INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2012; ~~00:1–21~~
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme

Using Krylov Subspace and Spectral Methods for Solving Complementarity Problems in Many-Body Contact Dynamics Simulation

Toby Heyn[1], Mihai Anitescu[2], Alessandro Tasora[3], Dan Negrut[1]*

[1]*Department of Mechanical Engineering, University of Wisconsin - Madison, Madison, WI 53706, USA*
[2]*Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, USA*
[3]*Department of Industrial Engineering, Università degli Studi di Parma, V.Usberti 181/A, 43100 Parma, Italy*

SUMMARY

Many-body dynamics problems are expected to handle millions of unknowns when, for instance, investigating the three-dimensional flow of granular material. Unfortunately, the size of the problems tractable by existing numerical solution techniques is severely limited on convergence grounds. This is typically the case when the equations of motion embed a differential variational inequality (DVI) problem that captures contact and possibly frictional interactions between rigid and/or flexible bodies. As the size of the physical system increases, the speed and/or the quality of the numerical solution decrease. This paper describes three methods - the gradient projected minimum residual (GPMINRES) method, the preconditioned spectral projected gradient with fallback (P-SPG-FB) method, and the Kučera method - that demonstrate better scalability than the projected Jacobi and Gauss-Seidel methods commonly used to solve contact problems that draw on a DVI-based modeling approach. Copyright © 2012 John Wiley & Sons, Ltd.

Received . . .

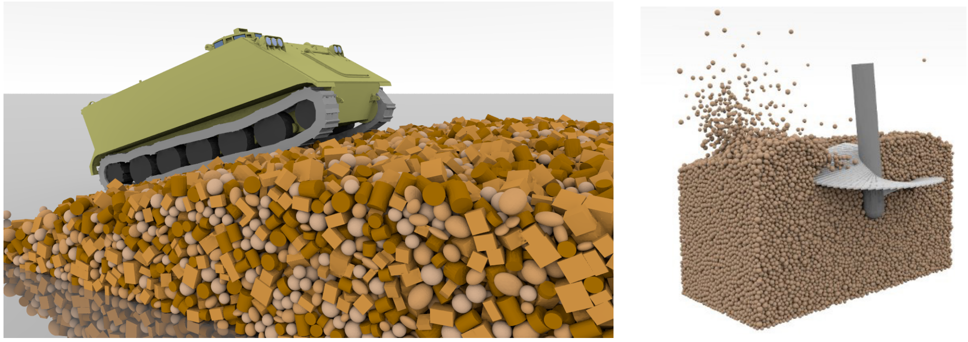
KEY WORDS: Solids; Contact; Differential Equations; Multibody Dynamics

1. INTRODUCTION

The ability to efficiently and accurately simulate the dynamics of rigid multibody systems is relevant in computer-aided engineering design, virtual reality, video games, and computer graphics. Devices composed of rigid bodies interacting through frictional contacts and mechanical joints pose numerical solution challenges because of the discontinuous nature of their motion [1]. Consequently, the simulation of even relatively small systems composed of a few hundred parts and constraints may require significant computational effort. More complex scenarios such as soil and rock dynamics, vehicles operating on pebbles and sand, and flow and packing of granular materials are particularly challenging and prone to long simulation times. A representative simulation problem is shown in Fig. 1(a), which illustrates a light autonomous tracked vehicle that negotiates a pile of rubble in which the material feature length is comparable with the dimensions of the vehicle. While mapping the differential variational inequality (DVI)-based numerical solution onto a parallel architecture for a problem such as in Fig. 1(a) reduces the simulation time, doing so does not address the underlying problem of slow solution convergence [2]. Existing commercial software solutions may also struggle with scalability issues. Results reported in [3] indicate that the most popular rigid body software for

[†]Preprint ANL/MCS-P2099-0612, Mathematics and Computer Science Division, Argonne National Laboratory
*Correspondence to: 2035 Mechanical Engineering Building, 1513 University Avenue, Madison, WI 53706. E-mail: negrut@engr.wisc.edu

engineering simulation, which uses an alternative approach based on the so-called discrete element method [4], runs into significant difficulties even when handling problems involving fewer than a thousand of contact events.



(a) Light autonomous vehicle negotiating a pile of rubble.

(b) Cutaway view: Anchor penetrating granular material [5].

Figure 1. Two examples in which hundreds of thousands of bodies interact mutually through contact and friction. Simulation results for systems with up to 1.1 million bodies are reported in [6]. Experimental validation results are reported in [5].

Unlike the so-called penalty or regularization methods, where the frictional interaction can be represented by a collection of stiff springs combined with damping elements that act at the interface of the two bodies [7], the approach embraced here draws on time-stepping procedures producing weak solutions of the DVI problem that characterizes the time evolution of rigid bodies with impact, contact, friction, and bilateral constraints. Early numerical methods based on DVI formulations can be traced back to the early 1980s and 1990s [8, 9, 10]. Recent approaches based on time-stepping schemes have included both acceleration-force linear complementarity problem (LCP) approaches [11, 12] and velocity-impulse LCP-based time-stepping methods [13, 14, 15]. The LCPs, obtained as a result of the introduction of inequalities accounting for nonpenetration conditions in time-stepping schemes coupled with a polyhedral approximation of the friction cone, must be solved at each time step in order to determine the system state configuration as well as the Lagrange multipliers associated with the reaction forces [9, 13]. If the simulation entails a large number of contacts and rigid bodies, as is the case in Fig. 1(a), the computational burden of classical LCP solvers becomes prohibitive. Indeed, a well-known class of numerical methods for LCPs based on simplex methods, also known as direct or pivoting methods [16], may exhibit exponential worst-case complexity [17]. Moreover, the three-dimensional Coulomb friction case leads to a nonlinear complementarity problem (NCP). The use of a polyhedral approximation to transform the NCP into an LCP introduces unwanted anisotropy and significantly augments the size of the numerical problem [13, 14].

The limitations imposed by the use of classical LCP solvers and the accuracy issues plaguing the polyhedral approximation of the friction cone can be avoided by introducing a relaxation over the complementarity constraints that transforms the original NCP into a cone complementarity problem (CCP) [18]. The CCP is currently solved by using a fixed-point iteration method with projection on a convex set [19]. Since the Jacobi and Gauss-Seidel approaches employed require a large number of iterations when handling large systems of engineering relevance, the CCP solution, which currently accounts for 90% of the total solution time, leads to prohibitively long simulations. The purpose of this paper is to address this issue by comparing the Jacobi method with three new iterative methods in order to identify a scalable method that demonstrates improved CCP convergence. Note that we do not report results for Gauss-Seidel because they are qualitatively similar to those obtained with the Jacobi method [20] and the method is not amenable to parallel computing.

The rest of the paper is organized as follows. The next section provides a brief description of the DVI formulation and how it leads to the CCP of interest. Three new iterative approaches are proposed for solving the CCP of multibody dynamics. Next, we analyze the performance of the proposed iterative methods using three benchmark numerical experiments. The paper concludes with brief comments on the performance of the proposed methods and a discussion of future directions of research.

2. SETTING UP THE CCP

The equations of motion are formulated hereby using a so-called absolute, or Cartesian, representation of the attitude of each rigid body in the system. The state of the system is represented by the generalized positions $\mathbf{q} = \left[\mathbf{r}_1^T, \epsilon_1^T, \ \ldots, \mathbf{r}_{n_b}^T, \epsilon_{n_b}^T\right]^T \in \mathbb{R}^{7n_b}$ and their time derivatives $\dot{\mathbf{q}} = \left[\dot{\mathbf{r}}_1^T, \dot{\epsilon}_1^T, \ldots, \dot{\mathbf{r}}_{n_b}^T, \dot{\epsilon}_{n_b}^T\right]^T \in \mathbb{R}^{7n_b}$, where n_b is the number of bodies, $\mathbf{r}_j$ is the absolute position of the center of mass of body j, and the quaternions (Euler parameters) ϵ_j are used to represent body orientation. The set of quaternions is identified by $\epsilon \equiv \left[\epsilon_1^T, \ldots, \epsilon_{n_b}^T\right]^T \in \mathbb{R}^{4n_b}$. Instead of using quaternion derivatives $\dot{\epsilon}$, it is more advantageous to work with angular velocities $\bar{\omega}_j^T \in \mathbb{R}^3$ expressed in the local (body-attached) reference frames; in other words, the formulation described will use the vector of generalized velocities $\mathbf{v} = \left[\dot{\mathbf{r}}_1^T, \bar{\omega}_1^T, \ldots, \dot{\mathbf{r}}_{n_b}^T, \bar{\omega}_{n_b}^T\right]^T \in \mathbb{R}^{6n_b}$. Note that the generalized velocity can be easily obtained as $\dot{\mathbf{q}} = \mathbf{L}(\mathbf{q})\mathbf{v}$, where $\mathbf{L}$ is a linear mapping that transforms each $\bar{\omega}_i$ into the corresponding quaternion derivative $\dot{\epsilon}_i$ by means of the linear transformation $\dot{\epsilon}_i = \frac{1}{2}\mathbf{G}^T(\epsilon_i)\bar{\omega}_i$, with the 3×4 matrix $\mathbf{G}(\epsilon_i)$ defined as in [21]. We denote by $\mathbf{f}^A\left(t, \mathbf{q}, \mathbf{v}\right)$ the set of applied, or external, generalized forces.

Bilateral constraints representing kinematic pairs (e.g., spherical, prismatic, or revolute joints) lead to algebraic equations constraining the relative position of two rigid bodies. Specifically, the set $\mathcal{B}$ of bilateral constraints present in the system leads to the scalar algebraic equations $\Psi_i(\mathbf{q}, t) = 0$, $i \in \mathcal{B}$. Each constraint $i \in \mathcal{B}$ transmits reactions to the connected bodies by means of a multiplier $\widehat{\gamma}_{i,b}$. Assuming smoothness of the constraint manifold, $\Psi_i(\mathbf{q}, t)$ can be differentiated to obtain the Jacobian $\nabla_q \Psi_i = \left[\partial\Psi_i/\partial\mathbf{q}\right]^T$. In what follows we will also use the notation $\nabla\Psi_i^T \equiv \nabla_q\Psi_i^T \cdot \mathbf{L}(\mathbf{q})$.

Given a large number of rigid bodies with different shapes, modern collision-detection algorithms are able to efficiently find a set of contact points, that is, points where a *gap function*, $\Phi(\mathbf{q})$, can be defined for each pair of near-enough shape features. Where defined, such a gap function must satisfy the nonpenetration condition $\Phi(\mathbf{q}) \geq \mathbf{0}$ for all contact points.

When a contact i is active, that is, $\Phi_i(\mathbf{q}) = 0$, a normal force and a tangential friction force act on each of the two bodies at the contact point. In what follows, $\mathcal{A}(\mathbf{q}(t))$ denotes the set of all active contacts for a given configuration $\mathbf{q}$ of the system at time t.

We use the classical Coulomb friction model to define these forces [14]. If the contact is not active, that is, $\Phi_i(\mathbf{q}) > 0$, no contact or friction forces exist. This situation implies that the mathematical description of the model leads to a complementarity problem [13]. Consider two bodies A and B in contact as shown in Fig. 2. Let $\mathbf{n}_i$ be the normal at the contact pointing toward the exterior of the body of lower index, which by convention is considered to be body A. Let $\mathbf{u}_i$ and $\mathbf{w}_i$ be two vectors in the contact plane such that $\mathbf{n}_i, \mathbf{u}_i, \mathbf{w}_i \in \mathbb{R}^3$ are mutually orthonormal vectors. The frictional contact force is impressed on the system by means of multipliers $\widehat{\gamma}_{i,n} \geq 0$, $\widehat{\gamma}_{i,u}$, and $\widehat{\gamma}_{i,w}$, which lead to the normal component of the force $\mathbf{F}_{i,N} = \widehat{\gamma}_{i,n}\mathbf{n}_i$ and the tangential component of the force $\mathbf{F}_{i,T} = \widehat{\gamma}_{i,u}\mathbf{u}_i + \widehat{\gamma}_{i,w}\mathbf{w}_i$. The Coulomb model is expressed by using the maximum dissipation principle:

$$(\widehat{\gamma}_{i,u}, \widehat{\gamma}_{i,w}) = \underset{\sqrt{\widehat{\gamma}_{i,u}^2 + \widehat{\gamma}_{i,w}^2} \leq \mu_i \widehat{\gamma}_{i,n}}{\operatorname{argmin}} \quad \mathbf{v}_{i,T}^T \left(\widehat{\gamma}_{i,u}\mathbf{u}_i + \widehat{\gamma}_{i,w}\mathbf{w}_i\right). \tag{1}$$

The time evolution of the dynamical system is governed by the following differential problem with set-valued functions and complementarity constraints, which is equivalent to a differential

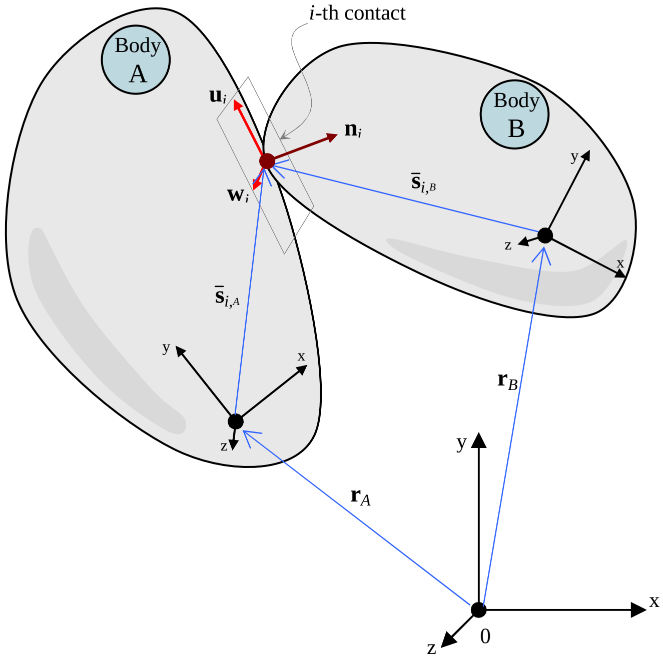


Figure 2. Contact i between two bodies $A, B \in \{1, 2, \ldots, n_b\}$

variational inequality [22]:

$$
\begin{aligned}
\dot{\mathbf{q}} \quad &= \quad \mathbf{L}(\mathbf{q})\mathbf{v} \\
\mathbf{M}\dot{\mathbf{v}} \quad &= \quad \mathbf{f}\,(t, \mathbf{q}, \mathbf{v}) + \sum_{i \in \mathcal{A}(\mathbf{q},\delta)} \left(\widehat{\gamma}_{i,n}\,\mathbf{D}_{i,n} + \widehat{\gamma}_{i,u}\,\mathbf{D}_{i,u} + \widehat{\gamma}_{i,w}\,\mathbf{D}_{i,w}\right) + \sum_{i \in \mathcal{B}} \widehat{\gamma}_{i,b} \nabla \Psi_i \\
i \in \mathcal{B} \quad &: \quad \Psi_i(\mathbf{q}, t) = 0 \qquad\qquad (2) \\
i \in \mathcal{A}(\mathbf{q}(t)) \quad &: \quad 0 \le \widehat{\gamma}_{i,n} \;\perp\; \Phi_i(\mathbf{q}) \ge 0, \qquad \text{and} \\
(\widehat{\gamma}_{i,u}, \widehat{\gamma}_{i,w}) \quad &= \quad \underset{\mu_i \widehat{\gamma}_{i,n} \ge \sqrt{(\widehat{\gamma}_{i,u})^2 + (\widehat{\gamma}_{i,w})^2}}{\operatorname{argmin}} \mathbf{v}^T \left(\widehat{\gamma}_{i,u}\,\mathbf{D}_{i,u} + \widehat{\gamma}_{i,w}\,\mathbf{D}_{i,w}\right)
\end{aligned}
$$

The tangent space generators $\mathbf{D}_i = [\,\mathbf{D}_{i,n},\ \mathbf{D}_{i,u},\ \mathbf{D}_{i,w}] \in \mathbb{R}^{6n_b \times 3}$ are defined as

$$
\mathbf{D}_i^T = \left[\begin{array}{ccccccccccc} \mathbf{0} & \ldots & -\mathbf{A}_{i,p}^T & \mathbf{A}_{i,p}^T \mathbf{A}_A \tilde{\bar{\mathbf{s}}}_{i,A} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{A}_{i,p}^T & -\mathbf{A}_{i,p}^T \mathbf{A}_B \tilde{\bar{\mathbf{s}}}_{i,B} & \ldots & \mathbf{0} \end{array}\right], \qquad (3)
$$

where $\mathbf{A}_{i,p} = [\mathbf{n}_i, \mathbf{u}_i, \mathbf{w}_i] \in \mathbb{R}^{3\times3}$ is the orientation matrix associated with contact i and the vectors $\bar{\mathbf{s}}_{i,A}$ and $\bar{\mathbf{s}}_{i,B}$ represent the contact point positions in body-relative coordinates as illustrated in Fig. 2.

The Coulomb model used in this work is the predominant model used in the engineering literature to describe dry friction. Unfortunately, the model may be inconsistent: configurations exist for which the resulting problem does not have a solution [11, 15]. This situation has led to the need to explore weaker formulations where the forces are measures and Newton's law is satisfied in a measure differential inclusion sense [15]. It has been shown that solutions in that sense do exist and can be found by time-stepping schemes [23].

Time-stepping scheme

The frictional contact dynamics problem formulated in terms of measure differential inclusions [15] is solved here by employing a time-stepping scheme that requires at each time step the solution of a complementarity problem. Specifically, given a position $\mathbf{q}^{(l)}$ and velocity $\mathbf{v}^{(l)}$ at time step

$t^{(l)}$, the numerical solution is found at the new time step $t^{(l+1)} = t^{(l)} + h$ by solving the following optimization problem with equilibrium constraints [24]:

$$\begin{aligned}
\mathbf{M}(\mathbf{v}^{(l+1)} - \mathbf{v}^{(l)}) &= h\mathbf{f}(t^{(l)}, \mathbf{q}^{(l)}, \mathbf{v}^{(l)}) + \sum_{i \in \mathcal{A}(q^{(l)},\delta)} (\gamma_{i,n}\,\mathbf{D}_{i,n} + \gamma_{i,u}\,\mathbf{D}_{i,u} + \gamma_{i,w}\,\mathbf{D}_{i,w}) \\
&\quad + \sum_{i \in \mathcal{B}} \gamma_{i,b} \nabla\Psi_i \qquad (4) \\
i \in \mathcal{B} &: \quad \frac{1}{h}\Psi_i(\mathbf{q}^{(l)}) + \nabla\Psi_i^T \mathbf{v}^{(l+1)} + \frac{\partial \Psi_i}{\partial t} = 0 \qquad (5) \\
i \in \mathcal{A}(q^{(l)}, \delta) &: \quad 0 \leq \frac{1}{h}\Phi_i(\mathbf{q}^{(l)}) + \mathbf{D}_{i,n}^T \mathbf{v}^{(l+1)} \perp \gamma_n^i \geq 0, \text{ and} \qquad (6) \\
(\gamma_{i,u}, \gamma_{i,w}) &= \underset{\mu_i \gamma_{i,n} \geq \sqrt{\gamma_{i,u}^2 + \gamma_{i,w}^2}}{\operatorname{argmin}} \quad \mathbf{v}^T\left(\gamma_{i,u}\,\mathbf{D}_{i,u} + \gamma_{i,w}\,\mathbf{D}_{i,w}\right) \qquad (7) \\
\mathbf{q}^{(l+1)} &= \mathbf{q}^{(l)} + h\mathbf{L}(\mathbf{q}^{(l)})\mathbf{v}^{(l+1)}. \qquad (8)
\end{aligned}$$

Here, γ_s represents the constraint impulse of a contact constraint; that is, $\gamma_s = h\widehat{\gamma}_s$, for $s = n, u, w$. The superscript $(l+1)$ on γ_s was dropped for notational brevity. The $\frac{1}{h}\Phi_i(\mathbf{q}^{(l)})$ term achieves constraint stabilization; its effect is discussed in [25]. Similarly, the term $\frac{1}{h}\Psi_i(\mathbf{q}^{(l)})$ achieves stabilization for bilateral constraints. The set of active unilateral constraint is denoted by $\mathcal{A}(q^{(l)}, \delta)$ to reflect the fact that at $t^{(l)}$ this set includes active as well as potential contacts between bodies that are less than a distance δ apart. The scheme converges to the solution of a measure differential inclusion when the step size $h \to 0$ [18].

Several approaches can be used to solve (4)–(7) and subsequently update the position configuration by using Eq. (8). Some authors suggested faceted pyramids to approximate friction cones so that the system of equations above, originally a nonlinear complementarity problem (NCP), turns into a linear complementarity problem (LCP) [14]. The resulting LCP can be solved by using pivoting or simplex methods. These numerical approaches, which belong to the class of direct methods, are computationally expensive and their complexity is in the worst case exponential [26]. Alternatively, the problem can be cast as a monotone optimization problem by introducing a relaxation over the complementarity constraints. Specifically, the time-stepping scheme is modified by replacing Eq. (6) with

$$i \in \mathcal{A}(q^{(l)}, \delta) : 0 \leq \frac{1}{h}\Phi_i(\mathbf{q}^{(l)}) + \mathbf{D}_{i,n}^T \mathbf{v}^{(l+1)} - \mu_i \sqrt{(\mathbf{v}^T\,\mathbf{D}_{i,u})^2 + (\mathbf{v}^T\,\mathbf{D}_{i,w})^2} \perp \gamma_n^i \geq 0\,. \qquad (9)$$

As $h \to 0$ the solution of the modified time-stepping scheme continues to approach the solution of the same measure differential inclusion as did the original numerical scheme [18]. It has been shown that the modified scheme is a CCP that can be solved by a family of iterative numerical methods that rely on projected contractive maps [19].

3. SOLVING THE CCP

The discussion here will concentrate only on the treatment of unilateral constraints. This is motivated by the observation that the number of unilateral constraints in real-life scenarios of interest (e.g., dynamics of granular material, granular terrain) dwarfs by five to six orders of magnitude the number of bilateral constraints. Moreover, the numerical solution challenges, from collision detection issues to scalability and convergence attributes, stem from difficulties associated with handling the unilateral constraints. Applications with tens of thousands of constraints for which the number of unilateral and bilateral constraints are comparable are currently effectively solved within the existing framework for DVI solution; see, for instance, [1, 25, 27, 2]. The numerical methods proposed here are immediately applicable to scenarios with bilateral constraints following the approach outlined, for instance, in [25]. Note that the no-bilateral-constraints assumption translates into $\mathcal{B} = \emptyset$, effectively eliminating Eq. (5) from the set of equations dealt with.

The following quantities will be used in posing in more compact form the CCP of interest: n_c is the number of active contacts in the system; $\mathbf{D} \equiv [\mathbf{D}_1, \cdots, \mathbf{D}_{n_c}] \in \mathbb{R}^{6n_b \times 3n_c}$ is the generalized contact transformation matrix; $\mathbf{D}_i \equiv [\mathbf{D}_{i,n}, \mathbf{D}_{i,v}, \mathbf{D}_{i,w}] \in \mathbb{R}^{6n_b \times 3}$ is the contact transformation matrix associated with contact $i \in \mathcal{A}(q^{(l)}, \delta)$; $\mathbf{r}_i \equiv \mathbf{b}_i + \mathbf{D}_i^T \mathbf{M}^{-1}\mathbf{f} \in \mathbb{R}^3$ is the generalized contact velocity for contact i; $\mathbf{b}_i \equiv \left[\frac{1}{h}\Phi_i(\mathbf{q}^{(l)}), 0, 0\right]^T \in \mathbb{R}^3$ is the unilateral constraint stabilization term; and $\mathbf{N} \equiv \mathbf{D}^T\mathbf{M}^{-1}\mathbf{D} \in \mathbb{R}^{3n_c \times 3n_c}$ is the contact associated symmetric positive-semidefinite Schur complement matrix, which is typically very sparse. The new quantities introduced – n_c, $\mathbf{D}$, $\mathbf{D}_i$, $\mathbf{r}_i$, $\mathbf{b}_i$, and $\mathbf{N}$ – should be further qualified by a superscript (l) to indicate that they are evaluated in the system configuration corresponding to t_l. For brevity, the superscript was omitted.

One can show that the CCP of Eqs. (4), (7), and (9) represents the first-order optimality condition of a constrained optimization problem with a quadratic cost function [25, 28]. This optimization problem, which must be solved iteratively at each time step of the dynamic simulation, assumes the form

$$\begin{aligned} \min q(\gamma) &= \frac{1}{2}\gamma^T\mathbf{N}\gamma + \mathbf{r}^T\gamma \\ & \text{subject to } \sqrt{{\gamma_{i,u}}^2 + {\gamma_{i,w}}^2} \leq \mu_i\gamma_{i,n} \quad \text{for } i = 1, 2, \ldots, n_c\ . \end{aligned} \tag{10}$$

Note that $\gamma = \left[\gamma_1^T, \gamma_2^T, \ldots, \gamma_{n_c}^T\right]^T$, where $\gamma_i = [\gamma_{i,n}, \gamma_{i,u}, \gamma_{i,w}]^T$ is the triplet expressing the magnitude of the contact impulses for contact i, i.e. $\gamma_n = h\widehat{\gamma}_{i,n}$, $\gamma_u = h\widehat{\gamma}_{i,u}$, $\gamma_w = h\widehat{\gamma}_{i,w}$.

Equation (10) effectively casts the DVI problem into an equivalent optimization problem. Just like any other solution method that relies on the DVI formulation (see, e.g., [29, 13, 30, 31, 32, 33]), this approach lacks the uniqueness attribute for the numerical solution both in force and in velocity distributions [15, 34, 35]. This issue can be traced back to the limitations associated with the rigid body model.[†] For this numerical investigation, the rigid body model limitations and ensuing lack of solution uniqueness will be controlled by comparing the methods analyzed here for benchmark tests with zero friction. The velocity distribution is now unique [15, 18] despite nonuniqueness of the force distribution, which can be traced back to the positive semi-definite attribute of $\mathbf{N}$. Consequently, when comparing different numerical methods (fixed-point iteration, Krylov subspace, or spectral methods), the convergence of the algorithms will be judged based on the value of the correction in velocities and not in $\gamma_{i,n}, i = 1, 2, \ldots, n_c$.

Note that when the mutual contact between bodies is characterized by $\mu_i = 0$, the friction cones degenerate into lines, and the CCP becomes a bound-constrained quadratic optimization problem, where $\gamma = [\gamma_{1,n}, \gamma_{2,n}, \ldots, \gamma_{n_c,n}]^T$, $\mathbf{N} \in \mathbb{R}^{n_c \times n_c}$, and $\mathbf{r} \in \mathbb{R}^{n_c}$. In this case, γ represents the vector of normal contact impulses. The problem in Eq. (10) assumes the form

$$\begin{aligned} \min q(\gamma) &= \frac{1}{2}\gamma^T\mathbf{N}\gamma + \mathbf{r}^{\mathbf{T}}\gamma \\ & \text{subject to } \gamma_{i,n} \geq 0 \quad \text{for } i = 1, 2, \ldots, n_c. \end{aligned} \tag{11}$$

This problem is typically solved by a projected-Jacobi or Gauss-Seidel method [19, 28], which has demonstrated poor convergence when the problem has bodies with vastly different inertia properties and/or when the problem size gets large (on the order of 1 million bodies). With the advent of high-speed parallel computing, this latter scenario is becoming more and more common [2].

4. SUMMARY OF ALGORITHMS CONSIDERED

Three algorithms are considered herein as candidates for solving the large scale CCP associated with many-body dynamics problems: the gradient projected minimum residual (GPMINRES) method, the preconditioned spectral projected gradient with fallback (P-SPG-FB) method, and the Kučera

[†] One simple illustration is the case of a perfectly rigid four-legged stool that is symmetric; it immediately leads to nonuniqueness in relation to the reaction force distribution in the four legs.

DOI: 10.1002/nme

method. The reference method is the Jacobi method currently implemented in the Chrono::Engine simulation package [36].

Jacobi

The algorithm currently used by the Chrono::Engine software [36] to solve the constrained quadratic optimization problem is based on the projected Jacobi method. Specifically, Chrono::Engine employs the following iterative scheme, where the superscript represents the iteration number:

$$\begin{aligned} \tilde{\gamma}^{r+1} &= \gamma^r + \omega \mathbf{B}[\mathbf{N}\gamma^r + \mathbf{r}] \qquad (12)\\ \hat{\gamma}^{r+1} &= \Pi_{\mathcal{K}}(\tilde{\gamma}^{r+1}) \\ \gamma^{r+1} &= \lambda \hat{\gamma}^{r+1} + (1-\lambda)\gamma^r \ . \end{aligned}$$

Here, $\Pi_{\mathcal{K}}(\tilde{\gamma})$ represents a projection operator. In the frictionless case, the projection is onto the non-negative numbers as seen in Eq. (11). For the frictionless case, the matrix $\mathbf{B}$ is defined as

$$\mathbf{B} = \begin{bmatrix} \eta_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \eta_{n_c} \end{bmatrix} ,$$

where

$$\eta_i = \frac{1}{\text{Trace}(\mathbf{D}_i^T \mathbf{M}^{-1} \mathbf{D}_i)} .$$

In terms of the other parameters, ω is chosen between 0 and 1, and typically for large problems $\omega \approx 0.3$. Note that when $\omega = 1$ and no friction is present in the problem, the iteration for $\tilde{\gamma}^{r+1}$ is *exactly* the Jacobi iteration. Also, λ, which represents a "damping" parameter, is typically 1.

Gradient Projected Minimum Residual

For the optimization problem defined in Eq. (11), the Karush-Kuhn-Tucker (KKT) first-order optimality conditions require that the projected gradient be zero:

$$\nabla_\Omega q(\gamma^{(opt)}) = \mathbf{0}_{nc} \ , \qquad (13)$$

where

$$[\nabla_\Omega q(\gamma)]_i \equiv \begin{cases} \frac{\partial q}{\partial \gamma_i}(\gamma) & \text{if} \quad \gamma_i > 0 \\ \min\{0, \frac{\partial q}{\partial \gamma_i}(\gamma)\} & \text{if} \quad \gamma_i = 0 \end{cases} . \qquad (14)$$

This algorithm, which uses the concept of projected gradient, draws on ideas presented in [37, 38] and seeks a solution by alternating two techniques: steepest descent and a Krylov subspace method. For the former, successive reductions of the cost function are accomplished by descending along the gradient. Based on a decision correlated to *(i)* the rate at which the cost function is reduced and *(ii)* the frequency at which the active set is changed, the projected gradient search is periodically replaced by a more aggressive search that draws on a Krylov subspace algorithm.

The projected gradient component at a step k starts by setting $\mathbf{y}^{(0)} = \gamma^{(k)}$. A refined value $\mathbf{y}^{(j+1)}$ is obtained by a projected descent step:

$$\mathbf{y}^{(j+1)} = \Pi_{\mathcal{K}}(\mathbf{y}^{(j)} - \alpha_j \nabla q(\mathbf{y}^{(j)})) \ , \qquad (15)$$

where $\Pi_{\mathcal{K}}(y)_i \equiv \max(0, y_i)$. The value α_j is selected so that

$$q(\mathbf{y}^{(j+1)}) \leq q(\mathbf{y}^{(j)}) + \mu \langle \nabla q(\mathbf{y}^{(j)}), \Pi_{\mathcal{K}}(\mathbf{y}^{(j)} - \alpha_j \nabla q(\mathbf{y}^{(j)})) - q(\mathbf{y}^{(j)}) \rangle \ , \qquad (16)$$

where $\mu \in (0, \frac{1}{2})$. More specifically, an optimal value $\alpha^\star$ is chosen to minimize the quadratic function $\alpha \to \mathbf{y}^{(j)} - \alpha \nabla q(\mathbf{y}^{(j)}) : \alpha > 0$. Then, α_j is computed by using the smallest l for which

DOI: 10.1002/nme

Eq. (16) holds with $\alpha_j = \alpha^\star \cdot \left(\frac{1}{2}\right)^l$, $l = 0, 1, \ldots$. This projected gradient technique is applied until one of the following two conditions is satisfied:

$$\begin{aligned} \mathcal{A}(\mathbf{y}^{(j+1)}) &= \mathcal{A}(\mathbf{y}^{(j)}) && (17) \\ q(\mathbf{y}^{(j)}) - q(\mathbf{y}^{(j+1)}) &\leq \eta_1 \max\{q(\mathbf{y}^{(l-1)}) - q(\mathbf{y}^{(l)}) : 1 \leq l < j\} \; . && (18) \end{aligned}$$

The first condition ensures that the projected gradient technique is abandoned when it slows in relation to changing the active set $\mathcal{A}(\mathbf{y}^{(j)})$. A similar strategy is pursued when the reduction in the cost function becomes sluggish. In Eq. (18) sluggishness is measured by the cost function reduction from iteration to iteration, for some tolerance $\eta_1 > 0$. If either condition is satisfied, the solution method switches to a Krylov subspace technique. To this end, we set $\gamma^{(k)} \equiv \mathbf{y}^{(j+1)}$ and search for $\bar{\mathbf{d}} \in \mathbb{R}^{n_c} : \bar{d}_i = 0, \;\; i \in \mathcal{A}(\gamma^{(k)})$ that minimizes the cost function $q(\gamma^{(k)} + \mathbf{d})$. This can be turned into an unconstrained optimization problem. To this end, we first define a matrix $\mathbf{Z}_k \in \mathbb{R}^{n_c \times m_k}$, where m_k is the number of free variables in $\gamma^{(k)}$ (i.e., variables $\gamma_l : l \in \mathcal{F}(\gamma^{(k)}) \equiv \{i : \gamma_i^{(k)} > 0\}$). Specifically, $\mathbf{Z}_k$ is defined by m_k columns of the identity matrix of dimension n_c associated with the set of free variables in $\gamma^{(k)}$. Minimizing $q(\gamma^{(k)} + \mathbf{d})$ with respect to $\mathbf{d}$ is equivalent to minimizing $q(\gamma^{(k)} + \mathbf{d}) - q(\gamma^{(k)})$. Defining $\mathbf{A}_k \equiv \mathbf{Z}_k^T \mathbf{N} \mathbf{Z}_k$, $\mathbf{r}_k \equiv \mathbf{Z}_k \nabla q(\gamma^{(k)})$, and $q_k(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T \mathbf{A}_k \mathbf{w} + \mathbf{w}^T \mathbf{r}_k$, we compute $\mathbf{d}_k$ as the solution of an unconstrained optimization problem:

$$\mathbf{d}_k = \mathbf{Z}_k \bar{\mathbf{w}}_k : \qquad \bar{\mathbf{w}}_k = \text{argmin } \; q_k(\mathbf{w}) \; . \tag{19}$$

In [37], finding $\bar{\mathbf{w}}_k$ relies on the conjugate gradient method. In fact, at step k of the algorithm, only several conjugate gradient iterations are carried out that stop short of finding the actual value $\bar{\mathbf{w}}_k$. The iterative process continues as long as the cost function reduction is vigorous, that is,

$$q_k(\mathbf{w}^{(j-1)}) - q_k(\mathbf{w}^{(j)}) \leq \eta_2 \max\{q_k(\mathbf{w}^{(lr-1)}) - q_k(\mathbf{w}^{(l)}), 1 \leq l < j\} \; , \tag{20}$$

for some constant $\eta_2 > 0$.

When the matrix in the cost function in Eq. (11) is positive definite, Burke and Moré proved that the algorithm above (i.e., the gradient projection combined with the conjugate gradient) has finite termination and converges to the unique solution of the problem [39]. For multibody dynamics problems the matrix $\mathbf{N}$ is positive semidefinite, leading to a convex problem that has multiple solutions. Consequently, the method proposed here no longer seeks a unique solution, and a decision was made to replace the conjugate gradient-based approach with a minimal residual Krylov subspace method. When the solution is unique, by virtue of the fact that a minimization of the error is producing the same solution as the minimization of the residual, the proposed algorithm called GPMINRES is conjectured to produce the unique solution in a finite number of steps. Thus, instead of using the conjugate gradient to solve the problem in Eq. (19), the approach adopted here minimizes the residual of $\nabla q_k(\mathbf{w}) = \mathbf{0}_{m_k}$ using MINRES [40].

The general algorithmic flow proceeds as follows:

```
ALGORITHM GPMINRES($\mathbf{N}$, $\mathbf{r}$, $\tau$, $\eta_1$, $\eta_2$, $N_{max}$, $M_{max}$)
    $\gamma^{(0)} := \mathbf{0}_{nc}$
    for $k := 0$ to $N_{max}$
        $\mathbf{y}^{(0)} = \gamma^{(k)}$
        while conditions in Eqs. (17) and (18) not violated
            $\mathbf{y}^{(j+1)} = \Pi_{\mathcal{K}}(\mathbf{y}^{(j)} - \alpha_j \nabla q(\mathbf{y}^{(j)}))$, see Eq. (15)
            $j = j + 1$
        endwhile
        $\gamma^{(k)} := \mathbf{y}^{(j)}$
        Determine active set $\mathcal{A}(\gamma^{(k)})$ and $\mathbf{Z}_k$ and $\mathbf{r}_k$ that go with the free set
        $\mathbf{w}_0 = \mathbf{0}_{m_k}$
        for $j := 0$ to $M_{max}$
            Improve value of $\mathbf{w}^{(j)} \rightarrow \mathbf{w}^{(j+1)}$ by taking one MINRES step
            $j = j + 1$
```

```
(14)            if condition in Eq. (20) violated
(15)                break
(16)        enfor
(17)        Set $\bar{\mathbf{w}}_k := \mathbf{w}^{(j)}$
(18)        Use backtracking line-search with direction $\mathbf{d}_k = \mathbf{Z}_k\bar{\mathbf{w}}_k$ to compute $\gamma^{(k+1)}$
(19)        if $||\nabla_\Omega q(\gamma^{(k+1)})||_\infty < \tau$ (see Eq. (13))
(20)            break
(21)    enfor
(22)    return Value at time step $t_{l+1}$, $\gamma^{l+1} := \gamma^{(k+1)}$ .
```

A more detailed version of this algorithm is provided in [20]. As discussed there, leaving the Krylov phase is not always done as indicated in Line (15) of the algorithm. This exit strategy is observed as long as the sets of active variables $\mathcal{A}(\gamma^{(k)})$ and binding variables $\mathcal{B}(\gamma^{(k)}) = \{i : \gamma_i = 0 \text{ and } \frac{\partial q}{\partial \gamma_i}(\gamma^{(k)}) \geq 0\}$ are different. Since the KKT conditions require that $\mathcal{A}(\gamma^{(k)}) = \mathcal{B}(\gamma^{(k)})$ at a solution [38], the Krylov subspace method is not abandoned even if the condition in Eq. (20) is violated as soon as the sets of active and binding variables are the same. Under these circumstances, the value of η_2 is reduced, and the Krylov subspace phase of the solution at iteration k is resumed right after Line (10). Note that the backtracking at Line (18) is done in order to get a valid $\gamma^{(k+1)}$. Specifically, starting with $\alpha = 1$, the sequence $\gamma^{(k+1)} \leftarrow \Pi_{\mathcal{K}}(\gamma^{(k)} + \alpha \mathbf{d}_k)$; $\alpha \leftarrow \alpha/2$, is carried out until $q(\mathbf{x}_{k+1}) \leq q(\mathbf{x}_k) + \mu\langle \nabla_q(\mathbf{x}_k), \mathbf{x}_{k+1} - \gamma_k \rangle$.

This work also investigated the use of preconditioning in the GPMINRES method to further increase solution speed. The resulting preconditioned algorithm, called GPMINRES-P, used a perturbed $\mathbf{LU}$ factorization when solving the subproblem, $\mathbf{A}_k\mathbf{w} = -\mathbf{r}_k$, with MINRES. In particular, each time the subproblem was solved the matrix $\mathbf{A}_k$ was factored as $\mathbf{A}_k = \mathbf{LU}$. To this end, the selected algorithm did not use pivoting. If a zero-entry was encountered on the diagonal during computation, it was perturbed to the local drop tolerance, and elimination continued as usual until the factors $\mathbf{LU}$ were completed. Therefore, the resulting factors satisfy $\mathbf{A}_k \approx \mathbf{LU}$. The implementation used MATLAB's ilu function, with arguments to force full factorization with no pivoting and perturbation of zero diagonal entries. This approach was selected because it could map well to parallel computation in future implementations, an important feature when dealing with many-body dynamics problems.

Preconditioned Spectral Projected Gradients with Fallback

The spectral projected gradient (SPG) method can be traced back to research on spectral-gradient (SG) methods [41], which were initially considered for unconstrained QPs. A generic proof of convergence was presented in [42]. Initially, the SG method was limited to the solution of linear problems such as those arising from unconstrained QPs, but its performance could not outperform the classic conjugate gradient method, which remained the de facto solver for that class of problems. Interest in the method was revived when a globalization strategy was added that enabled it to solve generic nonlinear optimization problems [43]. A further remarkable advancement was the projected version of the SG method, which is the SPG presented in [44]. The SPG method is able to solve convex-constrained optimization problems by performing a gradient projection at each step of the iteration. Since the method is nonmonotone, a line search with the Grippo-Lampariello-Lucidi (GLL) strategy has been suggested in [45].

```
ALGORITHM P-SPG-FB($\mathbf{N}$, $\mathbf{r}$, $\gamma^{(0)}$, $\mathcal{K}$, $\mathbf{P} \mapsto \gamma$)
    $\gamma^{(0)} := \Pi_{\mathcal{K}}(\gamma^{(0)})$, $\gamma_{FB} = \gamma^{(0)}$, $\breve{\alpha}^{(0)} \in [\alpha_{min}, \alpha_{max}]$, $\xi \in [0, 1]$
    $\mathbf{g}^{(0)} := \mathbf{N}\gamma^{(0)} + \mathbf{r}$, $f(\gamma^{(0)}) = \frac{1}{2}{\gamma^{(0)}}^T\mathbf{N}\gamma^{(0)} + {\gamma^{(0)}}^T\mathbf{r}$, $w^{(0)} = 10^{29}$
    for $j := 0$ to $N_{max}$
        $\mathbf{p}^{(j)} = \mathbf{P}^{-1}\mathbf{g}^{(j)}$
        $\mathbf{d}^{(j)} = \Pi_{\mathcal{K}}(\gamma^{(j)} - \breve{\alpha}^{(j)}\mathbf{p}^{(j)}) - \gamma^{(j)}$
        if $\langle \mathbf{d}^{(j)}, \mathbf{g}^{(j)} \rangle \geq 0$
        $\mathbf{d}^{(j)} = \Pi_{\mathcal{K}}(\gamma^{(j)} - \breve{\alpha}^{(j)}\mathbf{g}^{(j)}) - \gamma^{(j)}$
    $\lambda := 1$
    while line search
        $\gamma^{(j+1)} := \gamma^{(j)} + \lambda \mathbf{d}^{(j)}$
        $\mathbf{g}^{(j+1)} := \mathbf{N}\gamma^{(j+1)} + \mathbf{r}$
        $f(\gamma^{(j+1)}) = \frac{1}{2}{\gamma^{(j+1)}}^T \mathbf{N}\gamma^{(j+1)} + {\gamma^{(j+1)}}^T \mathbf{r}$
        if $f(\gamma^{(j+1)}) > \max_{i=0,..,\min(j,N_{GLL})} f(\gamma^{(j-i)}) + \xi\lambda \left\langle \mathbf{d}^{(j)}, \mathbf{g}^{(j)} \right\rangle$
            define $\lambda_{\text{new}} \in [\sigma_{\min}\lambda, \sigma_{\max}\lambda]$ and repeat line search
        else
            terminate line search
    $\mathbf{s}^{(j)} = \gamma^{(j+1)} - \gamma^{(j)}$
    $\mathbf{y}^{(j)} = \mathbf{g}^{(j+1)} - \mathbf{g}^{(j)}$
    if $j$ is odd
        $\breve{\alpha}^{(j+1)} = \frac{\left\langle \mathbf{s}^{(j)}, \mathbf{P}\mathbf{s}^{(j)} \right\rangle}{\left\langle \mathbf{s}^{(j)}, \mathbf{y}^{(j)} \right\rangle}$
    else
        $\breve{\alpha}^{(j+1)} = \frac{\left\langle \mathbf{s}^{(j)}, \mathbf{y}^{(j)} \right\rangle}{\left\langle \mathbf{y}^{(j)}, \mathbf{P}^{-1}\mathbf{y}^{(j)} \right\rangle}$
    $\breve{\alpha}^{(j+1)} = \min(\alpha_{\max}, \max(\alpha_{\min}, \breve{\alpha}^{(j+1)}))$
    $w^{(j+1)} = \left|\left| \left[\gamma^{(j+1)} - \Pi_{\mathcal{K}}(\gamma^{(j+1)} - \tau_g \mathbf{g}^{(j+1)})\right] / \tau_g \right|\right|_2 = ||\epsilon||_2$
    if $w^{(j+1)} \leq \min_{k=0,..,j} w^{(k)}$
        $\gamma_{FB} = \gamma^{(j+1)}$
return $\gamma_{FB}$
```

This algorithm inherits the properties of the SPG method presented in [44] and adds preconditioning and a safe fallback strategy, hence the name P-SPG-FB. The method requires the following parameters: two safeguards $\alpha_{\min}$ and $\alpha_{\max}$ for the spectral step length (respectively 10^{-9} and 10^9 in our tests), two safeguards for the line search $0 < \sigma_{\min} < \sigma_{\max} < 1$, an integer N_{GLL} to accommodate nonmonotone steps within the GLL line search (a value about 10 works well in most cases), the Armijo sufficient decrease parameter $\xi \in [0, 1]$, usually very low, and a small value τ_g for the fallback strategy.

Our implementation of preconditioning for the P-SPG-FB method is inspired by the scheme introduced in [46]; early preconditioning ideas for SG methods in general are discussed in [47]. We recall that the goal of preconditioning is to cluster the eigenvectors of the $\mathbf{N}$ matrix, trying to reduce its condition number $\kappa(\mathbf{N})$ by solving an equivalent problem with unknowns $\breve{\gamma}$. By adopting a left-right symmetry-preserving preconditioning [40], we define

$$\breve{\mathbf{N}} = \mathbf{L}^{-1}\mathbf{N}\mathbf{L}^{-T}, \quad \breve{\mathbf{r}} = \mathbf{L}^{-1}\mathbf{r}, \quad \breve{\gamma} = \mathbf{L}^T\gamma \; . \tag{21}$$

One can rewrite the original SG method using $\breve{\mathbf{N}}$, $\breve{\mathbf{r}}$, and $\breve{\gamma}$, substituting terms in Eq. (21) and showing that there is no need of using the $\mathbf{L}$ matrix explicitly: a more efficient version uses only the $\mathbf{P} = \mathbf{L}\mathbf{L}^T$ matrix and operates directly on the original γ unknowns. Differently from [47], we alternate two formulas for the computation of the spectral step size, as suggested in [48]: $\alpha_{BB1} = \langle \mathbf{s}, \mathbf{s} \rangle \,/\, \langle \mathbf{s}, \mathbf{y} \rangle$ and $\alpha_{BB2} = \langle \mathbf{s}, \mathbf{y} \rangle \,/\, \langle \mathbf{y}, \mathbf{y} \rangle$. In the preconditioned case,

$$\breve{\alpha}_{BB1} = \frac{\langle \breve{\mathbf{s}}, \breve{\mathbf{s}} \rangle}{\langle \breve{\mathbf{s}}, \breve{\mathbf{y}} \rangle} = \frac{\mathbf{s}^T\mathbf{L}\mathbf{L}^T\mathbf{s}}{\mathbf{s}^\mathbf{T}\mathbf{L}\mathbf{L}^{-\mathbf{1}}\mathbf{y}} = \frac{\mathbf{s}^\mathbf{T}\mathbf{P}\mathbf{s}}{\mathbf{s}^\mathbf{T}\mathbf{y}} \tag{22}$$

$$\breve{\alpha}_{BB2} = \frac{\langle \breve{\mathbf{s}}, \breve{\mathbf{y}} \rangle}{\langle \breve{\mathbf{y}}, \breve{\mathbf{y}} \rangle} = \frac{(\mathbf{L}^T\mathbf{s})^\mathbf{T}(\mathbf{L}^{-\mathbf{1}}\mathbf{y})}{(\mathbf{L}^{-\mathbf{1}}\mathbf{y})^\mathbf{T}(\mathbf{L}^{-\mathbf{1}}\mathbf{y})} = \frac{\mathbf{s}^\mathbf{T}\mathbf{y}}{\mathbf{y}^\mathbf{T}\mathbf{P}^{-\mathbf{1}}\mathbf{y}}. \tag{23}$$

We noticed that alternating $\breve{\alpha}_{BB1}$ and $\breve{\alpha}_{BB2}$ gives about the same rate of convergence of $\breve{\alpha}_{BB1}$ alone, or even less; however, the alternating scheme usually produces a smoother nonmonotone descent, thus allowing a more frequent update of the fallback vector γ_{FB}.

The continuous nonexpansive projection operator $\Pi(\cdot)$ is a mapping that satisfies $\Pi(\gamma)_{\mathcal{K}} = \underset{\mathbf{z}\in\mathcal{K}}{\arg\min}||\mathbf{z} - \gamma||$. In case of frictional contact simulations, this is a projection onto the simple second-order Lorentz cones. The computational overhead is linear with the number of contacts and thus almost negligible. For the preconditioned iteration, one could operate $\Pi(\breve{\gamma})_{\breve{\mathcal{K}}}$ or use the approach of [46] that projects γ on the original set $\mathcal{K}$; the latter option is easy to implement, but in some cases it might violate the Wolfe condition on the descent direction [49]. In [46] this is fixed by switching to a nonpreconditioned gradient if the Wolfe condition fails, and by turning on the preconditioner mostly during the last iterations. To avoid these problems, we use a custom diagonal preconditioner $\mathbf{P} = \overline{\mathrm{diag}}(\mathbf{A})$, where the diagonal elements relative to the same subset are averaged. In practical terms, for a problem with n_c frictional contacts, n_c triplets on the diagonal are averaged and inserted on the diagonal of $\mathbf{P}$. This $\mathbf{P} = \overline{\mathrm{diag}}(\mathbf{A})$ preconditioner is simple and can be implemented easily, without major impact on the computational times. Despite its simplicity, this type of preconditioner was able to improve the convergence of the P-SPG-FB method in various benchmarks, especially those that involved many light objects mixed with few heavy objects, stacked on flat surfaces. In other, less frequent cases, this preconditioner gave no major benefits or even produced worse convergence.

An important issue is that a fallback strategy is needed because the method is nonmonotone: in all SG methods the residual runs through various unpredictable peaks before converging to the stationary point. This is not a problem if one runs the iterations up to the exact solution; however, if one wishes to truncate prematurely the iteration because of real-time requirements, for instance as in vehicle simulators or video gaming, the last computed value $\gamma^{(j)}$ might not be the best choice. In case of an unconstrained QP, it is sufficient to resort to a past $\gamma^{(j)}$ whose $f(\gamma)$ was minimum, but for the generic case of convex constraints, this is not true because a vector might give very low $f(\gamma)$ with $\gamma \in \mathcal{K}$ and yet violate significantly the first order optimality condition. A better indicator comes from the fact that when approaching a stationary point $\left|\left|\gamma^{(j+1)} - \Pi(\gamma^{(j+1)} - \mathbf{g}^{(j+1)})\right|\right|_2$ becomes very small. In the algorithm we use a similar criterion, but we scale the gradient by τ_g and we postscale its projection by τ_g because it gives more reliable results when γ is still far from the stationary point:

$$||\epsilon||_2 \equiv \left|\left|\left[\gamma^{(j+1)} - \Pi_{\mathcal{K}}(\gamma^{(j+1)} - \tau_g \mathbf{g}^{(j+1)})\right] / \tau_g\right|\right|_2. \tag{24}$$

The Kučera Method

This algorithm draws on an approach for minimizing quadratic cost functions with separable convex constraints [50]. While the more general algorithm handles any separable convex constraints, the algorithm outlined here handles the case of Eq. (11) where all variables are subject to non-negativity constraints. We rely on the same condition for optimality stated in Eq. (13). An alternative definition of the projected gradient, $\tilde{\mathbf{g}} = \tilde{\mathbf{g}}(\gamma^{(k)})$, is used, where once again $\Pi_{\mathcal{K}}(y)_i \equiv \max(0, y_i)$, and $0 < \tilde{\alpha} \leq \|\mathbf{N}\|^{-1}$ is a constant step-size, usually taken as $\tilde{\alpha} = \|\mathbf{N}\|^{-1}$.

$$\tilde{\mathbf{g}}(\gamma^{(k)}) = \frac{1}{\tilde{\alpha}}(\gamma^{(k)} - \Pi_{\mathcal{K}}(\gamma^{(k)} - \tilde{\alpha}\mathbf{g}(\gamma^{(k)}))) \tag{25}$$

The projected gradient $\tilde{\mathbf{g}} = \tilde{\mathbf{g}}(\gamma^{(k)})$ can be decomposed into the projected free gradient $\tilde{\phi} = \tilde{\phi}(\gamma^{(k)})$ and the projected boundary gradient $\tilde{\beta} = \tilde{\beta}(\gamma^{(k)})$. Note that the subscript i indicates the ith component of a vector.

$$\tilde{\phi}_i = \tilde{\mathbf{g}}_i \text{ for } i \in \mathcal{F}(\gamma^{(k)}), \quad \tilde{\phi}_i = 0 \text{ for } i \in \mathcal{A}(\gamma^{(k)}) \tag{26}$$

$$\tilde{\beta}_i = 0 \text{ for } i \in \mathcal{F}(\gamma^{(k)}), \quad \tilde{\beta}_i = \tilde{\mathbf{g}}_i \text{ for } i \in \mathcal{A}(\gamma^{(k)}) \tag{27}$$

The gradient $\mathbf{g} = \nabla q(\gamma^{(k)})$ can be similarly decomposed. The free gradient $\phi = \phi(\gamma^{(k)})$ will be used in the remainder of the algorithm. Its ith component is defined as

$$\phi_i = \mathbf{g}_i \text{ for } i \in \mathcal{F}(\gamma^{(k)}), \phi_i = 0 \text{ for } i \in \mathcal{A}(\gamma^{(k)}) \tag{28}$$

DOI: 10.1002/nme

In general, the algorithm chooses one of three steps for each iteration. These steps are the expansion step, which may add indices to the active set; the proportioning step, which may release indices from the active set; and the conjugate gradient step, which minimizes the objective function value given the current active set. If $\tilde{\beta}(\gamma^{(k)})^T\mathbf{g}(\gamma^{(k)}) \leq \Gamma^2\tilde{\phi}(\gamma^{(k)})^T\mathbf{g}(\gamma^{(k)})$, then $\gamma^{(k)}$ is called strictly proportional, and $\gamma^{(k+1)}$ is generated by a conjugate gradient step if doing so maintains $\gamma^{(k+1)} \in \Omega$, or by the expansion step if the conjugate gradient step would end up outside the feasible region. If $\gamma^{(k)}$ is not strictly proportional, $\gamma^{(k+1)}$ is generated by the proportioning step. Here, Γ is a constant that controls how willing the method is to release indices from the active set. Good performance has been achieved with $\Gamma = 1$.

Once again, the conjugate gradient steps minimize the objective funtion for a given active set. For this situation to occur, a step in a conjugate gradient direction $\mathbf{p}^{(k)}$ should not alter the active set. In other words, we require $\mathbf{p}_i^{(k)} = 0$ for $i \in \mathcal{A}(\gamma^{(k)})$. To this end, we start an unbroken chain of conjugate gradient iterations with $\mathbf{p}^{(s)} = \phi(\gamma^{(s)})$ and use the following:

$$\mathbf{p}^{(k)} = \phi(\gamma^{(k)}) - \rho^{(k)}\mathbf{p}^{(k-1)}, \rho^{(k)} = \frac{\phi(\gamma^{(k)})^T\mathbf{N}\mathbf{p}^{(k-1)}}{(\mathbf{p}^{(k-1)})^T\mathbf{N}\mathbf{p}^{(k-1)}}, k > s \ . \tag{29}$$

Note that if the active set changes, the conjugate gradient direction must be restarted. With these preliminaries, the algorithm, which was proved in [50] to have a linear convergence rate in terms of the spectral condition number of the matrix $\mathbf{N}$, proceeds as follows:

```
ALGORITHM KUCERA($\mathbf{N}$, $\mathbf{r}$, $\gamma^{(0)}$, $\Gamma > 0$, $\tilde{\alpha} \in (0, \|\mathbf{N}\|^{-1}]$, $\epsilon > 0$)
 $k = 0$
 $\mathbf{g} = \mathbf{N}\gamma^{(0)} + \mathbf{r}$
 $\mathbf{p} = \phi(\gamma^{(0)})$
 while $\|\tilde{\mathbf{g}}(\gamma^{(k)})\| > \epsilon$
     if $\tilde{\beta}(\gamma^{(k)})^T\mathbf{g}(\gamma^{(k)}) \leq \Gamma^2\tilde{\phi}(\gamma^{(k)})^T\mathbf{g}(\gamma^{(k)})$
         $\alpha_{cg} = \mathbf{g}^T\mathbf{p}/\mathbf{p}^T\mathbf{N}\mathbf{p}$
         $\alpha_f = min(\alpha_{f,i})$ where $\alpha_{f,i} = \begin{cases} \gamma_i^{(k)}/\mathbf{p}_i, & \text{if } \mathbf{p}_i > 0 \\ \infty, & \text{if } \mathbf{p}_i \leq 0 \end{cases}$
         if $\alpha_{cg} < \alpha_f$
             $\gamma^{(k+1)} = \gamma^{(k)} - \alpha_{cg}\mathbf{p}$
             $\mathbf{g} = \mathbf{g} - \alpha_{cg}\mathbf{N}\mathbf{p}$
             $\rho = \phi(\gamma^{(k+1)})^T\mathbf{N}\mathbf{p}/\mathbf{p}^T\mathbf{N}\mathbf{p}$
             $\mathbf{p} = \phi(\gamma^{(k+1)}) - \rho\mathbf{p}$
         else
             $\gamma^{(k+1/2)} = \gamma^{(k)} - \alpha_f\mathbf{p}$
             $\gamma^{(k+1)} = \gamma^{(k+1/2)} - \tilde{\alpha}\tilde{\phi}(\gamma^{(k+1/2)})$
             $\mathbf{g} = \mathbf{N}\gamma^{(k+1)} + \mathbf{r}$
             $\mathbf{p} = \phi(\gamma^{(k+1)})$
     else
         $\gamma^{(k+1)} = \gamma^{(k)} - \tilde{\alpha}\tilde{\beta}(\gamma^{(k)})$
         $\mathbf{g} = \mathbf{N}\gamma^{(k+1)} + \mathbf{r}$
         $\mathbf{p} = \phi(\gamma^{(k+1)})$
     $k = k + 1$
 return $\gamma^{(k)}$
```

5. PERFORMANCE INVESTIGATION

The purpose of this section is to compare the iterative methods discussed for solving the quadratic optimization problem associated with the frictionless contact problem (see Eq. (11)). The data for all tests in this section was generated in Chrono::Engine [36]. All numerical experiments include a projected variant of the Jacobi solver as the reference solver. The "reference solver" choice is

Table I. Performance of iterative methods on Test Problem 1, a system with 1,000 bodies that led to a cost function in 3,525 variables. Notation used: FOFV - final objective function value; $N \supset M$ in the second column stands for "N MinRes iterations within M changes of active set". Reported solution time (Sol. Time) is in seconds.

Method	**Iterations**	**FOFV**	γ_{min}	γ_{max}	**Sol. Time (sec.)**
GPMINRES	$1000 \supset 100$	-2.9035	0.0	7.7487	6.70
GPMINRES	$10000 \supset 1000$	-2.9045	0.0	8.2002	61.07
GPMINRES-P	$100 \supset 100$	-2.8854	0.0	6.8551	1675
Jacobi	1000	-2.5077	0.0	4.4961	3.66
Jacobi	10000	-2.8983	0.0	7.4953	24.66
Jacobi	43000	-2.9035	0.0	7.9254	95.41
Jacobi	100000	-2.9043	0.0	8.0619	231.7

motivated by the observation that the Jacobi solver, unlike the marginally more efficient Gauss-Seidel approach, is more amenable to parallel computing; see, for instance, [51, 52, 53, 2]. For each test, a Chrono::Engine simulation was run in which a collection of spheres was allowed to settle within a fixed boundary. After the spheres had nearly settled, the simulation was frozen so that the optimization problem associated with the current time step could be extracted. The quadratic optimization problem solved is completely specified by the matrix $\mathbf{N}$ and the vector $\mathbf{r}$; see Eq. (11). For comparison, the Jacobi, GPMINRES, SPG-FB, and Kučera methods were all implemented in MATLAB.

Test Problem 1

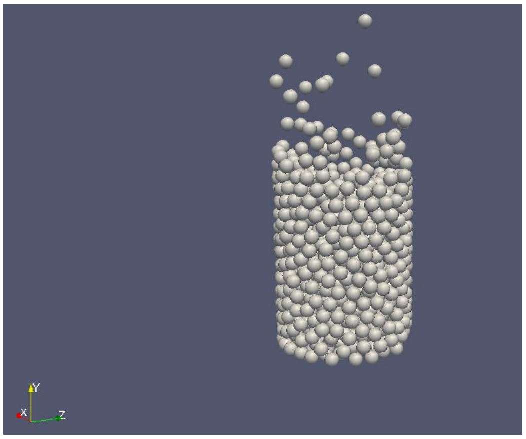


Figure 3. Test Problem 1, containing 1,000 bodies.

Test Problem 1 represented a system with 1,000 bodies and 3,525 contacts. The state of the system can be seen in Fig. 3. The problem was solved with GPMINRES with and without preconditioning (GPMINRES-P and GPMINRES, respectively) and Jacobi methods. Figure 4 shows the objective function value plotted versus the iteration number. Each dot in the figure represents an active set, or a certain subproblem. Note that for GPMINRES, 10 MINRES iterations were performed for each subproblem. With GPMINRES-P, the preconditioning allowed the subproblem to be solved more accurately in one iteration for each subproblem. The data plotted in Fig. 4 corresponds to the data in Table I. Specifically, the data from rows 2, 3, and 5 are plotted. However, note that the data in row 1, for example, simply comes from stopping the iterative process after 1,000 iterations, whereas the data in row 2 would be reached by performing 9,000 further iterations to reach 10,000 total MINRES iterations.

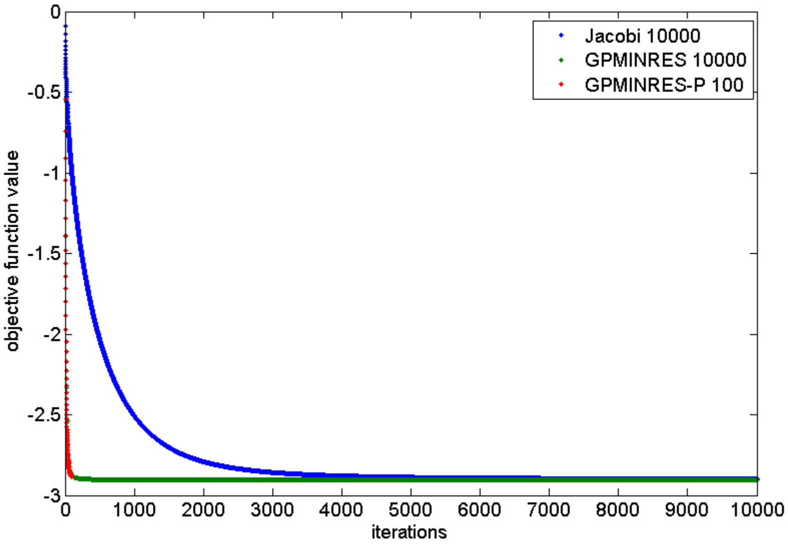


Figure 4. Objective function value of various iterative methods for Test Problem 1 (1,000 bodies/3,525 contacts)

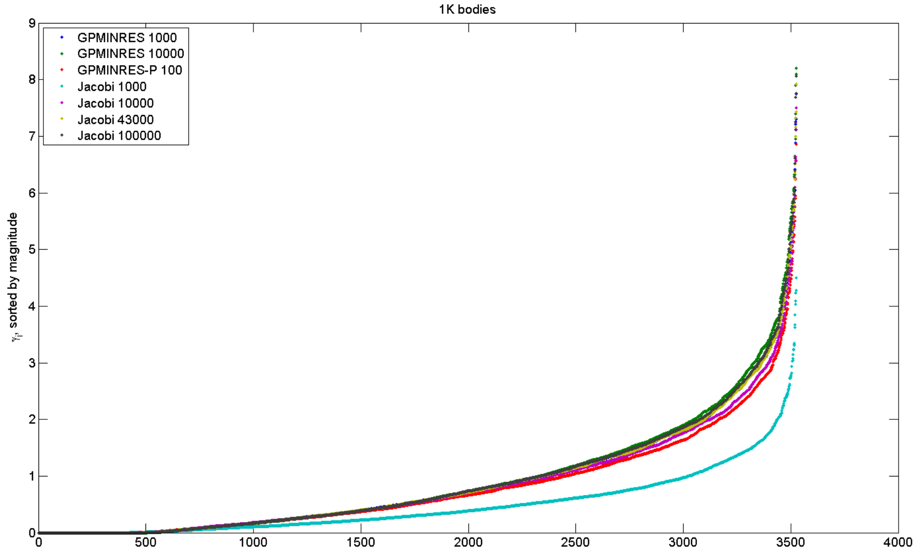


Figure 5. Comparison of solution vector γ for Test Problem 1 (1,000 bodies/3,525 contacts). Here, the solution vector is sorted by magnitude for several methods and different numbers of iterations, corresponding to Table I. Each point is the magnitude of one γ_i.

The performance of the iterative methods on Test Problem 1 is summarized in Table I. Note that preconditioning is expensive in this implementation. Additionally, preconditioning does not significantly improve the solution. In fact, the solution achieved without preconditioning after 1,000 total MINRES iterations (within 100 changes of the active set) was both more accurate and faster than that achieved with preconditioning after 100 total MINRES iterations (within 100 changes of the active set). Also, note that GPMINRES achieved the same objective function value after 1,000 iterations as Jacobi did after 43,000 iterations, and did so over 14 times faster in terms of computation speed. One also can observe in Fig. 5 that the solution vector becomes "sharper" as

DOI: 10.1002/nme

more iterations are performed, that is, as the solution converges, the maximum magnitude of the normal contact impulses in the system increases toward the correct value.

5.1. Test Problem 2

Test Problem 2 was used to gauge the performance of the algorithms as the problem size increased. Each test in this set consisted of a cylindrical boundary with radius 2 m and a collection of spherical bodies of radius 0.1 m and mass 6.28 kg. The three tests had 1,000, 2,000, and 4,000 bodies, with 3,634, 8,507, and 17,161 contacts, respectively. For each test all four algorithms were tested with initial guess $\mathbf{x}^0 = \mathbf{0}$ for an equal number of iterations. For each test, the objective function value and the residual $||\epsilon||_2$ (evaluated as indicated in Eq. (24)) are plotted.

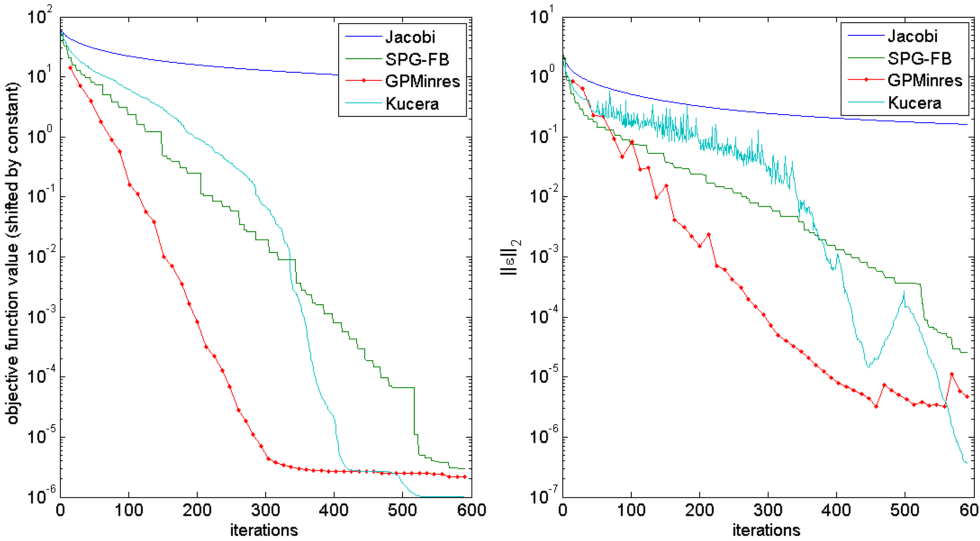


Figure 6. Results for System 1 of Test Problem 2 with $n_c = 3634$: Objective function value shifted by a constant and plotted on log axes; Residual $||\epsilon||_2$ (evaluated as indicated in Eq. (24)).

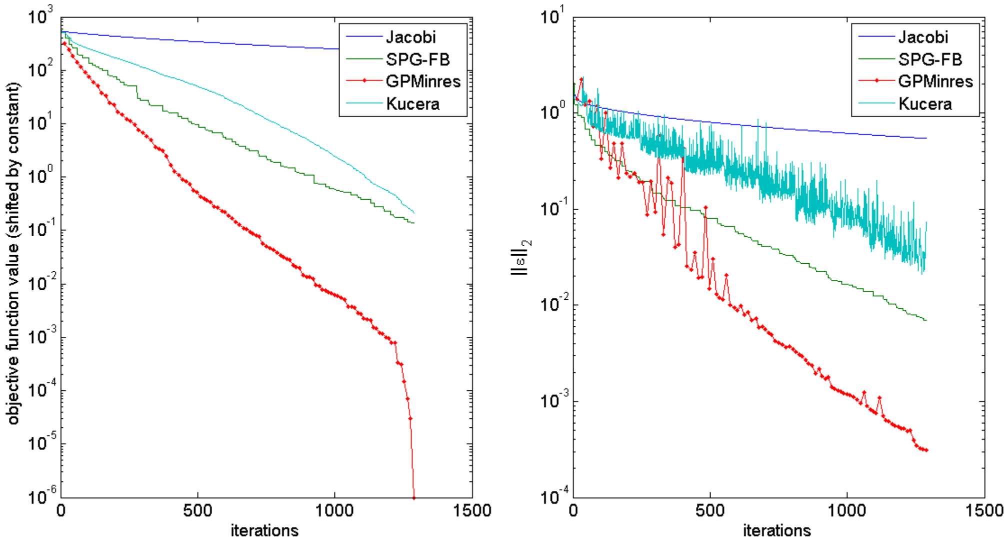


Figure 7. Results for System 2 of Test Problem 2 with $n_c = 8507$: Objective function value shifted by a constant and plotted on log axes; Residual $||\epsilon||_2$ (evaluated as indicated in Eq. (24)).

DOI: 10.1002/nme

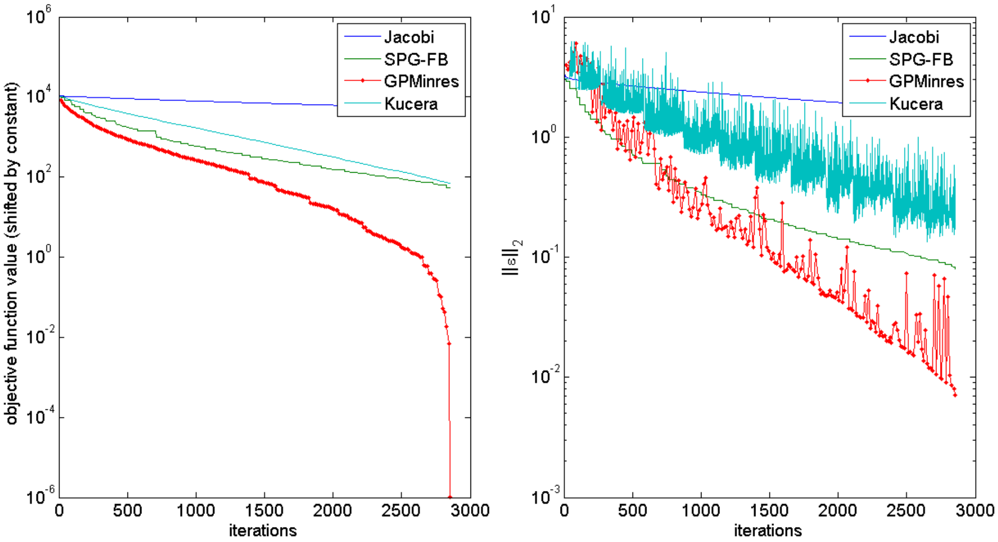


Figure 8. Results for System 3 of Test Problem 2 with $n_c = 17161$: Objective function value shifted by a constant and plotted on log axes; Residual $||\epsilon||_2$ (evaluated as indicated in Eq. (24)).

The results, plotted in Figs. 6-8, indicate that the SPG-FB, GPMINRES, and Kučera algorithms all perform significantly better than Jacobi for all systems in Test Problem 2 in terms of both objective function and residual value. Among these three algorithms, SPG-FB performs best at the beginning of the iterative process, but its performance degrades more quickly than that of GPMINRES. This result can be observed in the residual plots for all systems in this test set. We note that Jacobi and SPG-FB are monotone in terms of both the objective function and the residual. This property is advantageous when one may need to terminate the iterative process at an arbitrary point.

A measure of feasibility at the end of the iterative process is plotted in Fig. 9 for System 1 of Test Problem 2. This figure shows, for each method considered, the mean, standard deviation, and maximum feasibility defined as $f_i = \gamma(\mathbf{N}\gamma_i + \mathbf{r})_i$. Note that f should be identically zero if the complementarity problem is exactly satisfied.

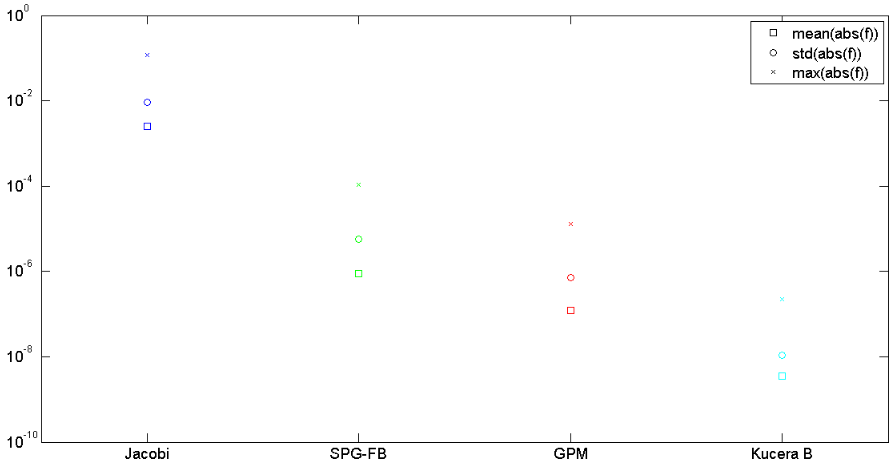


Figure 9. Feasibility at end of iterative processfor System 1 of Test Problem 2, where $f_i = \gamma(\mathbf{N}\gamma + \mathbf{r})_i$.

DOI: 10.1002/nme

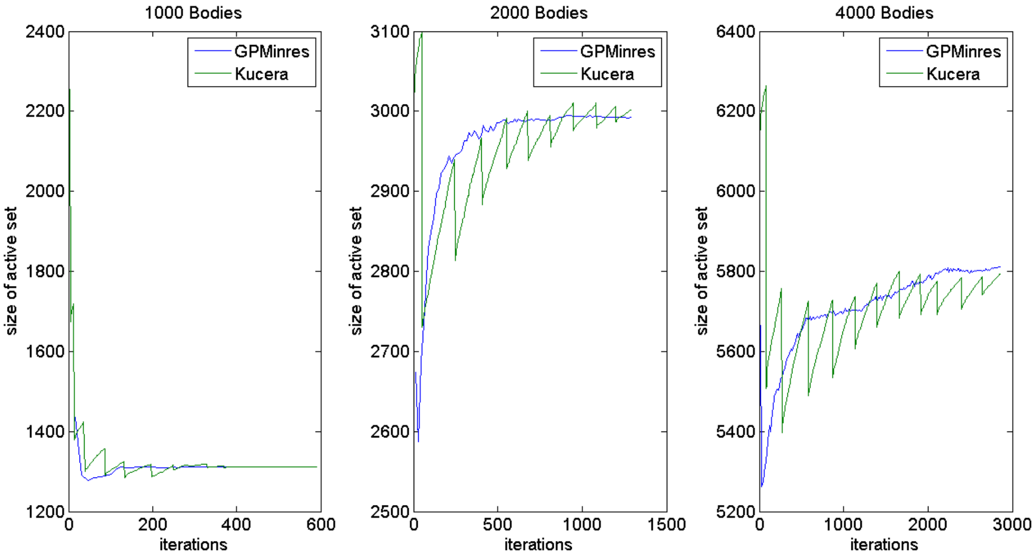


Figure 10. Size of the active set at each iteration for Test Problem 2 with $n_c = 3634$, $n_c = 8507$, and $n_c = 17161$.

The active set methods GPMINRES and Kučera can be compared by the size of the active set at each iteration. Note that we consider only the size of the active set here and not the actual contents. This data for the three systems of Test Problem 2 can be seen in Fig. 10. Note that Kučera experiences larger changes in the active set than does GPMINRES during the solution process. Additionally, GPMINRES seems to settle on an active set more quickly than Kučera. Note that the size of the active set can be unsettled even as the objective function approaches the minimum. This phenomenon is best observed in the results from the larger systems of this test problem. In fact, even if the size of the active set is constant, the contents of the active set can still be unsettled because of the redundancy of the contacts in the physical system. This also indicates that the stopping criteria for the iterative process should not be based on the active set, but rather on a measure of the change in velocity from the previous iteration or on a measure of the projected gradient. Recall that the velocity solution is unique, while the contact impulse solution, γ, is not.

5.2. *Test Problem 3*

Test Problem 3 was used to gauge the performance of the algorithms in a scenario where a heavy object was supported by relatively light and small spherical bodies, a situation encountered when simulating a tracked or wheel vehicle moving over granular terrain, see Fig. 1(a) [5]. Here, a cylindrical boundary with radius 1.5 m was filled with 1,000 spherical particles with radius 0.1 m and mass 6.28 kg. A cylindrical mass of radius 1.125 m was dropped on top of the bodies (see Fig. 11). Each system in this test set represents the same scenario, with the mass of the cylinder body increasing. The three systems set the mass of the cylinder to 1000 kg, 4000 kg, and 16000 kg respectively. This scenario had 4,295 contacts.

For this test a normalized objective function value was used in postprocessing. For each mass of the heavy cylinder; i.e., for each of the three systems, a scaling factor was found such that the minimum achieved objective function among all algorithms was -1. A different scaling factor was necessary for each mass test. This normalization allows comparisons between algorithms for a given mass test and between mass tests. The results can be seen in Fig. 12. Here, comparisons for a given mass test can be made by considering lines of the same color, while comparisons for a given algorithm with increasing applied mass can be made by considering lines of the same style. Once again, GPMINRES performs best for all three scenarios. Further, the performance degradation with increasing mass is less significant for GPMINRES than for other algorithms.

DOI: 10.1002/nme

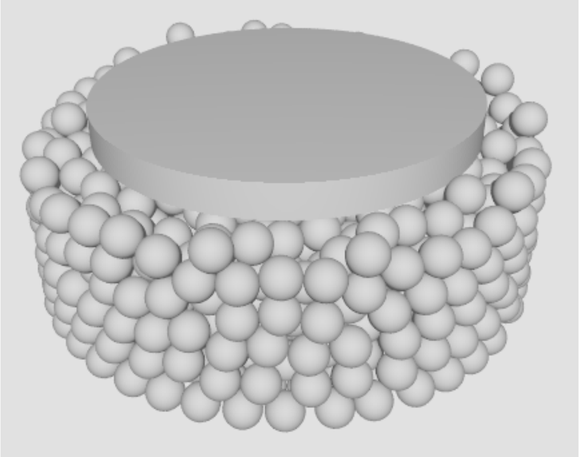


Figure 11. System for Test Problem 3 with 1,001 bodies and $n_c = 4295$. The mass of the cylindrical body was set to 1000 kg, 4000 kg, and 16000 kg for the three tests in this set.

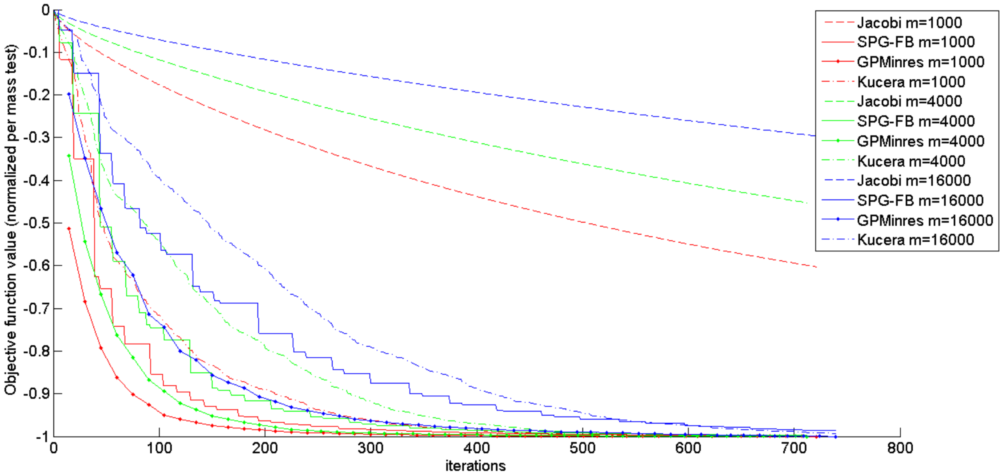


Figure 12. Normalized objective function values for Test Problem 3 ($n_c = 4295$), with 1000, 4000, or 16000 kg mass supported by spherical bodies. Normalization is performed per mass test.

The residual value $||\epsilon||_2$ is plotted for each mass test in Fig. 13.

6. CONCLUSIONS AND FUTURE WORK

This paper describes a performance analysis of several iterative methods for solving many-body dynamics problems formulated as complementarity problems. Using this formulation results in a constrained quadratic optimization problem that must be solved iteratively at each time step. Specifically, three new methods for solving this problem are described and compared with the commonly used Jacobi method through several numerical experiments. The results show several important aspects of the numerical solution. First, the ubiquitous Jacobi and Gauss-Seidel algorithms used in the contact dynamics community converge very slowly; see also [20]. Second, the scalability of Jacobi when applied to larger problems appears to be much worse than that of GPMINRES, SPG-FB, and Kučera, the three new algorithms investigated here. We note that one iteration of GPMINRES is more computationally expensive than one Jacobi sweep, yet, as illustrated by results in Table I, the former still is faster for large systems given the lower number

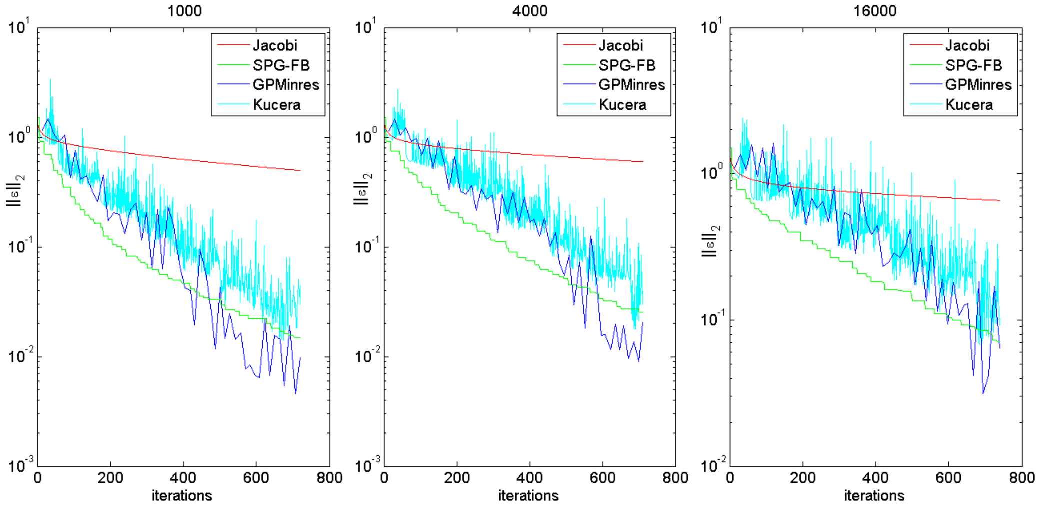


Figure 13. Residual values for Test Problem 3 ($n_c = 4295$), with 1000 kg, 4000 kg, or 16000 kg mass supported by spherical bodies.

of iterations required to achieve the same level of accuracy. Furthermore, preconditioning does not seem to provide a net benefit in the GPMINRES method. Using preconditioning in the solution of the subproblem allowed for fewer MINRES iterations to be used, but the computation of the preconditioning factors was prohibitively expensive and provided no significant benefit.

GPMINRES, SPG-FB, and Kučera have shown good performance for frictionless problems and can be mapped to leverage the parallel computing power of commodity graphics processing unit cards, as has already been done for the Jacobi method. Ongoing work will develop and test similar methods for use on problems with friction. Specifically, the cylindrical constraints associated with the Tresca friction model will be used in a fixed point iteration to lead to the conic constraints associated with the Coulomb friction model [54].

ACKNOWLEDGEMENTS

Financial support for Heyn and Negrut was provided in part by the National Science Foundation Award CMMI0840442. Financial support for Tasora was provided in part by the Italian Ministry of Education under the PRIN grant 2007Z7K4ZB. Anitescu was supported by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357.

REFERENCES

1. Pfeiffer F, Glocker C. *Multibody Dynamics with Unilateral Contacts*. John Wiley, 1996.
2. Negrut D, Tasora A, Anitescu M, Mazhar H, Heyn T, Pazouki A. Solving large multi-body dynamics problems on the GPU. *GPU Gems Vol. 4* 2011; :269–280.
3. Madsen J, Pechdimaljian N, Negrut D. Penalty versus complementarity-based frictional contact of rigid bodies: A CPU time comparison. *Technical Report TR-2007-06*, Simulation-Based Engineering Lab, University of Wisconsin, Madison 2007.
4. Cundall P. Formulation of a three-dimensional distinct element model–Part I. A scheme to detect and represent contacts in a system composed of many polyhedral blocks. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts* 1988; **25**(3):107–116.
5. Mazhar H. Parallel Multi-Body Dynamics on Graphics Processing Unit (GPU) Cards. M.S. thesis, Department of Mechanical Engineering, University of Wisconsin–Madison, http://sbel.wisc.edu/documents/HammadMazharMSthesisFinal.pdf 2012.
6. SBEL. Multibody Dynamics Simulation Movies, University of Wisconsin-Madison 2012. http://sbel.wisc.edu/Animations.
7. Song P, Pang JS, Kumar V. A semi-implicit time-stepping model for frictional compliant contact problems. *International Journal of Numerical Methods in Engineering* 2004; **60**(13):267–279.

DOI: 10.1002/nme

8. Moreau JJ. Standard inelastic shocks and the dynamics of unilateral constraints. *Unilateral Problems in Structural Analysis*, Piero GD, Macieri F (eds.), CISM Courses and Lectures no. 288, Springer–Verlag: New York, 1983; 173–221.
9. Lotstedt P. Mechanical systems of rigid bodies subject to unilateral constraints. *SIAM Journal of Applied Mathematics* 1982; **42**(2):281–296.
10. Marques MDPM. *Differential Inclusions in Nonsmooth Mechanical Problems: Shocks and Dry Friction*, *Progress in Nonlinear Differential Equations and Their Applications*, vol. 9. Birkhäuser Verlag: Basel, 1993.
11. Baraff D. Issues in computing contact forces for non-penetrating rigid bodies. *Algorithmica* 1993; **10**:292–352.
12. Pang JS, Trinkle JC. Complementarity formulations and existence of solutions of dynamic multi-rigid-body contact problems with Coulomb friction. *Mathematical Programming* 1996; **73**(2):199–226.
13. Stewart DE, Trinkle JC. An implicit time-stepping scheme for rigid-body dynamics with inelastic collisions and Coulomb friction. *International Journal for Numerical Methods in Engineering* 1996; **39**:2673–2691.
14. Anitescu M, Potra FA. Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlinear Dynamics* 1997; **14**:231–247.
15. Stewart DE. Rigid-body dynamics with friction and impact. *SIAM Review* 2000; **42(1)**:3–39.
16. Cottle RW, Dantzig GB. Complementary pivot theory of mathematical programming. *Linear Algebra and Its Applications* 1968; **1**:103–125.
17. Baraff D. Fast contact force computation for nonpenetrating rigid bodies. *Computer Graphics (Proceedings of SIGGRAPH)*, 1994; 23–34.
18. Anitescu M. Optimization-based simulation of nonsmooth rigid multibody dynamics. *Mathematical Programming* 2006; **105**(1):113–143, doi:http://dx.doi.org/10.1007/s10107-005-0590-7.
19. Anitescu M, Tasora A. An iterative approach for cone complementarity problems for nonsmooth dynamics. *Computational Optimization and Applications* 2010; **47**(2):207–235, doi:10.1007/s10589-008-9223-4.
20. Heyn T, Anitescu M, Tasora A, Negrut D. A Comparison of Several Solvers for Bound Constraint Quadratic Programming within the Context of Frictionless Multibody Dynamics. *Technical Report TR-2012-02*, Simulation-Based Engineering Laboratory, University of Wisconsin-Madison 2012.
21. Haug EJ. *Computer-Aided Kinematics and Dynamics of Mechanical Systems Volume-I*. Prentice-Hall: Englewood Cliffs, New Jersey, 1989.
22. Pang JS, Stewart DE. Differential variational inequalities. *Mathematical Programming* 2008; **113**:1–80.
23. Stewart DE. Convergence of a time-stepping scheme for rigid body dynamics and resolution of Painleve's problems. *Archive Rational Mechanics and Analysis* 1998; **145(3)**:215–260.
24. Tasora A. High performance complementarity solver for non-smooth dynamics. *Proceedings of the ECCOMAS Multibody Dynamics Conference*, Bottasso CL, Masarati P, Trainelli L (eds.), Milan, Italy, 2007.
25. Anitescu M, Hart GD. A constraint-stabilized time-stepping approach for rigid multibodydynamics with joints, contact and friction. *International Journal for Numerical Methods in Engineering* 2004; **60(14)**:2335–2371.
26. Anitescu M, Hart GD. A fixed-point iteration approach for multibody dynamics with contact and friction. *Mathematical Programming, Series B* 2004; **101(1)**:3–32.
27. Tasora A, Anitescu M. A convex complementarity approach for simulating large granular flows. *Journal of Computational and Nonlinear Dynamics* 2010; **5**(3):1–10, doi:10.1115/1.4001371.
28. Tasora A, Anitescu M. A matrix-free cone complementarity approach for solving large-scale, nonsmooth, rigid body dynamics. *Computer Methods in Applied Mechanics and Engineering* 2011; **200**(5-8):439–453, doi: doi:10.1016/j.cma.2010.06.030.
29. Moreau JJ, Jean M. Numerical treatment of contact and friction: The contact dynamics method. *Proceedings of the Third Biennial Joint Conference on Engineering Systems and Analysis*, Montpellier, France, 1996; 201–208.
30. Anitescu M, Potra FA. Time-stepping schemes for stiff multi-rigid-body dynamics with contact and friction. *International Journal for Numerical Methods in Engineering* 2002; **55(7)**:753–784.
31. Glocker C, Pfeiffer F. An LCP-approach for multibody systems with planar friction. *Proceedings of the CMIS 92 Contact Mechanics Int. Symposium*, Lausanne, Switzerland, 2006; 13–20.
32. Preclik TM, Iglberger K, Rde U. Iterative rigid multibody dynamics. *Proceeding of Multibody Dynamics ECCOMAS Thematic Conference*, 2009.
33. Shojaaee Z, Shaebani MR, Brendel L, Trk J, Wolf DE. An adaptive hierarchical domain decomposition method for parallel contact dynamics simulations of granular materials. *Journal of Computational Physics* 2012; **231**(2):612 – 628, doi:10.1016/j.jcp.2011.09.024. URL `http://www.sciencedirect.com/science/article/pii/S0021999111005675`.
34. Delannay R, Louge M, Richard P, Taberlet N, Valance A. Towards a theoretical picture of dense granular flows down inclines. *Nature Materials* 2007; **6**(2):99–108.
35. Preclik TM, Rde U. Solution existence and non-uniqueness of coulomb friction. *Technical Report 4*, Friedrich-Alexander-University Erlangen-Nurnberg, Institut Fur Informatik, Nurnberg, Germany 2011.
36. Tasora A. Chrono::Engine, An Open Source Physics–Based Dynamics Simulation Engine 2006. Available online at www.chronoengine.info.
37. Moré JJ, Toraldo G. On the solution of large quadratic programming problems with bound constraints. *SIAM Journal on Optimization* 1991; **1**(1):93–113, doi:10.1137/0801008. URL `http://link.aip.org/link/?SJE/1/93/1`.
38. Benson SJ, McInnes LC, Moré JJ. A case study in the performance and scalability of optimization algorithms. *ACM Trans. Math. Softw.* Sep 2001; **27**(3):361–376, doi:10.1145/502800.502805. URL `http://doi.acm.org/10.1145/502800.502805`.
39. Burke J, Moré JJ. Exposing constraints. *SIAM Journal on Optimization* 1994; **4**:573–595.
40. Saad Y. *Iterative methods for sparse linear systems*. Society for Industrial Mathematics, 2003.
41. Barzilai J, Borwein JM. Two-point step size gradient methods. *IMA Journal of Numerical Analysis* 1988; **8**(1):141–148, doi:10.1093/imanum/8.1.141. URL `http://imajna.oxfordjournals.org/content/8/1/141.abstract`.

42. Raydan M. On the Barzilai and Borwein choice of steplength for the gradient method. *IMA Journal of Numerical Analysis* 1993; **13**(3):321–326, doi:10.1093/imanum/13.3.321. URL `http://imajna.oxfordjournals.org/content/13/3/321.abstract`.
43. Raydan M. The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM Journal on Optimization* 1997; **7**(1):26–33. URL `http://link.aip.org/link/SJOPE8/v7/i1/p26/s1&Agg=doi`.
44. Birgin EG, Martínez JM, Raydan M. Nonmonotone spectral projected gradient methods on convex sets. *SIAM J. on Optimization* August 1999; **10**:1196–1211, doi:http://dx.doi.org/10.1137/S1052623497330963. URL `http://dx.doi.org/10.1137/S1052623497330963`.
45. Grippo L, Lampariello F, Lucidi S. A nonmonotone line search technique for newton's method. *SIAM J. Numer. Anal.* August 1986; **23**:707–716, doi:10.1137/0723046. URL `http://dl.acm.org/citation.cfm?id=13167.13169`.
46. Bello L, Raydan M. Preconditioned spectral projected gradient method on convex sets. *Journal of Computational Mathematics* 2005; **23**:225–232.
47. Luengo F, Raydan M, Glunt W, Hayden T. Preconditioned spectral gradient method. *Numerical Algorithms* 2002; **30**:241–258. URL `http://dx.doi.org/10.1023/A:1020181927999`, 10.1023/A:1020181927999.
48. Fletcher R. On the Barzilai-Borwein method. *Optimization and Control with Applications*, *Applied Optimization*, vol. 96, Qi L, Teo K, Yang X (eds.). Springer US, 2005; 235–256, doi:10.1007/0-387-24255-4_10. URL `http://dx.doi.org/10.1007/0-387-24255-4_10`.
49. Nocedal J, Wright SJ. *Numerical Optimization*, vol. 39. Springer, 1999.
50. Kucera R. Convergence rate of an optimization algorithm for minimizing quadratic functions with separable convex constraints. *SIAM Journal on Optimization* 2008; **19(2)**:846–862.
51. Tasora A, Negrut D, Anitescu M. Large-scale parallel multi-body dynamics with frictional contact on the Graphical Processing Unit. *Journal of Multi-body Dynamics* 2008; **222**(4):315–326.
52. Tasora A, Negrut D, Anitescu M. GPU-Based Parallel Computing for the Simulation of Complex Multibody Systems with Unilateral and Bilateral Constraints: An Overview. *Multibody Dynamics*, *Computational Methods in Applied Sciences*, vol. 23, Arczewski K, Blajer W, Fraczek J, Wojtyra M (eds.). Springer Netherlands, 2011; 283–307. URL `http://dx.doi.org/10.1007/978-90-481-9971-6_14`, 10.1007/978-90-481-9971-6_14.
53. Negrut D, Tasora A, Mazhar H, Heyn T, Hahn P. Leveraging parallel computing in multibody dynamics. *Multibody System Dynamics* 2012; **27**:95–117. URL `http://dx.doi.org/10.1007/s11044-011-9262-y`, 10.1007/s11044-011-9262-y.
54. Dostál Z, Kozubek T. An optimal algorithm and superrelaxation for minimization of a quadratic function subject to separable convex constraints with applications. *Mathematical Programming* 2012; **42**:1–26.