MULTIBODY DYNAMICS 2009, ECCOMAS Thematic Conference
K. Arczewski, J. Frączek, M. Wojtyra (eds.)
Warsaw, Poland, 29 June–2 July 2009

ITERATIVE RIGID MULTIBODY DYNAMICS

Tobias M. Preclik*, Klaus Iglberger* and Ulrich Rüde*

*Department of Computer Science, Chair for System Simulation
University of Erlangen-Nuremberg, Cauerstr. 6, 91058 Erlangen, Germany
e-mails: tobias.preclik@informatik.uni-erlangen.de,
klaus.iglberger@informatik.uni-erlangen.de,
ulrich.ruede@informatik.uni-erlangen.de
web page: http://www10.informatik.uni-erlangen.de/~toprec/

Keywords: iterative solvers, rigid multibody dynamics, computational methods, contact and impact

Abstract. *Large-scale simulations of rigid multibody systems are a challenge and require in particular robust and efficient methods to solve differential complementarity problems arising from non-smooth contact problems. In this paper established computational methods are described, analyzed and compared within a common physics framework in terms of convergence, active set stabilization and velocity error evolution. Solvers include various matrix splitting methods, a modified Conjugate Gradient method and generalized Newton methods. Additionally the application of an algebraic multigrid method is discussed. All solvers are based on a flexible formulation of the contact problem, which includes numerous modeling options, ranging in complexity from frictionless to isotropic Coulomb friction models.*

1 INTRODUCTION

The simulation of multibody systems is a very active research branch of theoretical, computational and applied mechanics. It attracts researchers from many different communities, having the most diverse applications in mind: Walking machines and robot-arm grippers, virtual reality environments, ground, water and air vehicle dynamics (with connections to multiphysics problems), physics-based animations for use in computer-generated imagery, biomechanics and more. Among the most published topics in the last years were the modeling and solution of contact and impact problems arising in the dynamics simulation of multibody systems [14].

This paper presents a formulation of the contact problem in rigid multibody systems as an impulse-velocity based complementarity problem or convex quadratic optimization problem building on the publications of Stewart and Trinkle [15], Anitescu and Potra [2], Sauer and Schömer [13] and Anitescu [1]. The derivation goes into the details of the friction modeling and includes options ranging from simple box friction to isotropic Coulomb friction. Subsequently, established solution approaches are presented and analyzed, including matrix splitting methods [8] [5], a Conjugate Gradient variant [12] and a generalized Newton method [9]. Additionally, the application of an algebraic multigrid method is discussed, in an attempt to improve the convergence of the matrix splitting methods. The solvers are all iterative in nature in order that the methods are computationally tractable in large-scale simulations. Importance is attached to the range of solvable models and the accompanying necessary solver modifications. All solvers are implemented in the physics framework *pe* [7] in order to allow for a fair comparison. Numerical tests on granular media problems are conducted to compare the solvers in terms of convergence, active set stabilization and velocity error evolution. Characteristic strengths and weaknesses are identified. Since all solvers are capable of solving a common set of models, also model dependencies are taken into account.

In Sec. 2 the modeling of the contact problem as a complementarity problem and a convex quadratic optimization problem is presented, including modeling options to ease the problem formulation. Sec. 3 presents the solution approaches: matrix splitting, Conjugate Gradient, Newton and multigrid methods. The range of solvable models and necessary solver extensions are discussed. In Sec. 4 the numerical experiments are carried out and interpreted. Sec. 5 concludes the paper.

2 MODELING

2.1 CONTACT PROBLEM MODELING

A rigid multibody system consisting of n rigid bodies is uniquely defined by its generalized positions $q(t)$ and its generalized velocities $v(t)$, where $q(t)$ is the aggregate of the locations and orientations of each body. The location of a body is given by a position vector in $\mathbb{R}^3$ specifying the location of the body's centroid in the inertial frame. The orientation of each body can also be expressed as a vector in $\mathbb{R}^3$ and specifies the orientation of the frame of reference attached firmly to each body. The differential equations governing the time evolution of a multibody system are provided by Newton's second law of motion, which states for a constant mass that the force is proportional to the acceleration by the mass. For generalized forces $f(t, q, v)$ and a generalized mass matrix $M(q)$ this reads:

$$f(t, q, v) = M(q)\dot{v}, \quad \dot{q} = v \tag{1}$$

$f(t, q, v)$ is the vector of applied forces, consisting of forces and torques acting at the centroid

of each body. The mass matrix $M(q)$ consistently is a diagonal matrix of body masses and body moment of inertia tensors on the diagonal, relating the linear and angular forces to the linear and angular accelerations.

The solution of systems of ordinary differential equations is well-known, however in multi-body dynamics the relative motion of bodies is additionally constrained. Consider e.g. two bodies a and b, which are joined by a bilateral constraint at position $\vec{p}$. In this case the relative motion of the bodies is constrained, so that no gap forms at the joint. Let us denote the function describing the distance of the two points involved (one on each surface) in this joint j as $\Phi_j(q)$ then the constraint is the following:

$$\Phi_j(q(t)) = \vec{0}, \quad \vec{x}_j \in \mathbb{R}^3 \tag{2}$$

Another type of constraint is the unilateral constraint, where two interacting bodies are (nearly) in contact at a point $\vec{p}$. In each potential contact j two points are involved, one on each of the surfaces of the contacting bodies. Without loss of generality we attach an orthonormal local coordinate frame to the first of the bodies (body a), where one of the axis points in surface normal direction $\vec{n}$ and the other two vectors $\vec{t}$ and $\vec{o}$ orthogonally span the tangential plane. Then the gap function $\Phi(q(t))_j$ measures the distance of the point associated with body b involved in the contact in terms of the contact frame. If the normal component is zero, then the bodies are actually in contact. Naturally we want to avoid penetration of the bodies and require that $\Phi_n(q(t)) \geq \vec{0}$ at all times. The penetration is prevented by a contact force This force must be repulsive (not adhesive) in nature and thus is constrained to be non-negative too. In order to produce stable simulations, we pose the non-penetration constraint at the end of the current time step Δt and at the same time move from contact forces to force integrals (impulses) $\vec{x}$ acting on the bodies over the next time step. The constraint is completed by adding a complementarity condition[1] between the motion and the impulses, which ensures that impulses act if and only if the bodies are actually in contact.

$$\Phi_n(q(t + \Delta t)) \geq \vec{0} \perp \vec{x}_n \geq \vec{0} \tag{3}$$

Furthermore, friction should be present. As customary for dry frictional contacts, the empirical Coulomb law is adopted, which loosely states that the friction must do its utmost to prevent relative tangential motion. It is bounded by the coefficient of friction μ times the normal component acting at the contact. Assuming $\Phi(q(t + \Delta t))$ can be expressed by an affine function of the contact impulses in the form $\mathbf{A}\vec{x} + \vec{b}$, then the friction constraints can be easily formulated in the form of the maximum dissipation problem for a contact j:

$$\vec{x}^*_{j_{to}} = \underset{\left\|\vec{x}_{j_{to}}\right\|_2 \leq \mu_j x_{j_n}}{\operatorname{argmin}} \frac{1}{2}\vec{x}^T_{j_{to}}(\mathbf{A}\vec{x})_{j_{to}} + \vec{x}^T_{j_{to}}\vec{b}_{j_{to}} \tag{4}$$

Note that the subscript j selects all components of the j-th contact and among those subscripts t and o select both tangential components. Equivalently this principle can be expressed as direct constraints, which are the Karush-Kuhn-Tucker conditions[2] of the optimization problem above.

[1] Componentwise either of the operands must be zero.

[2] First-order necessary optimality conditions.

$$\vec{x}_{j_{to}}^T(\mathbf{A}\vec{x}+\vec{b})_{j_{to}} = -\left\|(\mathbf{A}\vec{x}+\vec{b})_{j_{to}}\right\|_2 \cdot \|\vec{x}_{j_{to}}\|_2 \tag{5a}$$

$$\|\vec{x}_{j_{to}}\|_2 \leq \mu_j x_{j_n} \tag{5b}$$

$$\left\|(\mathbf{A}\vec{x}+\vec{b})_{j_{to}}\right\|_2 \cdot (\|\vec{x}_{j_{to}}\|_2 - \mu_j x_{j_n}) = 0 \tag{5c}$$

In other words, the friction force must exactly oppose the slip at the contact or prevent the tangential motion.

To state $\Phi(q(t+\Delta t))$ in terms of an affine function of $\vec{x}$, we set up the Taylor expansion of Φ at the point $q(t)$.

$$\begin{aligned}&\Phi(q(t+\Delta t)) = \Phi(q(t)) + J_\Phi(q(t))\Delta q + O(\Delta q^2)\\ &\text{where } \Delta q = q(t+\Delta t) - q(t)\end{aligned} \tag{6}$$

The Jacobian can be broken apart into a mapping $\mathbf{P}^T$ from generalized (body) velocities to relative contact velocities in the global reference frame and a rotation $\mathbf{Q}^T$ converting from the global reference frame to the orthonormal contact reference frame of each of the m contacts.

$$\begin{aligned} J_\Phi(q(t)) &= \mathbf{Q}^T\mathbf{P}^T = \mathbf{J}^T \\ \mathbf{Q} = \operatorname{diag}\left(\mathbf{Q_j}\right)_{j=1\ldots m} \quad \mathbf{Q_j} &= \begin{pmatrix}\vec{n}_j & \vec{t}_j & \vec{o}_j\end{pmatrix} \in \mathbb{R}^{3\times 3} \\ \mathbf{P} = \left(\mathbf{P_{ij}}\right)_{i=1\ldots n, j=1\ldots m} \quad \mathbf{P_{ij}} &= \begin{cases} -\begin{pmatrix}\mathbf{E_3}\\ \mathbf{r_{ij}^\times}\end{pmatrix} & \text{if } i = b_1(j) \\ \begin{pmatrix}\mathbf{E_3}\\ \mathbf{r_{ij}^\times}\end{pmatrix} & \text{if } i = b_2(j) \\ \mathbf{0} & \text{else}\end{cases} \in \mathbb{R}^{6\times 3}\end{aligned} \tag{7}$$

The matrix $\mathbf{E}_N$ denotes the unit matrix of size N and the vector $\vec{r}_{ij}$ points from the centroid of body i to contact j expressed in terms of the global reference frame. The $\times$ superscript turns a vector into a linear operator which calculates the cross-product of the vector and its right-hand side operand. The resulting matrix is skew-symmetric ($\mathbf{r}^{\times\,T} = -\mathbf{r}^\times$). The functions $b_{1,2}$ map contacts respectively to the index of the first and second body involved in the contact.

The exact derivation of Δq in Eq. (6) is irrelevant for the considerations to follow. However, it is important that Δq takes the form $\Delta t(\vec{b} + \mathbf{M}^{-1}\mathbf{J}\vec{x})$ and thus Eq. (6) is indeed an affine function of $\vec{x}$.

$$\frac{\Phi(q(t+\Delta t))}{\Delta t} = \frac{\Phi(q(t))}{\Delta t} + \mathbf{J}^T\mathbf{M}^{-1}\mathbf{J}\vec{x} + \mathbf{J}^T\vec{b} \tag{8}$$

This completes the problem formulation consisting of unilateral constraints (Eq. (3) and either Eq. (4) or Eq. (5)) and bilateral constraints (Eq. (2)). The term $\frac{\Phi(q(t))}{\Delta t}$ may be neglected if the gaps are small. On the other hand, it can be used for constraint stabilization.

2.2 FRICTION MODELING OPTIONS

Unifying the different constraints in a single mathematical problem formulation is surprisingly difficult. Bilateral constraints are simple linear equations. However, the non-penetration

Term	$\mathcal{C}$	Constraint Set $\mathcal{F}_j^{\mathcal{C}}$	Cross Sections
pyramidal cone		$\{\vec{x}_j \mid x_{j_n} \geq 0,\ \vec{x}_{j_{to}} \in \mathcal{S}^{\square}(\mu_j x_{j_n})\}$	$\mathcal{S}^{\square}(r) = [-r, r]^2$
box		$[0, \infty) \times \mathcal{S}^{\square}(\mu_j \tilde{x}_{j_n})$	
polyhedral cone		$\{\vec{x}_j \mid x_{j_n} \geq 0,\ \vec{x}_{j_{to}} \in \mathcal{S}^{\circ}(\mu_j x_{j_n})\}$	$\mathcal{S}^{\circ}(r) = \Big\{\vec{x}_{j_{to}} = \sum_{k=1}^{\frac{\eta}{2}} \lambda_{j_{t_k}} \vec{t}_k \ \Big\vert$
prism		$[0, \infty) \times \mathcal{S}^{\circ}(\mu_j \tilde{x}_{j_n})$	$\lvert\lambda_{j_{t_k}}\rvert \leq r,\ \lvert\sum_{k=1}^{\frac{\eta}{2}} \lambda_{j_{t_k}}\rvert \leq r\Big\}$
circular cone		$\{\vec{x}_j \mid x_{j_n} \geq 0,\ \vec{x}_{j_{to}} \in \mathcal{S}^{\circ}(\mu_j x_{j_n})\}$	$\mathcal{S}^{\circ}(r) = \left\{\vec{x}_{j_{to}} \ \middle\vert\ \lVert\vec{x}_{j_{to}}\rVert_2 \leq r\right\}$
cylinder		$[0, \infty) \times \mathcal{S}^{\circ}(\mu_j \tilde{x}_{j_n})$	

Table 1: Choices for the set of admissible contact impulses.

constraint Eq. (3) is a linear complementarity constraint and the maximum dissipation principle Eq. (4) is a convex optimization problem with a quadratic objective function coupled to the normal component. The optimization problem may be replaced by its optimality conditions Eq. (5), however those are no linear complementarity constraints due to the circular shape of the constraint set. By replacing this set by a box constraint $[-\mu_j x_{j_n}, \mu_j x_{j_n}]^2$, the optimality conditions become a box-constrained complementarity problem with variable bounds, which are due to the coupling to the normal component again.

$$\left(\mathbf{J}^T \mathbf{M}^{-1} \mathbf{J} \vec{x} + \mathbf{J}^T \vec{b}\right)_{j_{to}} \gtreqless \vec{0} \perp -\begin{pmatrix} \mu_j x_{j_n} \\ \mu_j x_{j_n} \end{pmatrix} \leq \begin{pmatrix} x_{j_t} \\ x_{j_o} \end{pmatrix} \leq \begin{pmatrix} \mu_j x_{j_n} \\ \mu_j x_{j_n} \end{pmatrix} \tag{9}$$

This effectively changes the friction cone (the set of feasible frictional contact impulses) from a circular to a pyramidal cone shape. Unfortunately this introduces anisotropic friction. By adding more facets to the pyramidal cone, the anisotropy can be reduced at the cost of more complicated subproblems. The order of the subproblems increase, because the tangential plane is spanned uniformly in all directions by an even number of vectors and each has an associated friction component. An additional constraint is introduced to keep the total friction impulse within the polyhedral cone, which is not guaranteed anymore due to the oversampling of the tangential plane. For more details refer e.g. to [15].

Though the introduction of a polyhedral friction cone removes nonlinearities from the bounds on the friction components, the coupling of the friction components to the normal components remains a problem. By fixing the bounds on the friction components, this coupling can be removed from the system. This requires a reasonable estimate $\tilde{x}_{j_n}$ of the contacts' normal components. The friction cone now takes the form of a box, a prism or a cylinder. Tab. (1) summarizes the different types of friction cones, introduces notations and defines the corresponding sets.

The decoupling enables the reformulation of the contact problem as a convex quadratic optimization problem, whose optimality conditions correspond to the complementarity conditions

Eq. (3), Eq. (5) or Eq. (9) (with constant bounds) respectively.

$$\text{minimize } \frac{1}{2}\vec{x}^T\mathbf{J}^T\mathbf{M}^{-1}\mathbf{J}\vec{x} + \vec{x}^T\mathbf{J}^T\vec{b} \quad \text{s.t. } \vec{x} \in \mathcal{F}^{\mathcal{C}} = \prod_{j=1\ldots m} \mathcal{F}_j^{\mathcal{C}} \tag{10}$$

The relationship between the convex optimization problem Eq. (10) and the complementarity conditions breaks down for *cone shaped* sets $\mathcal{F}^{\mathcal{C}}$. In these cases, the underlying model changes. The effects can be understood by observing that the optimality conditions of Eq. (10) correspond to the following Variational Inequality VI.

$$\text{find } \vec{x} \in \mathcal{F}^{\mathcal{C}} \text{ s.t. } \left(\mathbf{J}^T\mathbf{M}^{-1}\mathbf{J}\vec{x} + \mathbf{J}^T\vec{b}\right)^T (\vec{y} - \vec{x}) \geq 0 \quad \forall \vec{y} \in \mathcal{F}^{\mathcal{C}} \tag{11}$$

Eq. (11) requires that the relative contact velocity corresponding to a solution $\vec{x}$ is part of the negative normal cone at the point $\vec{x}$. In the case of dynamic friction (slip occurs at the contact), the friction force is bounded by the boundary of the constraint set. The normal cone at such a point equates to the ray in the outward normal direction of the constraint set surface. This implicates that contact velocities will not only have a tangential slip in case of a dynamic contact, but also a separating motion in the contact normal direction. The effect is negligible as long as the coefficient of friction μ is low or the relative contact velocity small. According to [16] the resulting complementarity conditions for the normal component, which replaces Eq. (3), is the following:

$$\left(\mathbf{J}^T\mathbf{M}^{-1}\mathbf{J}\vec{x} + \mathbf{J}^T\vec{b}\right)_{j_n} - \mu_j \left\| \left(\mathbf{J}^T\mathbf{M}^{-1}\mathbf{J}\vec{x} + \mathbf{J}^T\vec{b}\right)_{j_{to}} \right\|_2 \geq 0 \perp x_{j_n} \geq 0 \tag{12}$$

3 SOLUTION APPROACHES

3.1 MATRIX SPLITTING METHODS

The most widespread solution approach is to apply variants of the Gauss-Seidel iterative method. When solving linear systems of equations (LSEs), single linear equations are solved one after another. Here instead of the equations, complementarity subproblems must be solved. Solving complementarity problems of the order 1 is just a matter of projecting the solution of the corresponding linear equation on the constraint set. For instance a box-constrained linear complementarity problem is of the form $ax + b \gtreqless 0 \perp \underline{x} \leq x \leq \overline{x}$ and the solution is $x = \max\left(\underline{x}, \min\left(\overline{x}, -a^{-1}b\right)\right)$ (assuming $a > 0$). This is already enough to solve the frictionless contact problem and the box frictional problem componentwise and the solver is a standard Projected Gauss-Seidel (PGS).

For complementarity problems of order two and above, the projection method breaks down unless the subsystem matrix is a multiple of the identity matrix.

$$\begin{aligned} \vec{x}_s &= -\mathbf{A}^{-1}\vec{b} \\ \vec{x}_s &= \vec{x} + \underbrace{\Delta\vec{x}}_{\text{removed by projection}} \\ \mathbf{A}\vec{x} + \vec{b} &= \mathbf{A}(\vec{x}_s - \Delta\vec{x}) + \vec{b} = -\mathbf{A}\Delta\vec{x} \end{aligned} \tag{13}$$

Note that $-\mathbf{A}\Delta\vec{x}$ does not fulfill the complementarity condition, because $\mathbf{A}$ perturbs the direction of $\Delta\vec{x}$. Consequently, other subsystem solvers are needed for block Gauss-Seidel solvers, where subproblems in the order of the block size are solved. For box-constrained linear

complementarity problems (LCPs) of small size, total enumeration schemes are viable. Such a scheme enumerates all components' possibilities of boundedness (bounded above, bounded below, unbounded) combinations. Each possibility results in a reduced LSE, which is solved. If the solution meets all inequality constraints, the LSE solution is a solution of the complementarity problem. This approach also works for complementarity problems with variable bounds like Eq. (9). For increasing problem sizes N this approach quickly becomes infeasible because for each complementarity problem an exponential number (3^N) of LSEs (of order $\leq N$) has to be solved.

Other direct solvers for LCPs exist, which can replace a total enumeration subsystem solver. Well-known possibilities are the Lemke algorithm, Dantzig's algorithm or Zoutendijk's procedure. However, the direct solvers must be known to solve the subproblem class. Eventually, the subproblems have to be reformulated into standard LCPs because the direct solver might not be able to cope with box-constraints. Furthermore, direct LCP solvers usually also have exponential worst case complexity (though on average they perform significantly better). For details refer to [4].

Last but not least the subsystem solvers can be iterative solvers in itself. For example the well-known Non-smooth Contact Dynamics (NSCD) method (cf. [8]), employs a generalized Newton method to successively solve one-contact subproblems in a Gauss-Seidel fashion. The one-contact subproblems are reformulations of Eq. (4) and Eq. (3) into a root-finding problem of a non-smooth function. In the following, an iterative subsystem solver for solving the maximum dissipation problem Eq. (4) is presented, which treats only friction components as a block and thus is easier to solve than the one-contact subproblem.

$$\vec{x}^* = \operatorname*{argmin}_{\|\vec{x}\|_2 \leq r} \frac{1}{2}\vec{x}^T \mathbf{A}\vec{x} + \vec{x}^T\vec{b} \tag{14}$$

Initially the solution of the LSE $\mathbf{A}\vec{x}_s + \vec{b} = \vec{0}$ is calculated, e.g. by the means of a precomputed Cholesky factorization of $\mathbf{A}$ or by using Cramer's rule. If the solution is inside the disc with radius r, then a minimum is already found and the contact is static. Otherwise we know that the minimum is attained only if $\vec{x}$ resides on the boundary of the constraint set ($\|\vec{x}\|_2 = r$). Consequently, a change to polar coordinates is advantageous. The resulting optimization problem is smooth, unconstrained and one-dimensional.

$$\alpha^* = \operatorname*{argmin}_{\alpha} \frac{1}{2}r^2 \begin{pmatrix}\cos\alpha \\ \sin\alpha\end{pmatrix}^T \mathbf{A} \begin{pmatrix}\cos\alpha \\ \sin\alpha\end{pmatrix} + r \begin{pmatrix}\cos\alpha \\ \sin\alpha\end{pmatrix}^T \vec{b} = \operatorname*{argmin}_{\alpha} f(\alpha) \tag{15}$$

The classical Newton method perfectly solves this problem. As initial solution we choose the projection of the unconstrained solution $\vec{x}_s$ onto the disc with radius r. As noted above, the initial solution does not yield a minimum in general due to the perturbation by $\mathbf{A}$. However, for homogeneous spherical bodies in contact, where the diagonal block $\mathbf{A}$ is a multiple of the identity matrix, it does and no Newton iteration is required.

$$\begin{aligned} &k = 0 \\ &\alpha_0 = \operatorname{atan2}\left(\frac{r}{\|\vec{x}_s\|_2}\vec{x}_s\right) \\ &\frac{\mathrm{d}f}{\mathrm{d}\alpha}(\alpha^k) + \left(\alpha^{k+1} - \alpha^k\right)\frac{\mathrm{d}^2 f}{\mathrm{d}\alpha^2}(\alpha^k) = 0 \quad \forall k \geq 0 \end{aligned} \tag{16}$$

Eq. (16) solves the contact problem with isotropic Coulomb friction under one reservation: Eq. (14) requires a disc with constant radius r, however the bounds on the friction components depend on the normal component unless a cylindrical friction cone is chosen. This can be dealt with in several ways. One option is to generally only solve cylindrical, box or prism constraints and use an outer (fixed point) iteration to adapt the bounds of the friction components. The advantage is that the inner loop solves a quadratic instead of a nonlinear problem, which subserves the number of iterations. After every outer iteration a feasible solution exists, however the friction bounds can be far from the optimum. Instead of wrapping an outer iteration around the Gauss-Seidel, the bound adaptation can be performed continuously within the iteration. The described treatment of variable bounds of course also applies to the pyramidal and polyhedral cone friction case.

3.2 CONJUGATE PROJECTED GRADIENT METHOD

The Conjugate Gradient (CG) method has good reputation for solving symmetric and positive semi-definite (PSD) linear systems of equations. Essentially it works with an unconstrained quadratic optimization problem, where the minimum corresponds to the solution of the LSE. Compared to the splitting methods it has quadratic instead of linear convergence. To apply a CG method to a contact problem, the handling of constraints needs to be incorporated into the algorithm to solve the convex quadratic optimization problem (cf. Eq. (10)). The CG variant here is based on the results of [12]. Note that generally any solver for convex quadratic optimization problems is applicable. For example a GS approach exists to solve the convex quadratic optimization problem with a cone shaped constraint set as suggested in [16].

Omitting the conjugation, the CG method reduces to the Gradient Descent method. To integrate constraints into the Gradient Descent method, the descent direction $\vec{p}$ is projected to the tangent cone[3] T of the constraint set $\mathcal{F}^{\mathcal{C}}$ at the current iterate $\vec{x}^k$. Then an unconstrained linesearch along the projected direction is performed. Eventually, the iterate leaves the constraint set if new constraints are to become active. A subsequent projection step to the constraint set ensures, that the iterate is admissible again. Alg. (1) lists one iteration of the Projected Gradient Descent method.

1 $\vec{r}^k = -\left(\mathbf{A}\vec{x}^k + \vec{b}\right)$

2 $\vec{p}^k = \mathrm{proj}_{T_{\mathcal{F}^{\mathcal{C}}}(\vec{x}^k)}\left(\vec{r}^k\right)$

3 $\alpha^k = \dfrac{\vec{r}^{k\,T}\vec{p}^k}{\vec{p}^{k\,T}\mathbf{A}\vec{p}^k}$

4 $\vec{x}^{k+\frac{1}{2}} = \vec{x}^k + \alpha^k\vec{p}^k$

5 $\vec{x}^{k+1} = \mathrm{proj}_{\mathcal{F}^{\mathcal{C}}}\left(\vec{x}^{k+\frac{1}{2}}\right)$

Algorithm 1: A single iteration of the Projected Gradient Descent method.

To improve the convergence of the Projected Gradient Descent method, a conjugation of the previous descent direction and the projected gradient is performed. The resulting direction should never point to the outside of the constraint set, therefore the previous descent direction as well as the gradient is projected to the tangent cone before the conjugation takes place. As long as the active set does not change, the conjugation is fully effective. Alg. (2) lists one iteration of the CPG method.

[3]The polar cone of the normal cone.

1 $\vec{r}^k = -\left(\mathbf{A}\vec{x}^k + \vec{b}\right)$

2 $\hat{\vec{r}}^k = \mathrm{proj}_{T_{\mathcal{FC}}(\vec{x}^k)}\left(\vec{r}^k\right)$

3 $\hat{\vec{p}}^{k-1} = \mathrm{proj}_{T_{\mathcal{FC}}(\vec{x}^k)}\left(\vec{p}^{k-1}\right)$

4 $\beta^k = -\dfrac{\hat{\vec{r}}^{k\,T}\mathbf{A}\vec{p}^{k-1}}{\vec{p}^{k-1\,T}\mathbf{A}\vec{p}^{k-1}}$

5 $\vec{p}^k = \hat{\vec{r}}^k + \beta^k\hat{\vec{p}}^{k-1}$

6 $\alpha^k = \dfrac{\vec{r}^{k\,T}\vec{p}^k}{\vec{p}^{k\,T}\mathbf{A}\vec{p}^k}$

7 $\vec{x}^{k+\frac{1}{2}} = \vec{x}^k + \alpha^k\vec{p}^k$

8 $\vec{x}^{k+1} = \mathrm{proj}_{\mathcal{FC}}\left(\vec{x}^{k+\frac{1}{2}}\right)$

Algorithm 2: A single iteration of the Conjugate Projected Gradient method.

The basic algorithm can be used to solve most of the convex quadratic optimization problems considered here. Though the underlying models are simplified, the CPG can also be adapted to solve the nonlinear model with pyramidal friction: The basic algorithm works with a box friction model, where the size of the box is adapted in each iteration. Thus the tangent cone projections project to a box constraint set. Furthermore, the iterate correction ensures that components in the active set, stay at the bounds e.g. by setting them to the current value of the non-constant bound. So high friction coefficients will cause large adaptations, which in turn will destroy the conjugation of successive descent directions. The extension of the algorithm to non-polyhedral constraint sets is an open problem. In either case the CPG iteration described above needs two matrix-vector multiplications per iteration compared to just one of a standard CG where the residual is updated in terms of $\mathbf{A}\vec{p}^{k-1}$.

3.3 GENERALIZED NEWTON METHODS

Unlike CG methods, Newton's method is well suited for solving general nonlinear systems and still features quadratic convergence. Of course the benefits come at a price: The method is not guaranteed to converge globally. Nevertheless, it is the most flexible approach available. We consider a linear complementarity problem with variable bounds of the following form:

$$\mathbf{A}\vec{x} + \vec{b} \gtreqless \vec{0} \perp \underline{x}(\vec{x}) \leq \vec{x} \leq \overline{x}(\vec{x}) \tag{17}$$

The complementarity problem can be reformulated as a system of nonlinear equations as explained in [9] and [11]:

$$F(\vec{x}) = \max\left(\vec{x} - \overline{x}(\vec{x}), \min\left(\vec{x} - \underline{x}(\vec{x}), \mathbf{A}\vec{x} + \vec{b}\right)\right) = \vec{0} \tag{18}$$

The function F is non-smooth due to the $\min$ and $\max$ functions, which produce kinks. Consequently, the function is not differentiable in the classical sense. Pang extended the Newton method to B-differentiable functions in [10], where the B-differential is defined to be:

$$J_F^B(\vec{x}^*) := \left\{ \lim_{k\to\infty} J_F(\vec{x}^k) \;\middle|\; \lim_{k\to\infty} \vec{x}^k = \vec{x}^* \right\} \tag{19}$$

The method proposed by Pang linearizes the non-smooth system in each iteration and reduces it to a mixed linear complementarity problem[4] of smaller size. The resulting method is a *non-smooth* Newton method. To get around the expensive complementarity subproblem in each

[4]LCP with linear equations added.

iteration, *semi-smooth* methods pick an arbitrary derivative from $J_F^B(\vec{x}^k)$ and use it to define an LSE subproblem in each iteration. The resulting Newton equation is the following:

$$J_F(\vec{x}^k)\Delta\vec{x} = -F(\vec{x}^k) \tag{20}$$

For constant bounds $\underline{\vec{x}}$ and $\overline{\vec{x}}$, the Jacobian is symmetric, PSD and the subproblem is consistent. Thus it can always be solved by a Cholesky decomposition or a CG solver. However, if the bounds are variable, then the subsystem matrix becomes asymmetric and the Newton equation possibly singular. More general and expensive subsystem solvers are necessary like a Gaussian elimination, a GMRES method or other solvers to solve the normal equations. In both cases, solving the Schur complement is much more efficient. Once the subsystem is solved, a search direction $\Delta\vec{x}$ is known. A subsequent line-search along $\Delta\vec{x}$ ensures, that an objective function, like $\frac{1}{2}F(\vec{x})^T F(\vec{x})$ is monotonically reduced. For details refer to [9] and [11].

The Newton method in contrast to the CPG and Gauss-Seidel variants is slow at finding the correct active set of constraints. But as soon as the active set is correct, the solution is at hand within a single iteration. Furthermore, the method described does not ensure that the iterate stays within the bounds. Thus the constraints can be strongly violated until the solution is found.

As already noted, the Newton method can possibly get caught in local minima of the objective function. This is especially an issue for the truly nonlinear problem with variable bounds. A minimum step size of the line-search can help the iterate to escape the local minimum. Unfortunately it is subject to tuning. Instead of coping with asymmetric, singular subproblems and local minima, we can instead again use friction box adaptations to prevent the first two problems and ease the last problem. In each iteration of the Newton method, the bounds are adapted and the subsystem is constructed, pretending that the bounds are constant.

It should be noted that there exist numerous alternatives to reformulate the complementarity problem into a system of nonlinear equations. For instance the well-established PATH solver uses a non-smooth Newton method to find a root of the normal map [6]:

$$F(\min(\overline{\vec{x}}, \max(\underline{\vec{x}}, \vec{y}))) + \vec{y} - \min(\overline{\vec{x}}, \max(\underline{\vec{x}}, \vec{y})) = \vec{0} \tag{21}$$

Then $\vec{x} = \min(\overline{\vec{x}}, \max(\underline{\vec{x}}, \vec{y}))$ provably solves the complementarity problem.

3.4 MULTIGRID METHODS

In Sec. 3.1, several matrix splitting methods were introduced including methods, which can even compute the solution to contact problems with circular friction cones. All methods of Gauss-Seidel type exhibit only linear convergence. A popular approach to improve the their convergence are multigrid methods. In addition to the original system coarser versions of the system are constructed. Gauss-Seidel methods efficiently reduce local errors and therefore high frequency errors in the solution. However, error components, which are of a more global nature (low frequency errors), are not efficiently reduced and thus the Gauss-Seidel iterations only slowly improve the solution after local errors have been eliminated. The idea behind multigrid is that a coarser system representation or *coarse-grid* is used to remove lower frequency error components. In order to solve the coarse system, again the Gauss-Seidel method or in a general context a *smoother* is applied to reduce local errors. As a result, on the coarse-grid lower frequency errors than on the fine-grid are removed. The results are transfered from the coarser to the finer grid (interpolation) and vice versa (restriction). Typically a hierarchy of grids is established and the coarsest grid is usually solved using a direct solver.

Unfortunately, the contact problem does not have an underlying structured grid and it is not obvious how a coarser system looks like. Instead the unknowns (contact impulses) are distributed irregularly in space. Furthermore, the unknowns are subject to bounds and complementarity constraints. Even worse, the bounds depend on the unknowns. Consequently, it is not possible to define a coarser representation of the contact problem in a geometric sense. Instead we will apply algebraic multigrid ideas. For the moment we will consider frictionless and box frictional models only, where the bounds are fixed and do not depend on the unknowns, because in the cone friction case, the problem would become even more non-standard: Tangential components on the coarse-grid require that the corresponding normal component is also added to the coarse-grid variables, due to the coupling of the bounds. Thus, by sticking to the frictionless and box friction models we can ignore the coupling and coarsen the normal and tangential components independently of each other.

As a start we adopted standard algebraic multigrid approaches for LSEs. To identify a set of variables to be present on the coarse grid, the concept of strong dependency and strong influence is exploited: If a variable i strongly influences a variable j then j strongly depends on i and i is a good candidate for accurate interpolation of j. This observation drives the partitioning of the variables into coarse-grid variables C and fine-grid variables F. In order to counterbalance the strongly influencing variables in the coarse-grid set C, it is demanded that the strong dependencies within C are minimized [3].

After constructing the partitioning, the setup of the interpolation operator is performed. For LSEs this setup relies on the assumption that at least on average an algebraically smooth error $\vec{e}$ has small residuals ($\vec{r} = \mathbf{A}\vec{e} \approx \vec{0}$). The i-th row of the interpolation operator is either a weighted average of the neighboring, strongly influencing coarse-grid variables C_i^s if i is a fine-grid variable, or if i is a coarse-grid variable, then the interpolation is simply the identity.

$$\left(\mathbf{P}\vec{e}\right)_i = \begin{cases} e_i & \text{if } i \in C \\ \sum_{j \in C_i^s} \omega_{ij} e_j & \text{if } i \in F \end{cases} \tag{22}$$

It is mandatory to note, that the above construction of the interpolation operator is purely based on techniques for LSEs. To achieve better results, a more sophisticated approach must carry over the underlying principles to complementarity systems.

The setup of the coarse-grid matrix is the product of the restriction operator, the fine-grid matrix and the interpolation operator, where the restriction operator is the transpose of the interpolation operator due to the variational property. In the following, superscript H denote coarse-grid quantities and superscript h fine-grid quantities.

$$\mathbf{A}^H = \mathbf{P}^T \mathbf{A}^h \mathbf{P} \tag{23}$$

The residual complementarity problem is the following:

$$\begin{aligned} &\mathbf{A}^H \vec{e}^H - \vec{r}^H \gtrless \vec{0} \perp \left(\underline{\vec{x}} - \vec{x}\right)_C \leq \vec{x}^H \leq \left(\overline{\vec{x}} - \vec{x}\right)_C \\ &\vec{r}^H = \mathbf{P}^T \vec{r}^h \\ &\vec{r}^h = -\mathbf{A}^h \vec{x}^h - \vec{b}^h \end{aligned} \tag{24}$$

This complementarity problem can again be coarsened in a recursive fashion until the problem size is small enough for a direct solver to be efficient. The overall solution process advances in recursive correction cycles. At first the smoother is applied ν_1 times, then the residual

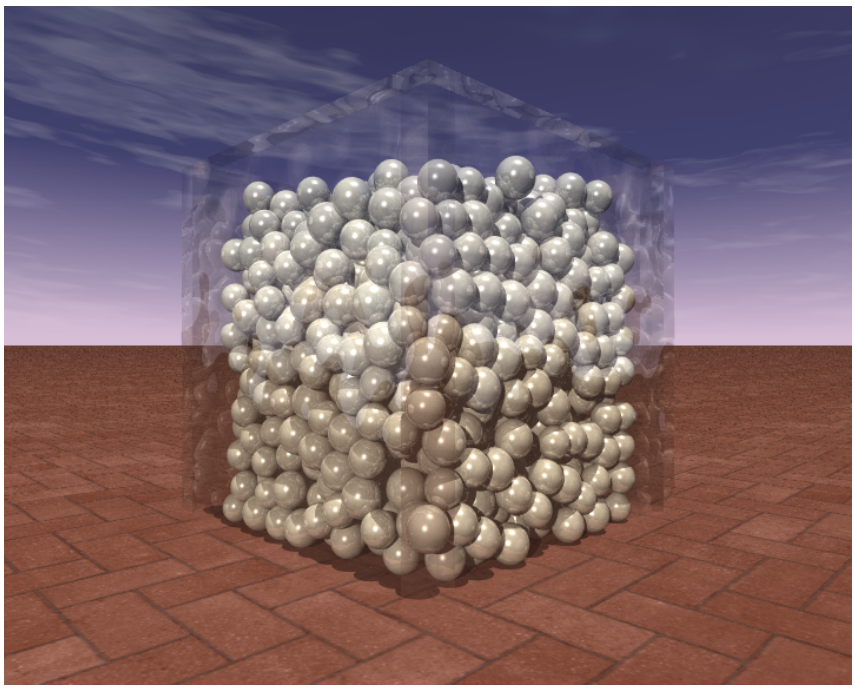


Figure 1: Spherical granular media inside a container.

is computed and restricted to the coarse-grid. The bounds on the coarse-grid are computed and the cycle proceeds in a recursive fashion μ times. On return from the recursion, the coarse-grid error is at hand. It is interpolated and added to the iterate, unless it violates the bounds, in which case the correction is rejected. Alternatively, the corrected variable can be projected to its constraint set. Eventually, the multigrid cycle is ended by ν_2 smoother iterations.

The solver described discloses several problems with the approach taken: Transferring the contact impulse errors is problematic. This stems from the fact that the contact problem is underdetermined, that is possibly an infinite number of solutions exist. A simple example is a frictionless box resting on a plane. At least at each corner a contact point must be generated. The sum of the normal forces must then sum up to counteract gravity on the box. The only other constraint is that the box may not start to rotate and thus the contact forces diagonally across must be of the same magnitude. However, the distribution of the forces to the four contacts still allows for an infinite number of solutions. This effects the transfer of the errors, because the different levels of the multigrid hierarchy may converge to different solutions. Consequently, the interpolated error from the coarse grid is counterproductive. Though the impulses are not uniquely determined, the velocities (body as well as contact velocities) are. This observation suggests to transfer velocity errors instead of impulse errors.

Naturally the interpolation will work best for LCPs, which have many unbounded variables and thus are very similar to LSEs (which have exclusively unbounded variables). A possible resort for problems with many dynamic contacts could be to warm-start the multigrid solver by executing several smoother iterations beforehand. A large fraction of the active set will be identified. The coarsening process can then take advantage of this information. Alternatively, the interpolation operator could be adapted in the course of the solution process.

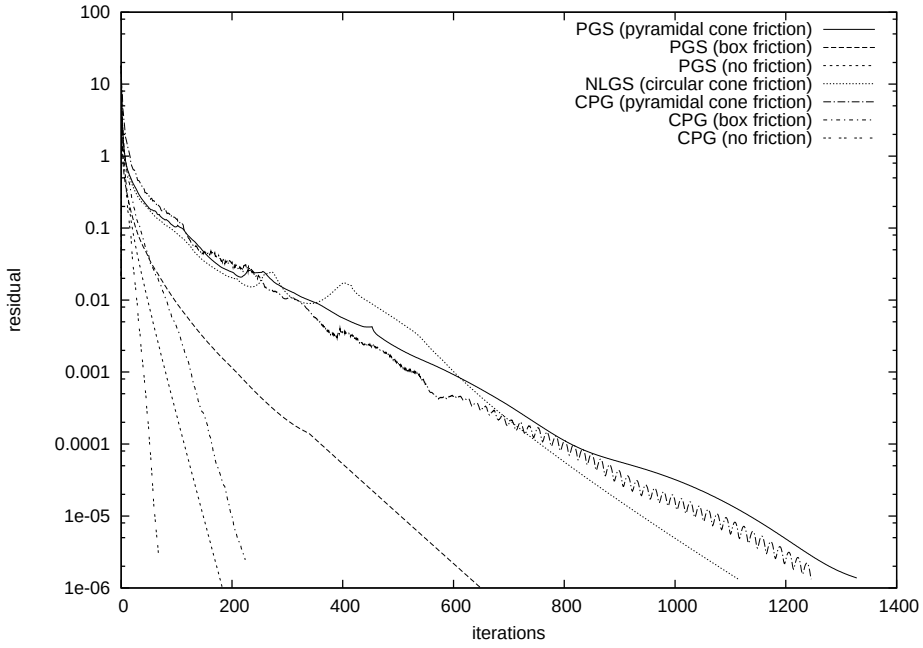


Figure 2: Residual plots of GS and CG variants on different models. Strong friction is present ($\mu = 0.8$).

4 NUMERICAL EXPERIMENTS

All solvers were implemented in the physics framework *pe* [7] in a modular and generic way. This enables the solvers to support at least the frictionless, the box friction and the pyramidal cone friction models. Isotropic Coulomb friction is supported only by the matrix splitting method. Unfortunately the CPG method is not extendible to the circular friction cone in a straightforward manner. Though the Newton method could be extended, preliminary tests indicated that the solver is not competitive and thus was not extended.

The solvers are tested with granular media test cases consisting exclusively of spheres. The spheres are settling in a box-shaped container but are not yet at rest to get a reasonable mixture of dynamic and static contacts. Fig. (1) gives an impression of such a test case.

Fig. (2) plots the residuals in the maximum norm on a granular media test case with 1702 contacts and 5106 variables (in the frictional case) of the PGS and CPG solvers performing on different model problems respectively. The Newton method was excluded from this test because it was clearly inferior. The two graphs on the left correspond to the CPG and PGS performing on a frictionless problem. The next two are again CPG and PGS, but this time performing on a box friction model, where the friction bounds have been approximated beforehand by solving a frictionless problem. The number of iterations increases compared to the frictionless case roughly by a factor of 3 for each solver. Each time the CPG needs only half the number of iterations compared to the PGS as expected due to the quadratic convergence. The last three graphs correspond to the solution processes of cone friction models. The solid line corresponds to the residual evolution of the PGS solver, where bound adaptations are performed to solve the pyramidal cone friction model. For spherical granular media this is identical to a solver, where contactwise subproblems (for example using a total enumeration scheme) are solved instead of componentwise subproblems. The jittering dashed line is the residual evolution of the CPG on

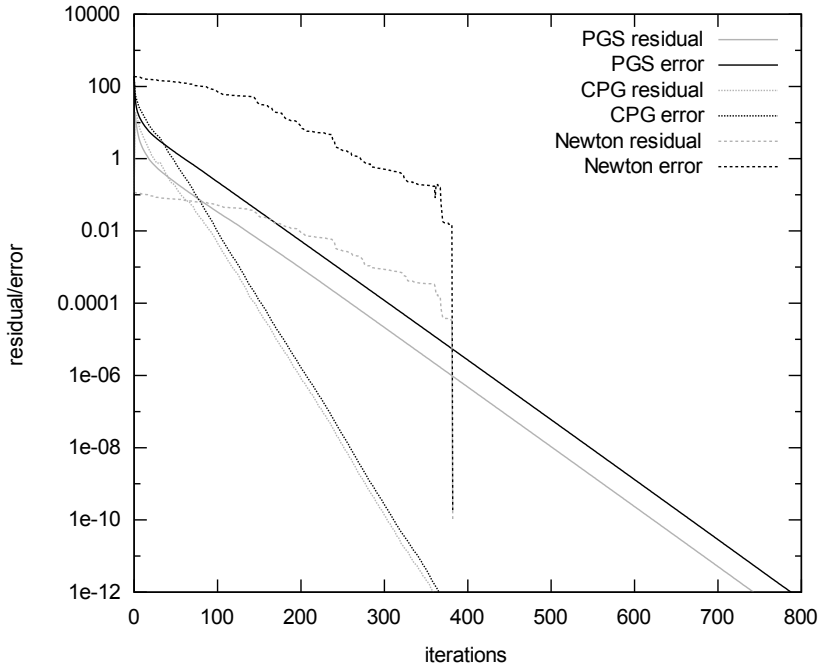


Figure 3: The residual and the corresponding error for each solver performing on a box friction model.

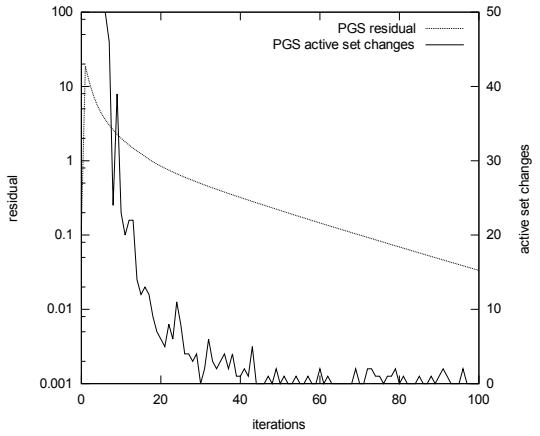
a pyramidal cone friction model. The dotted line corresponds to the nonlinear GS, which solves maximum dissipation problems Eq. (4) on the circular friction cones. Note that no Newton iterations are required by the subsystem solver due to a setup consisting exclusively of spheres.

Several conclusions can be drawn from the plots: Purely quadratic problems like the frictionless and box friction models can be solved much more efficiently and robustly than the nonlinear problems, where the coupling of friction and normal component is considered. The CPG method usually needs only half the number of iterations required by the PGS. However, if friction is strong in the nonlinear problem, the CPG's convergence deteriorates. The nonlinear GS solving the circular cone friction model typically takes slightly less iterations than a PGS for a pyramidal cone friction model. However, the subsystem solver for the friction components is more expensive. Finally it is not surprising, that all the solvers can fail on the nonlinear (pyramidal or circular cone frictional) models, if the friction is strong enough.

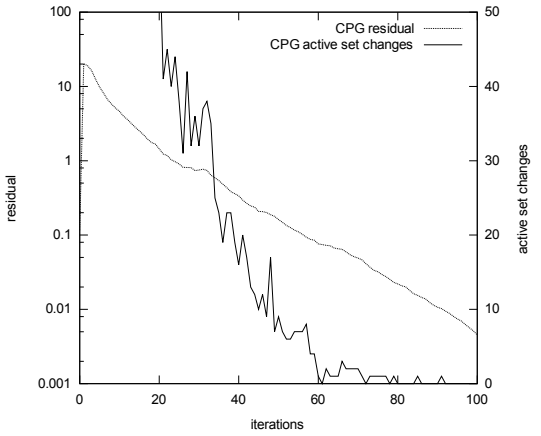
So far only the residual plots have been compared. More interesting is the comparison of the error evolution. Since the contact problem is singular as noted before, there exists no unique solution and computing the error of the contact impulses is not possible. But computing the contacts' *velocity* errors is a valid approach. Fig. (3) plots the residual and the velocity error of the PGS, the CPG and the Newton method on a box friction model.

The PGS and CPG error graphs are closely coupled to their residual graphs. In particular the error in the PGS drops again faster in the beginning than the CPG error just like the residual did. However, the velocity error of the Newton method is only loosely related to the residual. This makes the Newton method even more unattractive because it requires many iterations where the velocity error is barely reduced. Unless we iterate until the sharp drop-off occurs, the Newton method on its own is by far inferior. This property transfers to the cone friction model and to different coefficients of friction.

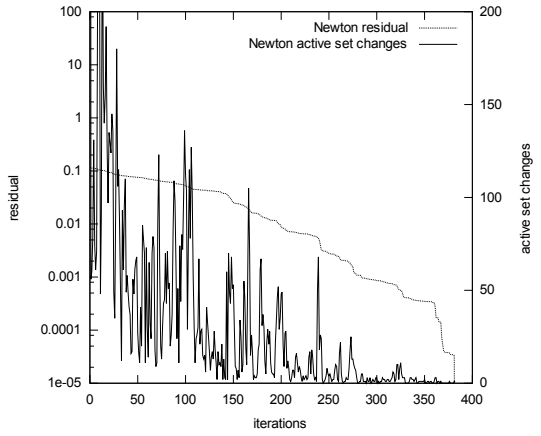
The CPG method relies on a stable active set in order to build up a conjugate subspace. Fig. (4) adds the number of active set changes to detail enlargements of residual graphs. The



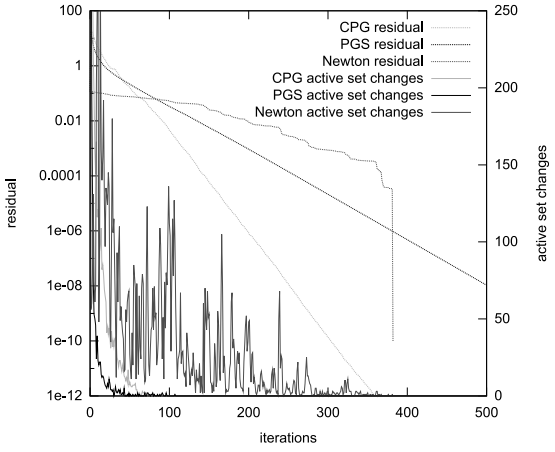
(a) Detail enlargement of the active set evolution of the PGS method.



(b) Detail enlargement of the active set evolution of the CPG method.



(c) Detail enlargement of the active set evolution of the Newton method.



(d) Overview of the active set evolutions.

Figure 4: Active set changes per iteration of different solvers.

solvers differ dramatically in their ability to stabilize the active set. The PGS and the CPG method both exhibit an exponential drop in the number of active set changes, whereas the Newton method's active set changes only decrease linearly on average. Please note the different scaling of the Newton chart.

5 CONCLUSIONS

Numerous solvers were presented and tested in the same framework on the same problems. It turned out that the GS solvers are robust and easy to implement solvers, which can solve the whole range of models. In comparison to the CPG, the convergence is slow. However, the CPG method presented here cannot benefit from the lower number of iterations because the iterations are also roughly twice as expensive. A CG variant, which omits the second matrix-vector product would perform considerably better. If this can be resolved, an investigation of good preconditioners would be worthwhile. Though generalized Newton method's are frequently used to solve complementarity problems, the tests show that the Newton method is usually inferior when solving rigid multibody contact problems. Essentially, this is because the subproblems are expensive to solve in every iteration and the resulting search direction is not targeted enough.

However, if the active set is accurate, the Newton method is good at reducing the remaining error. Therefore a combination with a matrix splitting method could be beneficial, because it could support the Newton method by finding the correct active set.

Though the solvers differ strongly in the convergence rates, it is evident that all solvers can solve the quadratic problems more efficiently. In this context, only frictionless and box friction models were considered as quadratic problems. However, as already presented in Eq. (10), there exists a trade-off between the nonlinear model with pyramidal or circular cone friction and the quadratic model with box or cylinder friction. Solving the convex quadratic optimization problem with the Cartesian product of (pyramidal or circular) friction cones as the constraint set results in problems, which are still quadratic but also have a coupling between the normal and friction components. This comes in exchange for a slightly modified non-penetration constraint, which allows small normal movements in the presence of friction. Solvers should reach convergence rates comparable to the solvers of the box friction model and still have a coupling between the normal and friction components. Including such a solver in the comparison would be worthwhile.

In view of large-scale simulations, better convergence rates would be attractive. However, applying multigrid concepts to improve the convergence of the GS solvers does not work out well. The setup costs are high and the convergence is impeded by the large null space, which is common for the system matrices arising from contact problems. Another necessity for large-scale simulations is the parallelization of the solvers, either for use on standard multi-core processors or on modern graphics hardware [16] or even for high-performance computing on today's supercomputers [7]. Either way, the parallelization will be a part of future work.

REFERENCES

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

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

[3] W.L. Briggs, V.E. Henson, and S.F. McCormick. *A multigrid tutorial (2nd ed.)*. Society for Industrial and Applied Mathematics, 2000.

[4] R.W. Cottle, J.-S. Pang, and R.E. Stone. *The Linear Complementarity Problem*. Academic Press, Inc., 1992.

[5] K. Erleben. *Stable, Robust, and Versatile Multibody Dynamics Animation*. PhD thesis, University of Copenhagen, 2004.

[6] M.C. Ferris and T.S. Munson. Complementarity Problems in GAMS and the PATH Solver. Technical report, University of Wisconsin Computer Sciences Department, 1998.

[7] K. Iglberger and U. Rüde. Massively parallel rigid body dynamics simulations. In *Proceedings of the International Supercomputing Conference 2009 (accepted for publication)*, 2009.

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

[9] R. Ortiz-Rosado. *Newton/Amg Algorithm for Solving Complementarity Problems Arising in Rigid Body Dynamics with Frictional Impacts*. PhD thesis, University of Iowa, 2007.

[10] J.-S. Pang. Newton's Method for B-differentiable equations. *Mathematics of Operations Research*, 15(2):311–341, 1990.

[11] T. Preclik. *Iterative Rigid Multibody Dynamics*. Diploma thesis, Friedrich-Alexander University Erlangen-Nuremberg, 2008.

[12] M. Renouf and P. Alart. Conjugate gradient type algorithms for frictional multi-contact problems: applications to granular materials. *Computer Methods in Applied Mechanics Engineering*, 194:2019–2041, 2005.

[13] J. Sauer and E. Schömer. A constraint-based approach to rigid body dynamics for virtual reality applications. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 153–162, 1998.

[14] W. Schiehlen. Research trends in multibody system dynamics. *Multibody System Dynamics*, 18(1):3–13, 2007.

[15] D.E. Stewart and J.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.

[16] A. Tasora, D. Negrut, and M. Anitescu. Large-scale parallel multibody dynamics with frictional contact on the graphical processing unit. Technical report, Argonne National Laboratory Mathematics and Computer Science Division, 2008.