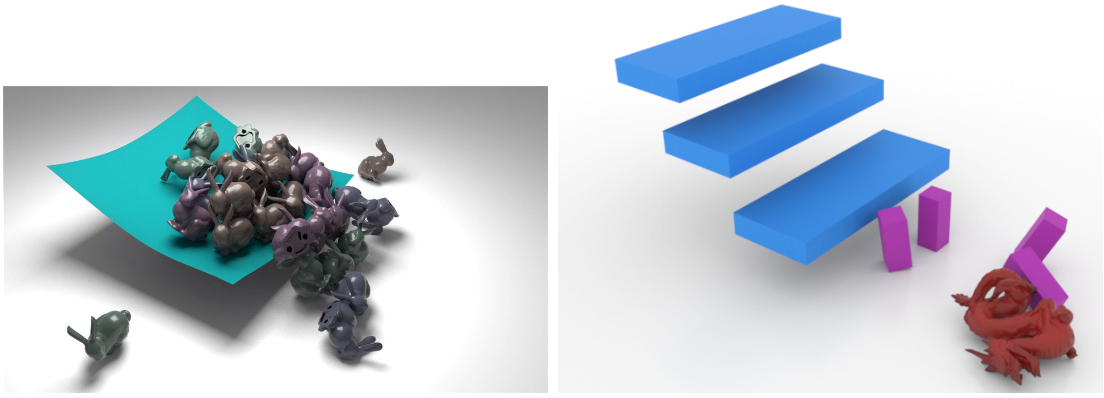
Unified simulation framework

8.1 Nonlinear constrained dynamics

Our unified simulator (Figure 8.1) relies on the theoretical aspects presented in the previous chapters. It is in essence a position based dynamics solver with several extensions. The most important result is that the problem can be formulated as a constrained minimization. This permits us to use a range of algorithms suited for nonlinear optimization.

Algorithm 3 Nonlinear projected gradient descent constraint solver using a Jacobi approach.

```
Unconstrained step to $\tilde{\mathbf{q}}, \tilde{\mathbf{v}}$
$\mathbf{q}_0 = \tilde{\mathbf{q}}, \mathbf{v}_0 = \tilde{\mathbf{v}}$
for k = 0:$k_{max} - 1$ do
    Compute $\mathbf{c}(\mathbf{q}_k)$ and $\mathbf{D}_k$
    Compute residual $\mathbf{r}_k$
    Update Lagrange multipliers (see Section 7.6)
    Apply generalized force $\mathbf{f}_c = \mathbf{D}_k \delta \boldsymbol{\gamma}_{k+1}$ using both position and velocity integrators
end for
```



(a) Rigid bunnies falling on a piece of cloth with two way coupling.

(b) Flexible dragon falling on stairs and hitting rigid boxes.

Figure 8.1: Two way coupling between rigid and flexible bodies.

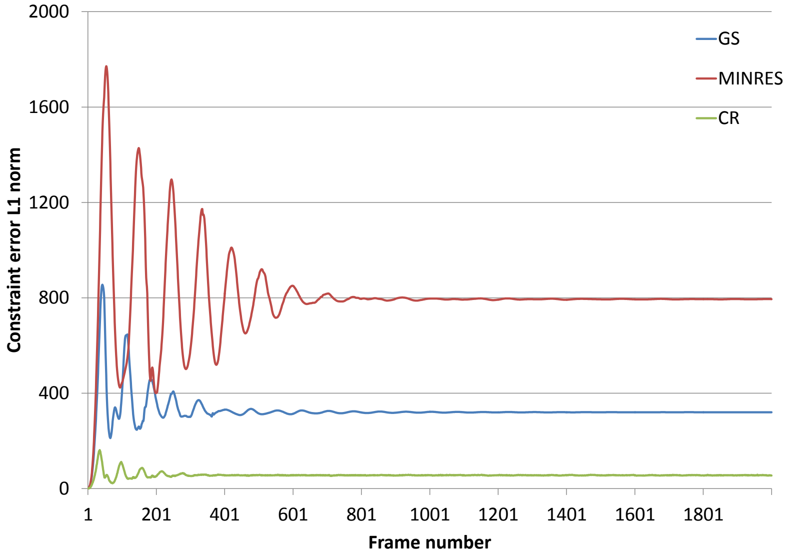


Figure 8.2: Evolution of cloth from initial state to steady state painted as L1 norm of the constraint error for 30 iterations per frame using GS (blue), MINRES (red) and CR (green).

8.2 Implementation and results

All of the algorithms were implemented in C++ in a unified manner such that all constraints were solved at the same time and in the same solver. For this we used a single common list of bodies that could have each a maximum of 6 degrees of freedom. This list was split into groups, each group having a different meaning (e.g. cloth, rigid body system or FEM soft body) and different types of constraint lists. Some constraint types were specific to only one group (e.g. link constraints for cloth), others were common among several groups (e.g. contact constraints) and the rest were specially designed for coupling between groups (e.g. rigid body vs. triangle).

In terms of constraint solving we used mainly two approaches: Gauss-Seidel and accelerated Jacobi. We further optimized the latter using OpenMP parallel for loop directives. Our experiments have shown that the CR method converges better than GS for the same number of iterations, thus making up for the extra computational cost. We illustrated stability and convergence information in Figure 8.2.

We tested the accuracy of our improved Jacobi method on bilateral constraints only. Our test scenario consisted of a 100×100 piece of cloth falling from a horizontal position and hanging from two corners. You can see the evolution in time of the system for three different solvers in Figure 8.3. Clearly our method performs better.

Collision detection was done using both Bullet [30] and our own triangle mesh tests. We implemented our own code because we needed continuous collision detection when performing tests versus cloth or for self-collisions. We accelerated these tests using OpenMP loops and a variant of dynamic AABB trees. Collision detection is run right after the unconstrained position step when non-penetration constraints are sure to be violated. We do cloth-primitive and cloth-mesh intersection tests by testing all pairs of vertices and triangles or other primitives (e.g. sphere) in a similar fashion to [72] and [26]. We also employed continuous collision detection techniques using the ideas presented in [89] based on swept spheres. This was done in order to catch tunneling artifacts although the contacts are still solved at the end of the time step, which is kept fixed.

Many of the simulations for this thesis were done in real-time inside our own OpenGL

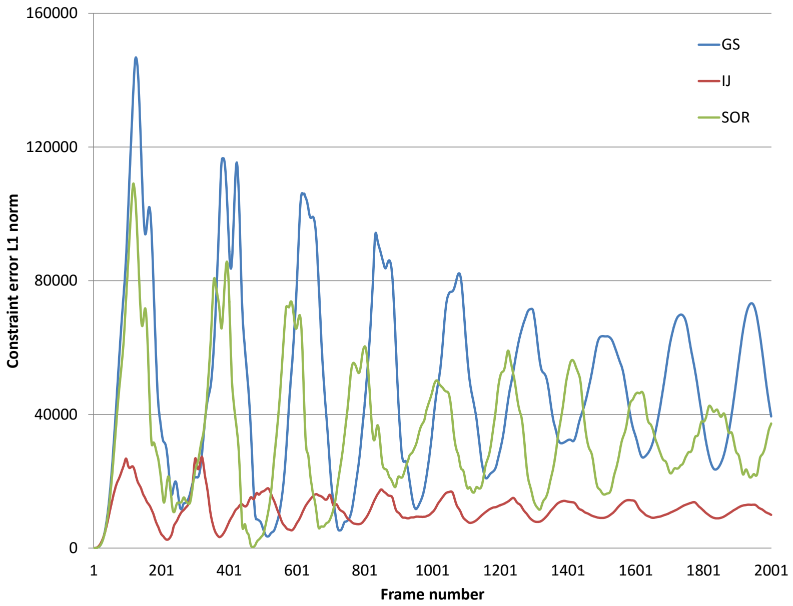
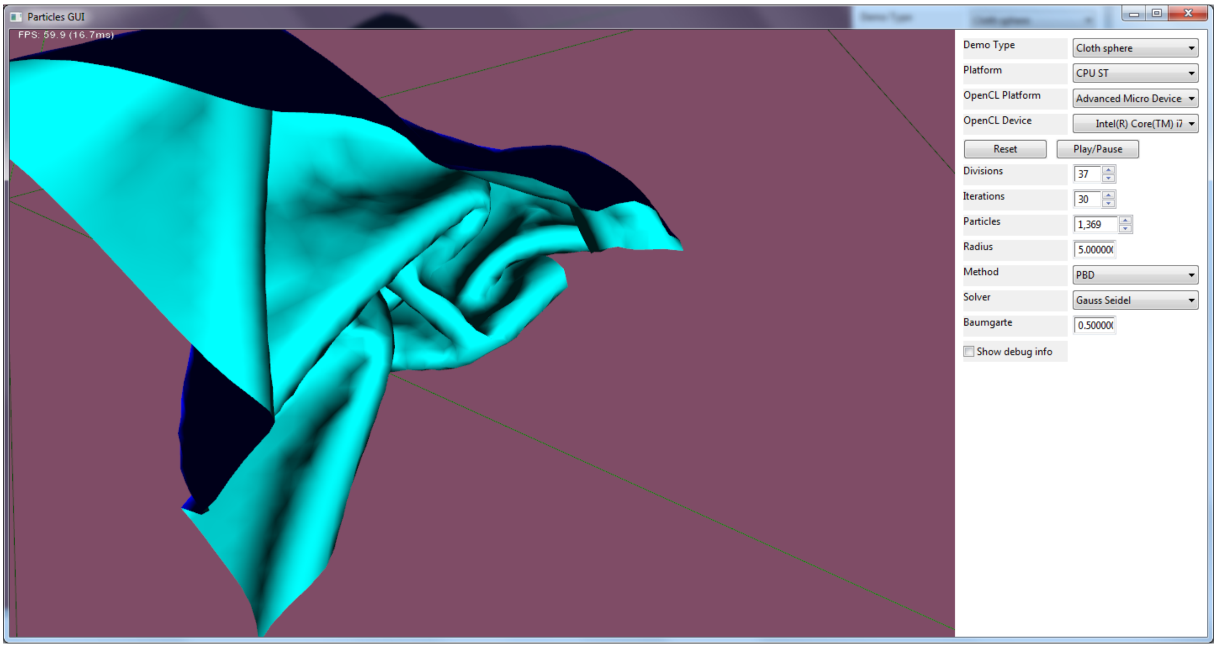


Figure 8.3: Plot of constraint error (L1 norm) for PBD cloth simulation with different solvers (frame number on the horizontal axis): Gauss-Seidel (blue), SOR (green) with $\omega = 1.2$, and improved Jacobi (red) with $\omega = 0.5$, $a = 1$ and $b = 0.6$.

powered Windows application (see Figure 8.4). Others were done in an offline manner and then exported as Alembic geometry caches to Autodesk Maya and rendered using Pixar RenderMan. However, the simulator was written with real-time in mind and a lot of the scenarios ran at interactive rates, some even at 60 Hz. Generally we used a time step $h = 16$ ms, gravity $g = -9.8m/s^2$ and 10 to 50 iterations or more for our iterative solvers. For elastic bodies we used a Young's modulus $E = 0.5$ GPa and a Poisson ratio below 0.2. The masses of cloth and the soft bodies were raised up to around 10 kg in order interact smoothly with rigid bodies of unit mass or less.



Demo Type	Cloth sphere
Platform	CPU ST
OpenCL Platform	Advanced Micro Device
OpenCL Device	Intel(R) Core(TM) i7
Divisions	37
Iterations	30
Particles	1,369
Radius	5.00000
Method	PBD
Solver	Gauss Seidel
Baumgarte	0.50000

□ Show debug info

Figure 8.4: Windows application written in MS Visual C++ using OpenGL.

Chapter 9

Conclusions and future work

9.1 Conclusions

We did not answer all the questions or fulfill all the goals we set in the introduction. But we did clarify a lot of aspects during this doctoral research. For instance, we know now that position based dynamics is truly a physically correct method and there is no cheating involved. It is also deeply connected to velocity time stepping methods used in rigid body simulation: VTS is just a linearization of PBD. We also found out that both PBD and VTS can be expressed as optimization problems and new solving strategies can be used. Even in the case of frictional contact we no longer need to express the problem solely as a LCP but we can turn it into a convex minimization. And this allows us a unifying view of all constraints and a general formulation of PBD as a fully implicit and nonlinear projection scheme.

9.2 Contributions

We will now give a brief list of the original contributions presented in this thesis together with a reference to the section where they are presented and the article where they were published (where applicable):

- a general formulation of position based dynamics as a minimization problem with nonlinear constraints (Section 6.4) equivalent with implicit Euler integration;
- a nonlinear minimum residual and a conjugate residuals (CR) solver for the position based dynamics optimization problem (Section 6.6.2) also published in [36];
- an improved Jacobi scheme (with two variants) based on an extra momentum term that has better convergence than standard Jacobi, is comparable to Gauss-Seidel and can be parallelized (Section 6.6.3);
- an equivalence result between the implicit integration of elastic potentials and position based dynamics with soft constraints (or regularization - Section 6.7) published in [37];
- an accurate finite element method expressed in terms of constraint solving (Section 6.10) based on two aforementioned equivalences: 1. position based dynamics is equivalent to implicit integration and 2. elastic potentials are equivalent to regularized constraints;
- an accurate frictional contact model for position based dynamics stemming from nonsmooth dynamics (Chapter 7);

9.3 Future work

There are many topics that we did not get enough time to study, investigate, understand or implement. We would have liked to have a benchmark system made out of nonsmooth dynamics methods not only from Stewart, Anițescu, Negruț, Tasora but also others, e.g. the sweeping process, the nonsmooth contact dynamics (NSCD) method, variational contact integrators and so on. We also wanted to extend our FEM support to fracture, invertible elements, model reduction and other elasticity models, e.g. co-rotational or neo-Hookean. We need to study more the relationship between the number of iterations and the observed stiffness and behavior. We would also use more off the shelf optimization solvers like Mosek, PATH or MATLAB.

Simulation-wise we want to spend more time on hair and fluids. For the implementation part there is of course a lot of work to do too. Our biggest remaining task is to implement a full unified physics pipeline on the GPU (and replace OpenCL by CUDA). Collision detection also needs a lot of work on many aspects: self-collision, midphase, triangle meshes etc.

Bibliography

[1] I. Alduán and M. A. Otaduy. SPH Granular Flow with Friction and Cohesion. In *ACM SIGGRAPH 2011 Papers*, SIGGRAPH '11, New York, NY, USA, 2011. ACM.

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

[3] H. C. Andersen. Rattle: A "velocity" version of the shake algorithm for molecular dynamics calculations. *Journal of Computational Physics*, 52(1):24–34, 1983.

[4] M. Anitescu. Optimization-based simulation of nonsmooth rigid multibody dynamics. *Mathematical Programming*, 105(1):113–143, 2006.

[5] M. Anitescu and G. D. Hart. A Constraint-Stabilized Time-Stepping Approach for Rigid Multibody Dynamics with Joints, Contact and Friction. *International Journal for Numerical Methods in Engineering*, 60(14):2335–2371, 2004.

[6] M. Anitescu and F. A. Potra. Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlinear Dynamics*, 14(3):231–247, 1997.

[7] V. I. Arnol'd. *Mathematical methods of classical mechanics*, volume 60. Springer Science & Business Media, 2013.

[8] U. M. Ascher, H. Chin, L. R. Petzold, and S. Reich. Stabilization of constrained mechanical systems with daes and invariant manifolds. *Journal of Structural Mechanics*, 23(2):135–157, 1995.

[9] D. Baraff. Fast Contact Force Computation for Nonpenetrating Rigid Bodies. In *ACM SIGGRAPH 1994 Papers*, pages 23–34, 1994.

[10] D. Baraff. Physically based modeling: Rigid body simulation. *SIGGRAPH Course Notes, ACM SIGGRAPH*, 2(1):2–1, 1997.

[11] D. Baraff and A. Witkin. Large Steps in Cloth Simulation. In *ACM SIGGRAPH 1998 Papers*, pages 43–54, 1998.

[12] J. Barbic, Y. Zhao, and H. Xu. Implicit Multibody Penalty-based Distributed Contact. *IEEE Transactions on Visualization and Computer Graphics*, 99(PrePrints):1, 2014.

[13] E. Barth, K. Kuczera, B. Leimkuhler, and R. D. Skeel. Algorithms for Constrained Molecular Dynamics. *J. Comp. Chem*, 16:1192–1209, 1995.

[14] K.-J. Bathe. *Finite element procedures*. Klaus-Jurgen Bathe, 2006.

[15] J. Baumgarte. Stabilization of constraints and integrals of motion in dynamical systems. *Computer methods in applied mechanics and engineering*, 1(1):1–16, 1972.

[16] N. Bell, Y. Yu, and P. J. Mucha. Particle-based Simulation of Granular Materials. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '05, pages 77–86, New York, NY, USA, 2005. ACM.

[17] J. Bender, K. Erleben, J. Trinkle, and E. Coumans. Interactive Simulation of Rigid Body Dynamics in Computer Graphics. *Computer Graphics Forum*, 33(1):246–270, 2014.

[18] J. Bender, D. Koschier, P. Charrier, and D. Weber. Position-Based Simulation of Continuous Materials. *Computers & Graphics*, 44:1–10, 2014.

[19] M. Bergou, M. Wardetzky, D. Harmon, D. Zorin, and E. Grinspun. A quadratic bending model for inextensible surfaces. In *Symposium on Geometry Processing*, pages 227–230, 2006.

[20] F. Bertails, B. Audoly, M.-P. Cani, B. Querleux, F. Leroy, and J.-L. Lévêque. Super-helices for predicting the dynamics of natural hair. *ACM Transactions on Graphics (TOG)*, 25(3):1180–1187, 2006.

[21] F. Bertails-Descoubes, F. Cadoux, G. Daviet, and V. Acary. A nonsmooth Newton solver for capturing exact Coulomb friction in fiber assemblies. *ACM Transactions on Graphics (TOG)*, 30(1):6, 2011.

[22] K. Bodin, C. Lacoursié, and M. Servin. Constraint fluids. *Visualization and Computer Graphics, IEEE Transactions on*, 18(3):516–526, 2012.

[23] S. Bouaziz, S. Martin, T. Liu, L. Kavan, and M. Pauly. Projective Dynamics: Fusing Constraint Projections for Fast Simulation. *ACM Trans. Graph.*, 33(4):154:1–154:11, 2014.

[24] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[25] K. E. Brenan, S. L. Campbell, and L. R. Petzold. *Numerical solution of initial-value problems in differential-algebraic equations*, volume 14. Siam, 1996.

[26] R. Bridson, R. Fedkiw, and J. Anderson. Robust Treatment of Collisions, Contact and Friction for Cloth Animation. In *ACM SIGGRAPH 2002 Papers*, pages 594–603, 2002.

[27] E. Catto. Iterative Dynamics with Temporal Coherence. *Game Developer Conference*, Jan. 2005.

[28] E. Catto. Soft Constraints: Reinventing the Spring. *Game Developer Conference*, 2010.

[29] R. Cottle, J. Pang, and R. Stone. *The Linear Complementarity Problem*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 1992.

[30] E. Coumans. Bullet physics engine. *Open Source Software: http://bulletphysics.org*, 2010.

[31] E. Coumans. Destruction. In *Game Developers Conference Proceedings*. CMP Media, Inc., 2012.

[32] K. Erleben. *Physics-based Animation*. Charles River Media, 2005.

[33] K. Erleben. Velocity-based Shock Propagation for Multibody Dynamics Animation. *ACM Trans. Graph.*, 26, 2007.

[34] J. M. Finn. *Classical mechanics*. Jones & Bartlett Publishers, 2009.

[35] M. Francu and F. Moldoveanu. An improved jacobi solver for particle simulation. In *VRIPHYS 14 - 11th Workshop on Virtual Reality Interactions and Physical Simulations*, pages 125–134, 2014.

[36] M. Francu and F. Moldoveanu. Minimum residual methods for cloth simulation. In *System Theory, Control and Computing (ICSTCC), 2014 18th International Conference*, pages 550–555, Oct 2014.

[37] M. Frâncu and F. Moldoveanu. Cloth simulation using soft constraints. *Journal of WSCG*, 2015.

[38] M. Frâncu and F. Moldoveanu. Virtual Try On Systems for Clothes: Issues and Solutions. *Scientific Bulletin*, 2015.

[39] G. Gilardi and I. Sharf. Literature survey of contact dynamics modelling. *Mechanism and machine theory*, 37(10):1213–1239, 2002.

[40] A. R. Goldenthal. *Implicit Treatment of Constraints for Cloth Simulation*. PhD thesis, 2010.

[41] R. Goldenthal, D. Harmon, R. Fattal, M. Bercovier, and E. Grinspun. Efficient Simulation of Inextensible Cloth. In *ACM SIGGRAPH 2007 Papers*, 2007.

[42] H. Goldstein, C. Poole, and J. Safko. *Classical Mechanics*. Addison Wesley, 2002.

[43] E. Guendelman, R. Bridson, and R. Fedkiw. Nonconvex Rigid Bodies with Stacking. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pages 871–878, New York, NY, USA, 2003. ACM.

[44] S. Hadap. Oriented strands: dynamics of stiff multi-body system. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 91–100. Eurographics Association, 2006.

[45] E. Hairer, C. Lubich, and M. Roche. The numerical solution of differential-algebraic systems by runge-kutta methods. 1989.

[46] E. Hairer, C. Lubich, and G. Wanner. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, volume 31. Springer Science & Business Media, 2006.

[47] T. Harada. Real-Time Rigid Body Simulation on GPUs. In H. Nguyen, editor, *GPU Gems 3*, chapter 29. Addison Wesley Professional, Aug. 2007.

[48] E. J. Haug and J. Yen. Generalized coordinate partitioning methods for numerical integration of differential-algebraic equations of dynamics. In *Real-time integration methods for mechanical system simulation*, pages 97–114. Springer, 1990.

[49] T. Heyn, M. Anitescu, A. Tasora, and D. Negrut. Using Krylov Subspace and Spectral Methods for Solving Complementarity Problems in Many-Body Contact Dynamics Simulation. *International Journal for Numerical Methods in Engineering*, 2012.

[50] M. Hong, M.-H. Choi, S. Jung, S. Welch, and J. Trapp. Effective constrained dynamic simulation using implicit constraint enforcement. In *Robotics and Automation, ICRA 2005*, pages 4520–4525, 2005.

[51] M. Ihmsen, A. Wahl, and M. Teschner. High-resolution Simulation of Granular Material with SPH. In *Workshop on Virtual Reality Interaction and Physical Simulation*, pages 53–60. The Eurographics Association, 2012.

[52] H. M. Jaeger, S. R. Nagel, and R. P. Behringer. Granular solids, liquids, and gases. *Rev. Mod. Phys.*, 68:1259–1273, Oct 1996.

[53] T. Jakobsen. Advanced Character Physics. In *Game Developers Conference Proceedings*, pages 383–401, 2001.

[54] C. Kane, J. E. Marsden, M. Ortiz, and M. West. Variational integrators and the newmark algorithm for conservative and dissipative mechanical systems. *Internat. J. Numer. Methods Engrg*, 49:1295–1325, 1999.

[55] M. Kelager, S. Niebe, and K. Erleben. A triangle bending constraint model for position-based dynamics. *VRIPHYS*, 10:31–37, 2010.

[56] B. Kubiak, N. Pietroni, F. Ganovelli, and M. Fratarcangeli. A robust method for real-time thread simulation. In *Proceedings of the 2007 ACM symposium on Virtual reality software and technology*, pages 85–88. ACM, 2007.

[57] C. Lacoursière. *Ghosts and machines: regularized variational methods for interactive simulations of multibodies with dry frictional contacts*. PhD thesis, Umeå University, Computing Science, 2007.

[58] C. Lanczos. *The Variational Principles of Mechanics*. Dover Publications, 1970.

[59] B. Leimkuhler and S. Reich. *Simulating hamiltonian dynamics*, volume 14. Cambridge University Press, 2004.

[60] T. Liu, A. W. Bargteil, J. F. O'Brien, and L. Kavan. Fast Simulation of Mass-spring Systems. *ACM Trans. Graph.*, 32(6):214:1–214:7, 2013.

[61] M. Macklin and M. Müller. Position Based Fluids. *ACM Trans. Graph.*, 32(4):104:1–104:12, July 2013.

[62] M. Macklin, M. Müller, N. Chentanez, and T.-Y. Kim. Unified Particle Physics for Real-Time Applications. *ACM Transactions on Graphics (TOG)*, 33(4):104, 2014.

[63] L. Mariti, N. Belfiore, E. Pennestrì, and P. Valentini. Comparison of solution strategies for multibody dynamics equations. *International Journal for Numerical Methods in Engineering*, 88(7):637–656, 2011.

[64] F. L. Markley. Attitude error representations for kalman filtering. *Journal of guidance, control, and dynamics*, 26(2):311–317, 2003.

[65] S. Martin, B. Thomaszewski, E. Grinspun, and M. Gross. Example-based elastic materials. *ACM Transactions on Graphics (TOG)*, 30(4):72, 2011.

[66] H. Mazhar, M. Frâncu, and D. Negrut. Simulating large scale coupled granular material simulations using position based dynamics. In *The 4th Joint International Conference on Multibody System Dynamics*, 2016.

[67] H. Mazhar, T. Heyn, D. Negrut, and A. Tasora. Using Nesterov's Method to Accelerate Multibody Dynamics with Friction and Contact. *ACM Transactions on Graphics (TOG)*, 34(3):32, 2015.

[68] B. V. Mirtich. *Impulse-based dynamic simulation of rigid body systems*. PhD thesis, University of California at Berkeley, 1996.

[69] A. Moravánszky. NovodeX Demo Exercise. 2005.

[70] J. J. Moreau. Unilateral contact and dry friction in finite freedom dynamics. In *Nonsmooth mechanics and Applications*, pages 1–82. Springer, 1988.

[71] M. Müller, N. Chentanez, T.-Y. Kim, and M. Macklin. Strain based dynamics. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '14, pages 149–157, Aire-la-Ville, Switzerland, Switzerland, 2014. Eurographics Association.

[72] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff. Position Based Dynamics. *J. Vis. Comun. Image Represent.*, 18(2):109–118, 2007.

[73] M. Müller, J. Stam, D. James, and N. Thürey. Real Time Physics: Class Notes. In *ACM SIGGRAPH 2008 Classes*, pages 88:1–88:90, 2008.

[74] R. Narain, A. Golas, and M. C. Lin. Free-flowing Granular Materials with Two-way Solid Coupling. In *ACM SIGGRAPH Asia 2010 Papers*, SIGGRAPH ASIA '10, pages 173:1–173:10, New York, NY, USA, 2010. ACM.

[75] D. K. Pai. Strands: Interactive simulation of thin solids using cosserat models. In *Computer Graphics Forum*, volume 21, pages 347–352. Wiley Online Library, 2002.

[76] M. Poulsen, S. Niebe, and K. Erleben. Heuristic Convergence Rate Improvements of the Projected Gauss–Seidel Method for Frictional Contact Problems. pages 135–142. Václav Skala - UNION Agency, 2010.

[77] T. Preclik. Iterative Rigid Multibody Dynamics, 2008.

[78] W. H. Press, S. Teukolsky, W. Vetterling, and B. Flannery. Numerical recipes in c. *Cambridge University Press*, 1:3, 2007.

[79] X. Provot. Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior. In *In Graphics Interface*, pages 147–154, 1996.

[80] M. Renouf and P. Alart. Conjugate Gradient Type Algorithms for Frictional Multi-contact Problems: Applications to Granular Materials. *Computer Methods in Applied Mechanics and Engineering*, 194(18):2019–2041, 2005.

[81] J.-P. Ryckaert, G. Ciccotti, and H. J. C. Berendsen. Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. *J. Comput. Phys*, pages 327–341, 1977.

[82] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 2nd edition, 2003.

[83] M. Servin, C. Lacoursière, and N. Melin. Interactive Simulation of Elastic Deformable Materials. In *SIGRAD 2006 Conference Proceedings*, pages 22–32, 2006.

[84] J. R. Shewchuk. An Introduction to the Conjugate Gradient Method Without the Agonizing Pain. Technical report, Pittsburgh, PA, USA, 1994.

[85] E. Sifakis and J. Barbic. FEM Simulation of 3D Deformable Solids: A Practitioner's Guide to Theory, Discretization and Model Reduction. In *ACM SIGGRAPH 2012 Courses*, SIGGRAPH '12, pages 20:1–20:50, New York, NY, USA, 2012. ACM.

[86] M. Silcowitz-Hansen, S. Niebe, and K. Erleben. A nonsmooth nonlinear conjugate gradient method for interactive contact force problems. *The Visual Computer*, 26(6-8):893–901, 2010.

[87] M. Silcowitz-Hansen, S. Niebe, and K. Erleben. Projected Gauss-Seidel Subspace Minimization Method for Interactive Rigid Body Dynamics - Improving Animation Quality using a Projected Gauss-Seidel Subspace Minimization Method. In P. Richard, J. Braz, and A. Hilton, editors, *GRAPP*, pages 38–45. INSTICC Press, 2010.

[88] R. Smith. Open dynamics engine v0.5 user guide. 2006.

[89] J. Stam. Nucleus: Towards a unified dynamics solver for computer graphics. In *Computer-Aided Design and Computer Graphics*, pages 1–11, 2009.

[90] D. Stewart and J. C. Trinkle. An Implicit Time-Stepping Scheme for Rigid Body Dynamics with Coulomb Friction. *International Journal for Numerical Methods in Engineering*, 39:2673–2691, 1996.

[91] M. Tanaka, M. Sakai, Ishikawajima-Harima, and S. Koshizuka. Rigid Body Simulation Using a Particle Method. In *ACM SIGGRAPH 2006 Research Posters*, SIGGRAPH '06, New York, NY, USA, 2006. ACM.

[92] A. Tasora and M. Anitescu. A matrix-free cone complementarity approach for solving large-scale, nonsmooth, rigid body dynamics. *Computer Methods in Applied Mechanics and Engineering*, 200(5):439–453, 2011.

[93] B. Thomaszewski, S. Pabst, and W. Straßer. Continuum-based strain limiting. *Comput. Graph. Forum*, 28(2):569–576, 2009.

[94] B. Thomaszewski, M. Wacker, and W. Straßer. Advanced topics in virtual garment simulation – part 1, 2007.

[95] R. Tonge. Solving Rigid Body Contacts. *Game Developer Conference*, 2012.

[96] R. Tonge, F. Benevolenski, and A. Voroshilov. Mass Splitting for Jitter-free Parallel Rigid Body Simulation. *ACM Trans. Graph.*, 31(4):105:1–105:8, July 2012.

[97] P. Volino, N. Magnenat-Thalmann, and F. Faure. A simple approach to nonlinear tensile stiffness for accurate cloth simulation. *ACM Trans. Graph.*, 28(4):105:1–105:16, 2009.

[98] A. Witkin. Physically based modeling: Principles and practice - constrained dynamics. *Computer graphics*, pages 11–21, 1997.

[99] P. Wriggers and T. A. Laursen. *Computational contact mechanics*, volume 30167. Springer, 2006.

[100] Y. Zhu and R. Bridson. Animating Sand As a Fluid. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, pages 965–972, New York, NY, USA, 2005. ACM.