

A nonsmooth nonlinear conjugate gradient method for interactive contact force problems

Morten Silcowitz-Hansen · Sarah Niebe ·
Kenny Erleben

Published online: 17 April 2010
© Springer-Verlag 2010

Abstract Interactive rigid body simulation is important for robot simulation and virtual design. A vital part of the simulation is the computation of contact forces. This paper addresses the contact force problem, as used in interactive simulation. The contact force problem can be formulated in the form of a nonlinear complementarity problem (NCP), which can be solved using an iterative splitting method, such as the projected Gauss–Seidel (PGS) method. We present a novel method for solving the NCP problem by applying a Fletcher–Reeves type nonlinear nonsmooth conjugate gradient (NNCG) type method. We analyze and present experimental convergence behavior and properties of the new method. Our results show that the NNCG method has at least the same convergence rate as PGS, and in many cases better.

Keywords Contact force computation · Rigid body simulation · Nonsmooth conjugate gradients

1 Introduction

Most open source software for interactive real-time rigid body simulation use the widespread Projected Gauss–Seidel (PGS) method for solving the contact force problem, exam-

ples are Bullet [6] and Open Dynamics Engine [21]. The technology of such physics engines forms the basis in most robot simulators today such as Gazebo and Webots [7, 12]. However, the PGS method is not always satisfactory, it suffers from two problems: linear convergence rate [5, 8, 17] and inaccurate friction forces in stacks [11]. Linear convergence results in viscous motion at contacts and loss of high frequency effects. The viscous appearance results in a time delay in contact responses and reduces plausibility [16]. In this work, we devise a novel method based on a nonlinear conjugate gradient method. We hypothesize that a better convergence rate would allow one to do more iterations per frame in the visualization. Furthermore, we speculate that more iterations will imply better accuracy, thereby improving fidelity. However, to assess such an improvement from animation results would require a user study, which is not tractable in the scope of this work. This is therefore left for future work. In the following, we will present the NCP contact force formulation, derive the PGS method for this formulation, and proceed to adopt the PGS method into a novel NNCG method.

1.1 Previous work

Rigid body simulation is dominated by two major paradigms, impulse based methods [9, 10, 14], and constraint based methods [1, 22, 23]. The constraint based approach is usually formulated as complementarity problems at velocity level, and uses simplified friction modeling for interactive simulation [8]. Alternatives to complementarity formulations are based on kinetic energy [13] and motion space [18]. As mentioned in the previous section, the focus of this work will be the NCP model as described in [8].

M. Silcowitz-Hansen · S. Niebe (✉) · K. Erleben
Department of Computer Science, University of Copenhagen,
Universitetsparken 1, 2100 Copenhagen, Denmark
e-mail: niebe@diku.dk

M. Silcowitz-Hansen
e-mail: mosh@diku.dk

K. Erleben
e-mail: kenny@diku.dk

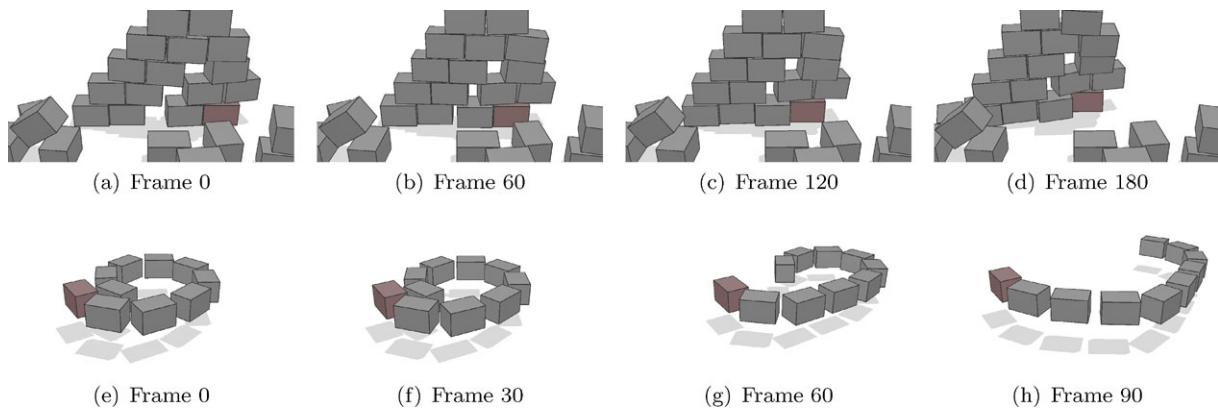


Fig. 1 Two video sequences of the NNCG method in action. The box that appears in a reddish tone is being interactively manipulated by the user. Both sequences were simulated using 50 NNCG iterations per frame. Both configurations are challenging in the context of interactive

simulation. Stacking requires accurate friction forces to remain stable, especially during interaction from the user. The long chain like construction creates a heavily coupled problem that causes trouble for the PGS solver; see Fig. 5(i)

2 The nonlinear complementarity problem formulation

The frictional contact force problem can be stated as a linear complementarity problem (LCP) [23]. However, a different formulation is used in interactive physical simulations; we will derive this formulation. Without loss of generality, we will only consider a single contact point. The focus of this paper is on the contact force model, so the time stepping scheme and matrix layouts are based on the velocity-based formulation in [8]. We have the Newton–Euler equations,

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

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

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

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

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

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

$$y < 0 \Rightarrow \lambda_t = \mu\lambda_n, \tag{4a}$$

$$y > 0 \Rightarrow \lambda_t = -\mu\lambda_n, \tag{4b}$$

$$y = 0 \Rightarrow -\mu\lambda_n \leq \lambda_t \leq \mu\lambda_n. \tag{4c}$$

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

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

where

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

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

$$\mathbf{y}^+ - \mathbf{y}^- = \mathbf{A}\lambda + \mathbf{b}, \tag{7a}$$

$$\mathbf{y}^+ \geq 0, \tag{7b}$$

$$\mathbf{y}^- \geq 0, \tag{7c}$$

$$u(\lambda) - \lambda \geq 0, \tag{7d}$$

$$\lambda - l(\lambda) \geq 0, \tag{7e}$$

$$(\mathbf{y}^+)^T (\lambda - l(\lambda)) = 0, \tag{7f}$$

$$(\mathbf{y}^-)^T (u(\lambda) - \lambda) = 0, \tag{7g}$$

$$(\mathbf{y}^+)^T (\mathbf{y}^-) = 0, \tag{7h}$$

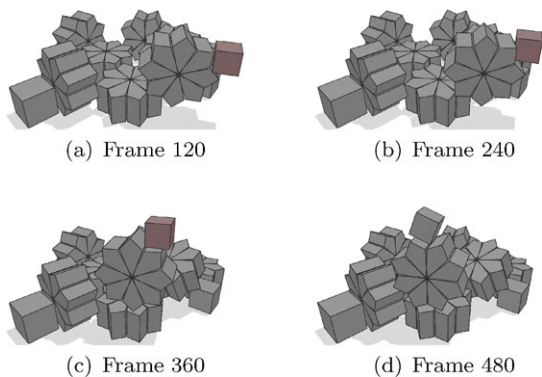


Fig. 2 The NNCG method in action. Similarly to the description found in Fig. 1, the reddish appearing box is interactively moved around by the user. 50 NNCG iterations were used per frame. Notice that the stability of the simulation allows the user to place one gear on top of another, by applying an indirect force through the small box

where

$$l(\lambda) = [l_n(\lambda) \quad \mathbf{l}_t(\lambda)^T]^T,$$

$$u(\lambda) = [u_n(\lambda) \quad \mathbf{u}_t(\lambda)^T]^T.$$

The advantage of the NCP formulation is a much lower memory footprint than for the LCP formulation. The disadvantage is solving the friction problem as two decoupled one-dimensional Coulomb friction models. As mentioned in the Introduction, numerous applications use the derived NCP formulation in practice. The motivation for this choice is presumably its relative simplicity and direct connection to the easily implemented PGS method. However, to our best knowledge, there exists no theorems for solution existence. Furthermore, it is well known that there exists simple configuration examples, which can be handled only with great difficulty by the NCP/PGS approach [20]. Finally, many physical phenomena such as rolling and anisotropic friction is not handled by the NCP model. However, the widespread usage of the NCP model, as well as the lack of a practically tractable alternative for interactive animation, the NCP model still appears as a reasonable choice of model.

3 The projected Gauss–Seidel method

The following is a derivation of the PGS method for solving the frictional contact force problem, stated as the NCP (7). The result of this derivation is a well-known result, and is as such of little novel contribution. However, we feel it is good instructional practice to deliver a sound derivation of the PGS method, which is neglected elsewhere in literature. A similar derivation is found in [17]. Using a minimum map reformulation, the i th component of (7) can be written

as

$$(\mathbf{A}\lambda + \mathbf{b})_i = \mathbf{y}_i^+ - \mathbf{y}_i^-, \tag{8a}$$

$$\min(\lambda_i - l_i, \mathbf{y}_i^+) = 0, \tag{8b}$$

$$\min(u_i - \lambda_i, \mathbf{y}_i^-) = 0, \tag{8c}$$

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

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

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

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

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

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

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

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

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

$$(\lambda^{k+1} - \mathbf{B}\lambda^{k+1} - \mathbf{c}^k)_i < l_i \implies \lambda_i^{k+1} = l_i, \tag{13a}$$

$$(\lambda^{k+1} - \mathbf{B}\lambda^{k+1} - \mathbf{c}^k)_i > u_i \implies \lambda_i^{k+1} = u_i, \tag{13b}$$

$$l_i \leq (\lambda^{k+1} - \mathbf{B}\lambda^{k+1} - \mathbf{c}^k)_i \leq u_i$$

$$\implies \lambda_i^{k+1} = (\lambda^{k+1} - \mathbf{B}\lambda^{k+1} - \mathbf{c}^k)_i. \tag{13c}$$

Case (13c) reduces to,

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

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

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

Thus, our iterative splitting method becomes

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

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

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

is the $(k + 1)$ th iterate obtained by projecting the vector λ' onto the box given by \mathbf{I}^k and \mathbf{u}^k . Using the splitting $\mathbf{B} = \mathbf{D} + \mathbf{L}$ and $\mathbf{C} = -\mathbf{U}$ results in the PGS method. The PGS method (17) can be efficiently implemented as a forward loop over all components and a component wise projection. To our knowledge, no convergence theorems exist for (16) in the case of variable bounds $l(\lambda)$ and $u(\lambda)$. However, for fixed constant bounds, the formulation can be algebraically reduced to that of a LCP formulation [3]. In general, LCP formulations can be shown to have linear convergence rate and unique solutions, when A is symmetric positive definite [5]. However, the \mathbf{A} matrix equivalent of our frictional contact model is positive symmetric semidefinite and uniqueness is no longer guaranteed, but existence of solutions is [5]. Experiments suggesting linear convergence rate of the PGS method for the NCP formulation can be found in [8, 17].

4 A non-smooth nonlinear conjugate gradient method

Experience from previous work [17, 20] indicates that the PGS method possesses desirable properties in terms of robustness and versatility in solving NCP posed contact force problems. It is therefore relevant to try and capture the robustness of the PGS method, by incorporating the PGS method into a new method that allows for an improved convergence rate. The PGS iteration can be written in generic form as

$$\lambda^{k+1} = \min \left(\underbrace{\mathbf{u}(\lambda^k)}_{\mathbf{T}_U \lambda^k + \mathbf{t}_U}, \max \left(\underbrace{\mathbf{l}(\lambda^k)}_{\mathbf{T}_L \lambda^k + \mathbf{t}_L}, \underbrace{-\mathbf{(D+L)^{-1}(U\lambda^k + b)}}_{\mathbf{T}\lambda^k + \mathbf{t}} \right) \right), \tag{18}$$

where the lower and upper bound functions $\mathbf{l}, \mathbf{u} : \mathbb{R}^n \mapsto \mathbb{R}^n$ are affine functions. The \mathbf{T}_L and \mathbf{T}_U matrices express the linear relations between tangential friction forces and their associated normal forces. The \mathbf{t}_L and \mathbf{t}_U vectors can be used to express fixed bound constraints, such as a normal force constraint. Thus, the PGS iteration can be perceived as a selector function on three affine functions. Assuming convergence sequence $\lambda^k \rightarrow \lambda^*$ for $k \rightarrow \infty$ the solution of PGS can be written as the fixed point formulation,

$$\lambda^* = \min \left(\underbrace{\mathbf{T}_U \lambda^* + \mathbf{t}_U, \max(\mathbf{T}_L \lambda^* + \mathbf{t}_L, \mathbf{T}\lambda^* + \mathbf{t})}_{\simeq \mathbf{H}\lambda^* + \mathbf{h}} \right). \tag{19}$$

The right-hand side of (19) can conceptually be considered as the evaluation of an affine function, $\mathbf{H}\lambda^* + \mathbf{h}$. This is always true if the active set of constraints was known in advance. Therefore, we have

$$0 = (\mathbf{H} - \mathbf{I})\lambda^* + \mathbf{h}. \tag{20}$$

Observe, explicit assembly is not needed for any of the matrices, instead one can rely on the PGS method to implicitly evaluate the residual of any given iterate, $\mathbf{r}^k = (\mathbf{H} - \mathbf{I})\lambda^k + \mathbf{h}$. Thus, if we write one iteration of the standard PGS method as

$$\lambda^{k+1} = \mathbf{PGS}(\lambda^k) \tag{21}$$

then $\mathbf{r}^k = \lambda^{k+1} - \lambda^k = \mathbf{PGS}(\lambda^k) - \lambda^k$. This can be interpreted as the gradient of a nonsmooth nonlinear quadratic function $f(\lambda^k) \equiv \frac{1}{2} \|\mathbf{r}^k\|^2$. By observing that $\nabla f(\lambda^k) = -\mathbf{r}^k$ and the solution we are seeking for corresponds to the local minimizer of f , it seems appropriate to use a Fletcher–Reeves nonlinear conjugate gradient method to search for the local minimizer of f [15]. In each step, one performs the update,

$$\lambda^{k+1} = \lambda^k + \tau^k \mathbf{p}^k, \tag{22}$$

where \mathbf{p}^k is the search direction and τ^k can be found using a line search method. Next, a new search direction is computed by

$$\beta^{k+1} = \frac{\|\nabla f^{k+1}\|^2}{\|\nabla f^k\|^2}, \tag{23a}$$

$$\mathbf{p}^{k+1} = \beta^{k+1} \mathbf{p}^k - \nabla f^{k+1}. \tag{23b}$$

To remain in an interactive context, we avoid performing line search on $f(\lambda^{k+1})$. Instead, we use the full step length $\tau = 1$, and restart the method whenever $\|\nabla f^{k+1}\|^2 > \|\nabla f^k\|^2$. This approach is motivated by our experiments, which show that nothing is gained from combining the NNCG method with a line search step; see Fig. 3. Further, we take into account that when computing $\mathbf{PGS}(\lambda^k)$, it is practical to do so in-place, meaning that a PGS step is taken implicitly on λ^k . Therefore, the update of λ^k is done separately in two places in each iteration. We state the full algorithm as shown below:

- 1 : $\lambda^1 \leftarrow \mathbf{PGS}(\lambda^0)$
- 2 : $\nabla f^0 \leftarrow -(\lambda^1 - \lambda^0)$
- 3 : $\mathbf{p}^0 \leftarrow -\nabla f^0$
- 4 : $k \leftarrow 1$
- 5 : **while not converged do**
- 6 : $\lambda^{k+1} \leftarrow \mathbf{PGS}(\lambda^k)$
- 7 : $\nabla f^k \leftarrow -(\lambda^{k+1} - \lambda^k)$
- 8 : $\beta^k \leftarrow \|\nabla f^k\|^2 / \|\nabla f^{k-1}\|^2$
- 9 : **if** $\beta^k > 1$
- 10 : $\mathbf{p}^k \leftarrow 0$ // restart
- 11 : **else**
- 12 : $\lambda^{k+1} \leftarrow \lambda^{k+1} + \beta^k \mathbf{p}^{k-1}$
- 13 : $\mathbf{p}^k \leftarrow \beta^k \mathbf{p}^{k-1} - \nabla f^k$
- 14 : $k \leftarrow k + 1$

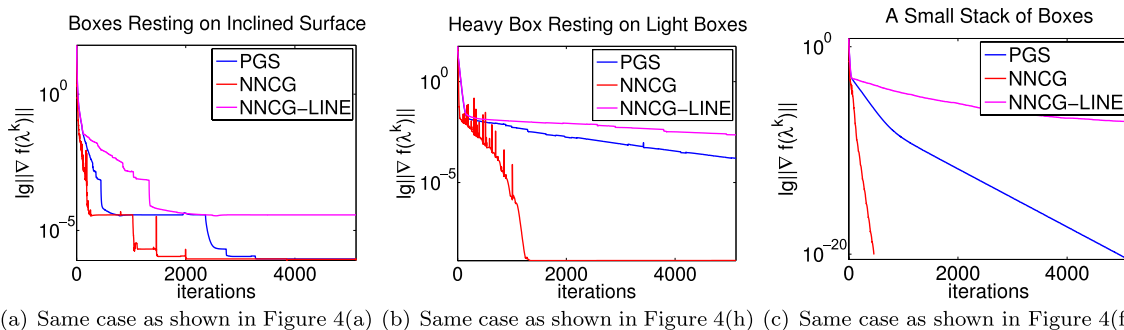


Fig. 3 Test results: The NNCG method combined with line search. The line search is a backtracking line search, using the Armijo-condition $f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) \leq f(\mathbf{x}_k) + c_1 \alpha_k \nabla f_k^T \mathbf{p}_k$ with $c_1 = 0.01$. The

NNCG method with line search does not improve convergence rate, when compared to either the PGS method or the NNCG method. Notice that no restarts are evident in the NNCG-LINE plot

As a termination criteria one may use $\|\nabla f^k\| < \varepsilon$ for some user specified value ε . A Polak–Ribiere variant may be used in place of the Fletcher–Reeves method. Further, one may wish to use different restarting criteria for the nonlinear conjugate gradient method [15].

Clearly f is a nonsmooth function. However, it is by definition also a B-differentiable function [19] and as such it belongs to the class of semi-smooth functions where one can extend the concept of the gradient to a general definition. We use $\nabla f(\lambda) = -(\mathbf{PGS}(\lambda) - \lambda)$ which is an element of the generalized Jacobian according to Clarke’s definition [4].

5 Results

The test configurations are shown in Fig. 4. These configurations were chosen to test the solver’s ability to handle traditionally difficult problems, such as stacking and resting structures that rely on static friction forces. The test involving gears in Fig. 4(e) and (d) was chosen to produce NCP problems that consist of numerous redundant contact points; see Table 1 and contacts that induce a high amount of angular velocity on bodies. The test results are shown in Fig. 5 and in Fig. 6. In Fig. 5, only the iterations 0 through 150 are shown, to clarify the behavior of the NNCG method within an interactive context. The implementation was done in Java, and the tests were performed on a Lenovo T61 2.0 Gz machine. The error measure used in the plots is the merit function $\|\nabla f^k\|^2$. This is a natural choice, as it is the function minimized by the method. Other measures could be used, such as the Fischer function [20], but might be misleading because a descending step on $\|\nabla f^k\|^2$ may not always be a descending step on other merit functions. The iteration count is done in the number of PGS iterations, and so a NNCG iteration is slightly more expensive than a PGS iteration, but this is

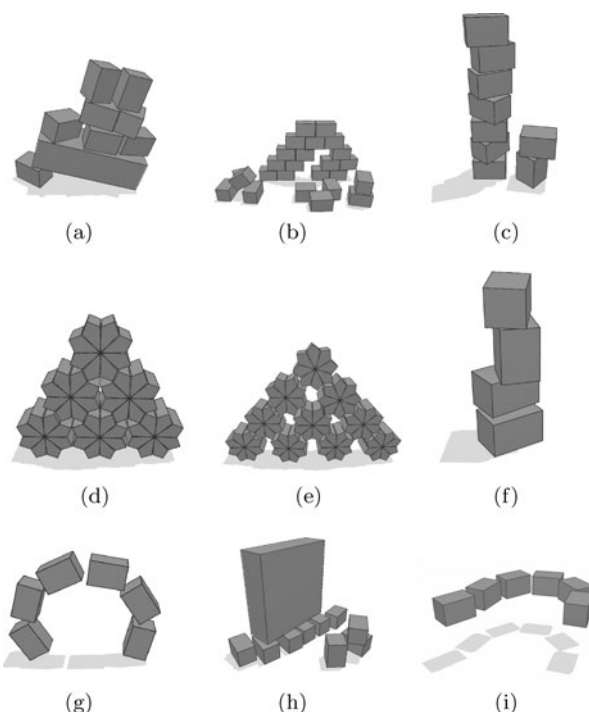


Fig. 4 Illustrated test cases used for the nonlinear conjugate gradient method: (a) Boxes resting on an inclined surface resulting in static friction forces; (b) A large configuration of boxes stacked in a friction inducing manor; (c) A large stack of boxes of equal mass; (d) A small pyramid of gears; (e) A medium-scale pyramid of gears; (f) A small stack of boxes of equal mass; (g) A standing arched snake, composed of boxes and hinge-joints; (h) A heavy box placed on top of lighter boxes with mass ratio $\frac{1}{100}$; (i) A snake suspended in the air by a terminating fixed link. Notice the resting friction dependence in (a) and (b)

highly implementation dependent. It should be noted that the work done in one NNCG iteration, aside from that of the PGS iteration, can be computed in body space, which is often of significantly lower dimension. The overall result is that the NNCG method has equivalent or better convergence properties, both in interactive and noninteractive set-

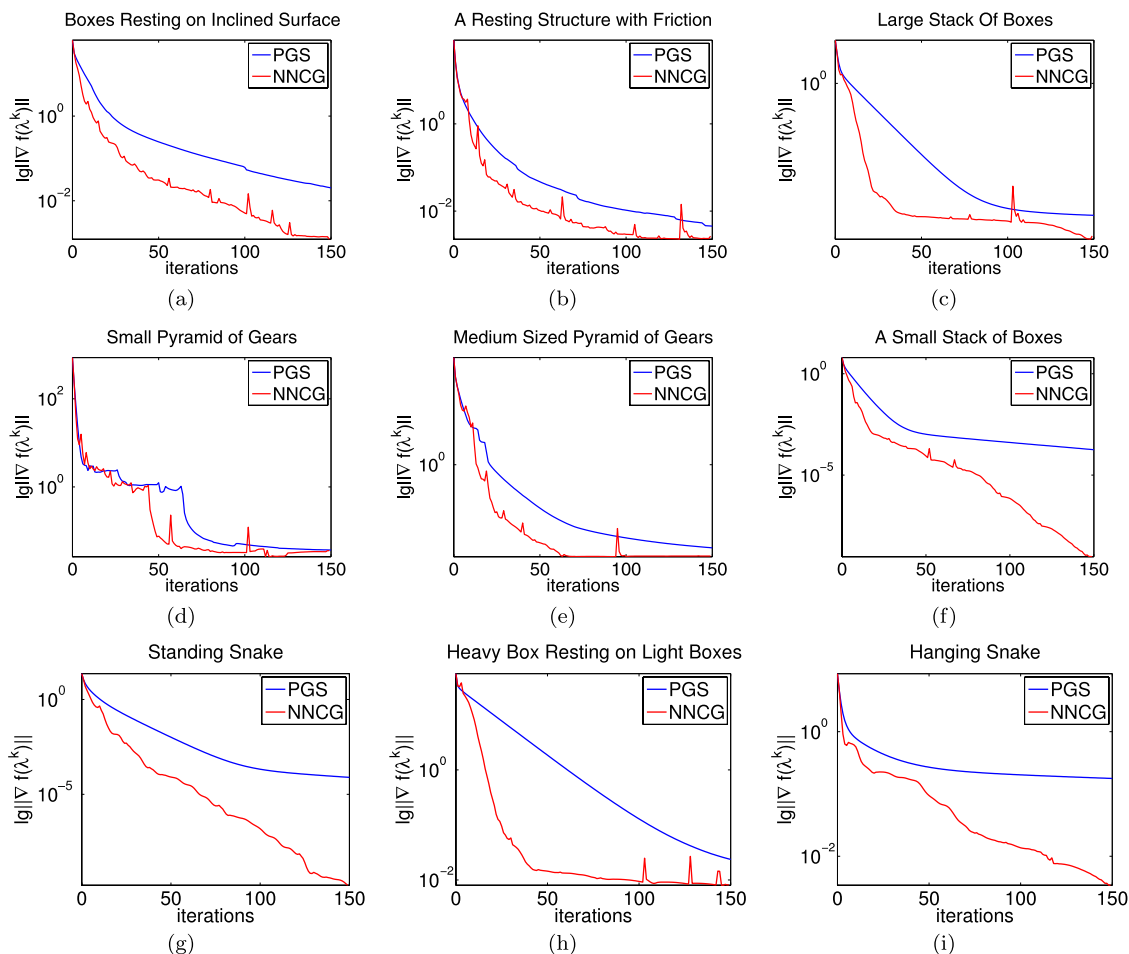


Fig. 5 Test results: The amount of iterations is kept within 150 to emphasize the effect of the methods in an interactive setting. The small spikes indicate that the method has started to diverge, thus triggering a restart

Table 1 Configuration properties of the test cases illustrated in Fig. 4. Test configurations may contain more bodies than appears in the illustrations, because invisible features such as floor and walls also count as physical bodies. The rightmost column contains the number of eigenvalues significantly above zero present in the **A**-matrix

Configuration	Bodies	Constraints	# A eigen. > ϵ
Figure 4(a)	14	171	54 (31.5%)
Figure 4(b)	30	519	157 (30.2%)
Figure 4(c)	14	156	54 (34.6%)
Figure 4(d)	11	216	36 (16.6%)
Figure 4(e)	15	198	60 (30.3%)
Figure 4(f)	9	51	24 (47.0%)
Figure 4(h)	15	168	62 (36.9%)
Figure 4(g)	11	42	36 (85.7%)
Figure 4(i)	11	30	30 (100.0%)

tings. Some test cases even show drastically improved convergence, such as in Figs. 5(f) and 6(f). In the noninteractive cases of Fig. 6, the NNCG method is observed to con-

verge to a lower error measure than the PGS method. The convergence rate of the NNCG method is linear or super-linear, although a slight tendency towards locally quadratic convergence can be observed, as in Fig. 6(h). The restarts, which are triggered by a rising merit function, are evident as spikes in all the convergence plots. However, the overall convergence does not seem to be harmed significantly, as the method quickly recovers to the same level of error as before the restart. When disregarding restarts, the convergence rate shown in Fig. 6(h) is clearly quadratic. The number of restarts was high in the test case corresponding to Fig. 6(a), and it should be noted that this test case has the highest amount of contact points; see Table 1. A high amount of spikes is also seen in Fig. 6(h). However, this test case shows a clear quadratic convergence rate tendency, despite the large amount of restarts. In the noninteractive plots in Fig. 6, the NNCG method has a tendency to stagnate at some level of error, then suddenly drop to a lower level. This behavior is also true for the PGS method, but the NNCG method does so in fewer iterations, and thus reaches a lower error level faster.

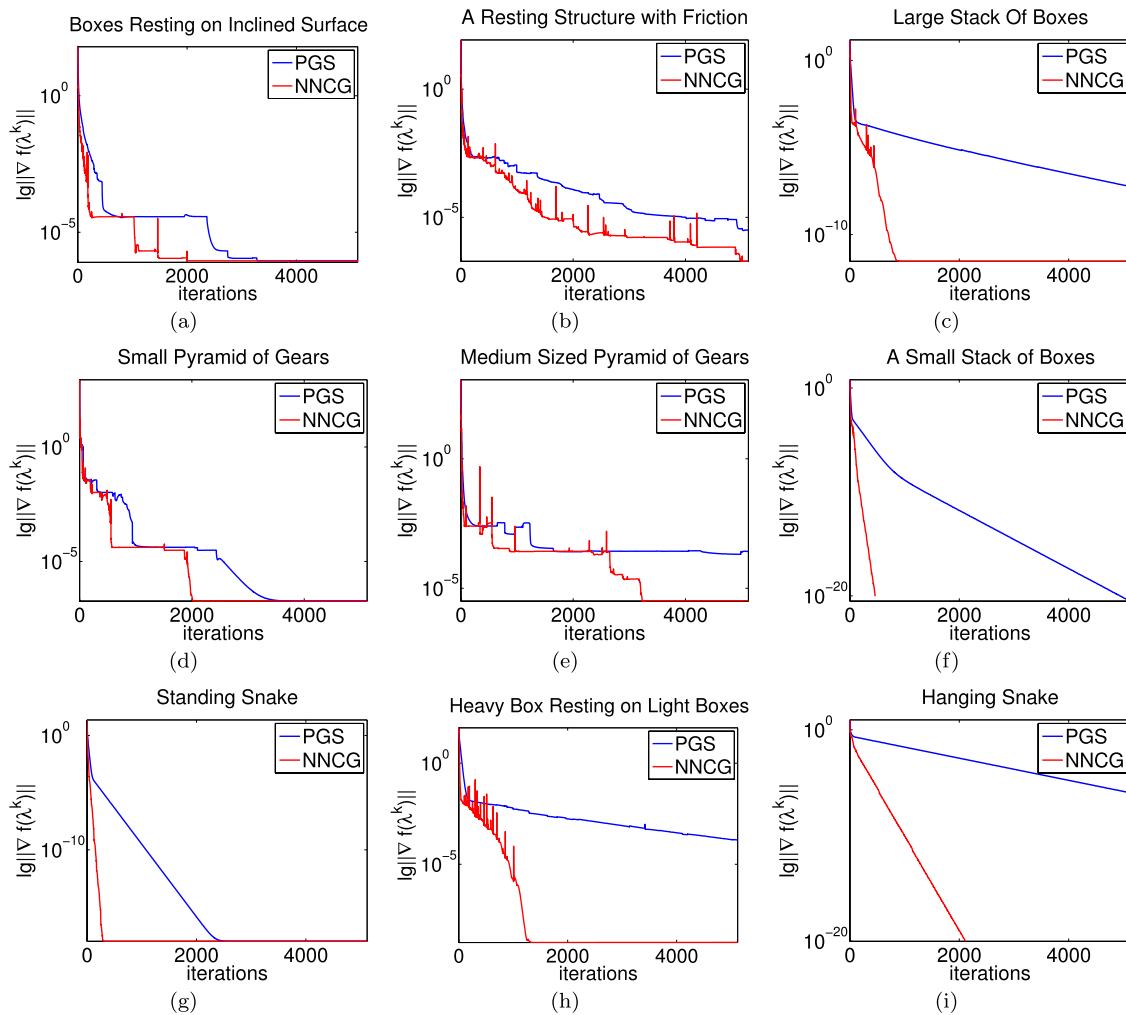


Fig. 6 Test results: Up to 5,000 iterations of both the PGS method and the NNCG method. The NNCG method clearly converges faster, and often to a higher accuracy than that of the PGS method. Notice the superior rate of convergence in (f)–(i)

6 Conclusion

We have presented a novel method for solving the contact force problem, when stated as a NCP formulation. The method is a nonsmooth nonlinear Fletcher–Reeves type optimization (NNCG) method, applied to a function whose gradient is obtainable by performing a single Projected Gauss–Seidel (PGS) iteration. The accuracy obtained by the novel method was equivalent or better than that of the PGS method. The best observed convergence rate was super-linear, with slight local tendency toward quadratic convergence. In general, the final accuracy obtained by the NNCG method was better or equivalent to PGS in all experiments performed. To our best knowledge, there exists no other published work on solving NCP contact force problems, that demonstrates better convergence results than the

NNCG method within interactive frame rates. The NNCG method showed the best results when applied to smaller configurations, with an obvious structure, such as small or medium sized stacks. This property was also observed by the authors in [20], where a Newton type method was applied to the NCP contact force problem. We speculate that the inability to improve convergence on larger and more complex contact force problems, comes from the drastically increased over-determinacy when the number of contacts grows. This over-determinacy causes the dependence between solution variables to become increasingly complex, possibly preventing the NNCG method from exploiting the structure present in contact force problems. Future work may therefore focus on methods for handling this problem, possible preconditioning, or multigrid techniques.

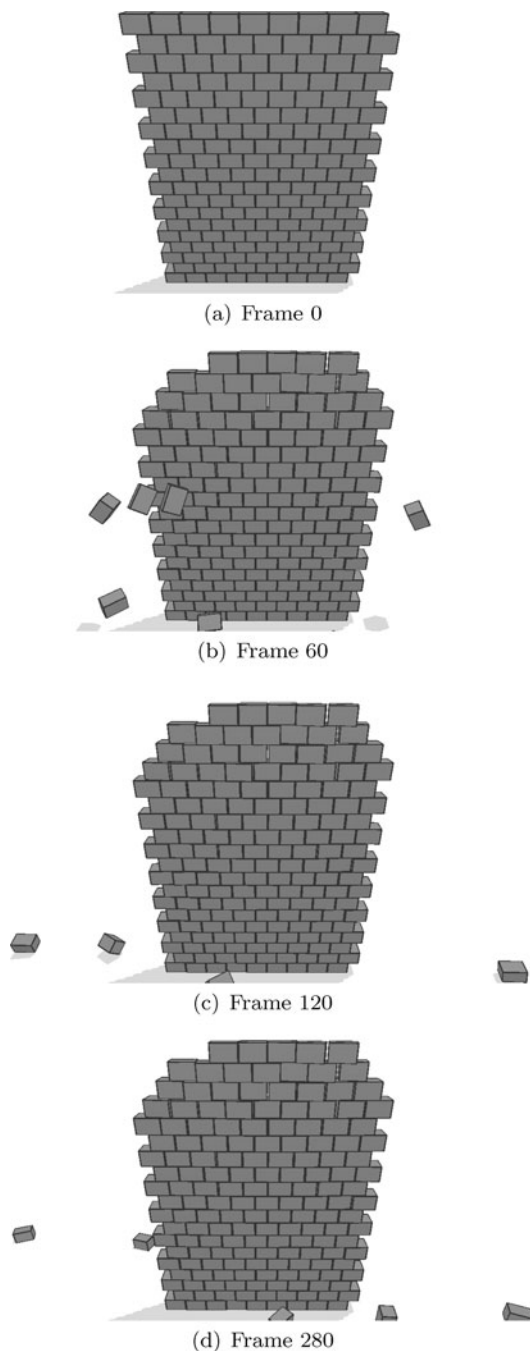


Fig. 7 The NNCG method applied to a large wall structure. The wall is initially falling onto the floor, and thus a shock effect makes the top level bricks bounce off. The animation was done using 500 NNCG iterations per frame. The configuration contains 174 bodies, and has on average 4083 constraints. The obtained error at final resting frame was $\|\nabla f_k\| = 4.62 \times 10^{-3}$. This large scale animation is not within interactive frame rates, but it demonstrates the robustness of the NNCG method when applied to a large scale configuration

References

- Anitescu, M., Potra, F.A.: Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlinear Dyn. An International Journal of Nonlinear Dynamics and Chaos in Engineering Systems* (1997)
- Baraff, D.: Fast contact force computation for nonpenetrating rigid bodies. In: SIGGRAPH '94: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques (1994)
- Billups, S.C.: Algorithms for complementarity problems and generalized equations. PhD thesis, University of Wisconsin at Madison (1995)
- Clarke, F.: *Optimization and Nonsmooth Analysis*. Society for Industrial Mathematics (1990)
- Cottle, R., Pang, J.-S., Stone, R.E.: *The Linear Complementarity Problem*. Academic Press, San Diego (1992)
- Coumans, E.: The bullet physics library. <http://www.continuousphysics.com> (2005)
- Cyberbotics: Webots 6. <http://www.cyberbotics.com/products/webots/> (2009)
- Erleben, K.: Velocity-based shock propagation for multibody dynamics animation. *ACM Trans. Graph.* **26**(2) (2007)
- Guendelman, E., Bridson, R., Fedkiw, R.: Nonconvex rigid bodies with stacking. *ACM Trans. Graph.* (2003)
- Hahn, J.K.: Realistic animation of rigid bodies. In: SIGGRAPH '88: Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques (1988)
- Kaufman, D.M., Sueda, S., James, D.L., Pai, D.K.: Staggered projections for frictional contact in multibody systems. *ACM Trans. Graph.* **27**(5) (2008)
- Koeng, N., Polo, J.: Gazebo, 3d multiple robot simulator with dynamics. <http://playerstage.sourceforge.net/index.php?src=gazebo> (2009)
- Milenkovic, V.J., Schmidl, H.: A fast impulsive contact suite for rigid body simulation. *IEEE Trans. Vis. Comput. Graph.* **10**(2) (2004)
- Mirtich, B.V.: Impulse-based dynamic simulation of rigid body systems. PhD thesis, University of California, Berkeley (1996)
- Nocedal, J., Wright, S.J.: *Numerical Optimization*. Springer Series in Operations Research. Springer, New York (1999)
- O'Sullivan, C., Dingliana, J., Giang, T., Kaiser, M.K.: Evaluating the visual fidelity of physically based animations. *ACM Trans. Graph.* **22**(3) (2003)
- Poulsen, M., Niebe, S., Erleben, K.: Heuristic convergence rate improvements of the projected Gauss–Seidel method for frictional contact problems. In: Proceedings of WSCG (2010)
- Redon, S., Kheddar, A., Coquillart, S.: Gauss least constraints principle and rigid body simulations. In: Proceedings of IEEE International Conference on Robotics and Automation (2003)
- Scholtes, S.: Introduction to piecewise differential equations. Prepring No. 53, May (1994)
- Silcowitz, M., Niebe, S., Erleben, K.: Nonsmooth Newton Method for Fischer Function Reformulation of Contact Force Problems for Interactive Rigid Body Simulation. In: VRIPHYS 09: Sixth Workshop in Virtual Reality Interactions and Physical Simulations, pp. 105–114. Eurographics Association (2009)
- Smith, R.: Open dynamics engine. <http://www.ode.org> (2000)
- Stewart, D.E.: Rigid-body dynamics with friction and impact. *SIAM Rev.* (2000)
- Stewart, D.E., Trinkle, J.C.: An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *Int. J. Numer. Meth. Eng.* (1996)



Morten Silcowitz-Hansen holds a B.Sc in Computer Science from the University of Copenhagen (2008) and is currently a M.Sc student at the department of Computer Science, University of Copenhagen. He is a computer graphics enthusiast, involved in both research and open-source projects.



Sarah Niebe received her M.Sc in Computer Science at the University of Copenhagen, Denmark (2009). Her main research focus is physical simulation and rigid body dynamics, as well as optimization methods. Since graduating, Sarah has worked as a research assistant at the eScience Center, University of Copenhagen.



Kenny Erleben After his completion of master in Computer Science, Erleben was employed as full time researcher in 3DFacto A/S for a period of 10 months. In 2001 Erleben started on his Ph.D. studies. During 2004 Erleben stayed 3 months at the Department of Mathematics, University of Iowa. He received his Ph.D. degree in the beginning of 2005 and finally late 2005 Erleben was employed as an Assistant Professor at the Department of Computer Science, University of Copenhagen. Erleben has been chairman of the OpenTissue project since 2007, which he was co-founder of in late 2001. In 2009 Erleben was employed as an Associate Professor.