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

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

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

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

ABSTRACT

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

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

1 SLOW CONVERGENCE RATES

Most open source software for interactive real time rigid body simulation uses the Projected Gauss–Seidel (PGS) method for computing contact forces. This includes the two most popular open source simulators Bullet and Open Dynamics Engine. However, the PGS method is not always satisfactory as it suffers from two major problems: a linear convergence rate [4] and inaccurate friction forces in stacks [9]. The linear convergence rate results in viscous motion at contacts, as well as loss of high frequency effects. The viscous appearance results in a delayed contact response which reduces plausibility [15]. Improving convergence might lead to increased animation quality and higher fidelity. The prospect of improving a state-of-the-art method – the PGS method – has motivated this study of heuristics, aimed at improving convergence.

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

2 PREVIOUS WORK

Rigid body simulation was introduced to the graphics community in the late 1980's [8, 12], using penalty based and impulse based approaches to describe physical interactions. Penalty based simulation is not easily adopted to different simulations. Mirtich [11] presented an extended and improved impulse based formulation, however stacking was a problem and it suffered from creeping. These problems has since been rectified [7]. Constraint based simulation [2] has received much attention as an alternative to penalty based and impulse based simulation. Constraint based simulation can be divided into two groups: maximal coordinate and minimal coordinate methods [6]. The focus of this paper is maximal coordinate methods, which are dominated by complementarity formulations. Alternatives to complementarity formulations are based on kinetic energy [10] and motion space [16]. However, the former solves a more general problem and is not attractive for performance reasons, the latter does not include frictional forces.

Complementarity formulations are either acceleration based formulations [20] or velocity based formulations [19]. Acceleration based formulations can not handle collisions [1], in addition they suffer from indeterminacy and inconsistency [18]. Velocity based formulations suffer from none of these problems, for this reason we use a velocity based formulation for this paper. The approach we present here is based on a reformulation of the frictional problem as a nonlinear complementarity problem. This results in a slightly inaccurate model with relatively few variables to solve for. This makes it advantageous in interactive simulations from a performance viewpoint.

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

WSCG'2010, February 1 – February 4, 2010
Plzen, Czech Republic.
Copyright UNION Agency – Science Press

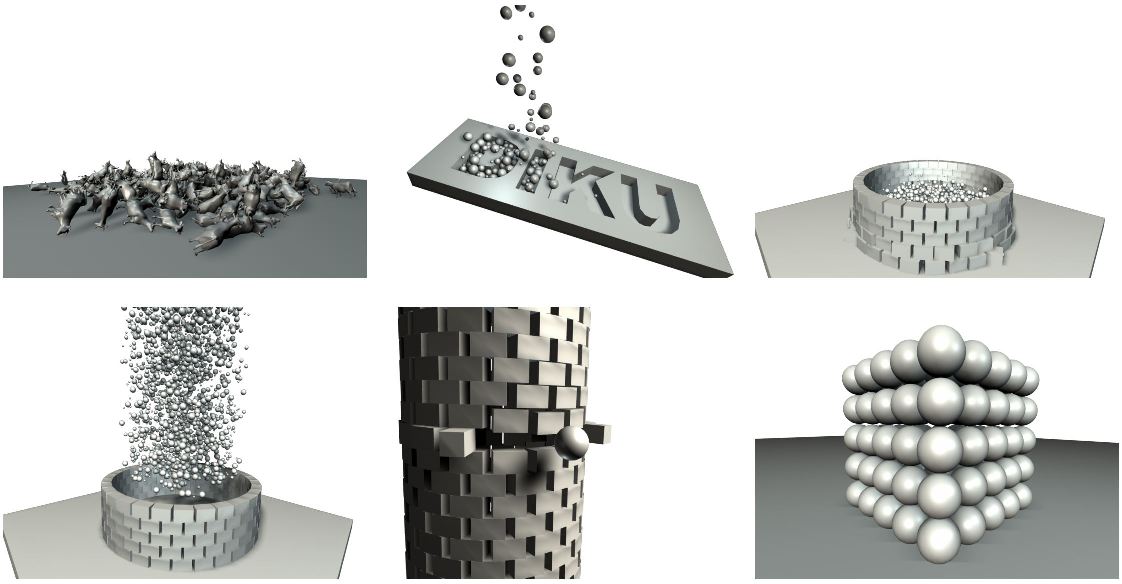


Figure 1: Renderings of selected setups from the test data sets used to examine the performance achieved by using different heuristics for the projected Gauss–Seidel method. The setups varies from 100-10000 interacting rigid bodies in varying degrees of structured configurations.

for interactive rigid body simulation has been presented in the Computer Graphics literature, nor has any studies on its convergence behavior or rates been published.

3 THE NONLINEAR COMPLEMENTARITY PROBLEM FORMULATION

The frictional contact force problem can be stated as a linear complementarity problem (LCP) [19]. However, a slightly different formulation is used in interactive physical simulations, we will derive this formulation. Without loss of generality, we will only consider a single contact point. The focus of this paper is on the contact force model, so the time stepping scheme and matrix layouts are based on the velocity-based formulation in [5]. We have the Newton–Euler equations,

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

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

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

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

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

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

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

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

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

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

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

where

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

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

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

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

4 THE PROJECTED GAUSS–SEIDEL METHOD

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

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

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

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

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

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

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

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

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

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

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

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

Case (13c) reduces to,

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

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

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

Thus, our iterative splitting method becomes,

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

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

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

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

$$
\begin{aligned}
\mathbf{M} &= \mathbf{D} && \wedge && \mathbf{N} = -\mathbf{D} - \mathbf{U}, && \text{(18a)}\\
\mathbf{M} &= \mathbf{D} + \mathbf{L} && \wedge && \mathbf{N} = -\mathbf{U}, && \text{(18b)}\\
\mathbf{M} &= \mathbf{D} + \omega\mathbf{L} && \wedge && \mathbf{N} = (1-\omega)\mathbf{D} - \omega\mathbf{U}, && \text{(18c)}
\end{aligned}
$$

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

```
for $k = 1$ to $k_{\max}$ do
  for $i = 1$ to $n$ do
    $\lambda_i' \leftarrow \frac{-\sum_{j=1}^{i-1} A_{i,j}\lambda_j - \sum_{j=i+1}^{n} A_{i,j}\lambda_j - b_i}{A_{i,i}}$
    $\lambda_i \leftarrow \min(u_i, \max(l_i, \lambda_i'))$
    for all $j$ dependent on $i$ do
      $(l_j, u_j) \leftarrow$ update$(\lambda_i)$
    next $j$
  next $i$
next $k$
```

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

5 HEURISTICS FOR CONVERGENCE IMPROVEMENTS

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

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

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

We have researched numerous heuristics, measuring improvements in convergence, speedup and accuracy. In an effort not to obfuscate the interesting and important results, we will only present the most interesting or promising heuristics. We have divided the heuristics into three groups, physics-based, geometry-based and splitting-based. To represent physics-based heuristics, we have chosen an impulse propagation permutation (IPP) heuristic. This is interesting as it is based on the impulse-based formulation of rigid body dynamics. From the group of geometry-based heuristics, a coordinate permutation (CP) heuristic is examined. Inspired by [9], the splitting-based heuristics are represented by both an extreme staggered permutation (ESP) heuristic as well as a greedy staggered permutation (GSP) heuristic.

Impulse Propagation Permutations: The PGS method solves the contact problem sequentially, which is numerically similar to sequential impulse propagation methods [11]. However, the basis of the PGS method is a simultaneous contact model, while a sequential model is used for impulse propagation methods. Impulse propagation methods attempt to model the propagation of collision impulses through rigid bodies, so perhaps a similar physical principle is beneficial with the PGS method. The idea is to apply a permutation of the problem, mimicking impulse propagation. We base the permutation on a sorting of the relative contact velocities **y**. The sorting is performed as a preprocessing step, the same sorting is used throughout the PGS method. The permutation could be incrementally updated, possibly improving the convergence. This increases the time complexity of each Gauss–Seidel iteration by $\mathscr{O}(nlgn)$. Experience showed that this was to costly, so we only performed an ascending and decreasing pre-processing sorting. Results are presented for an ascending ordering permutation heuristic.

Coordinate Permutations: Consider computing contact impulses for a stack of boxes, this problem has a distinctly dominating up/down direction. If one solves for contacts from bottommost boxes before uppermost boxes, then the solution may propagate in a bottom-up fashion similar to [5]. This indicates there might be certain directions, where shocks can be propagated and vanish in a single sweep. We sort the contact points by their position along each coordinate axis, considering both ascending and descending ordering. Different choices for coordinate axes has also been examined. We also considered a symmetric blocked approach. Blocking was introduced to make sure normal impulses where solved prior to the dependent friction impulses. This set of heuristics are represented by a descending ordering by y coordinates.

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

As in [9], the full problem can also be split into two coupled subproblems, where one alternate between solving normal and friction impulses in each iteration. To realize the benefit of this extreme staggered approach, let us first split (3) into two coupled subproblems by partitioning the matrix-vector equation into two coupled equations according to the normal and friction sequences such that,

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

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

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

and

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

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

6 EXPERIMENTS AND RESULTS

We define a residual function from (10) as,

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

The natural merit function of (22) is,

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

which we use as an absolute error measure for the PGS method. We use Q-convergence measures for our analysis [13]. A comparison of convergence rates is done by visual inspections of logarithmic plots of $(iteration, \log(\Theta))$, observe Figure 2.

The test data has been generated using a number of setups from [14]. The setups uses the negative z-axis for the direction of gravity and the x-axis and y-axis span the horizontal plane. The only exception is the *diku*-setup, where the negative y-axis is the direction of gravity. For each setup, **A** has been examined by plotting nonzero elements and eigenvalues. Many setups have a spectral radius $\rho(\mathbf{M}^{-1}\mathbf{N}) \geq 1$, which is cause for concern when considering convergence proofs for the Gauss–Seidel method for linear equation systems. The **A**-matrices are positive semi definite, having a large ratio of zero valued eigenvalues, ranging from (50%-90%) – only the *diku*-setup has no zero valued eigenvalues.

To clarify the experimental results, the test data has been partitioned into equivalency classes [5]. For this paper, only one representative data set from each class is presented.

Non-structured: Few contacts are present, **J** is small.

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

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

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

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

Table 1: Sample setups and their complexity.

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

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

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

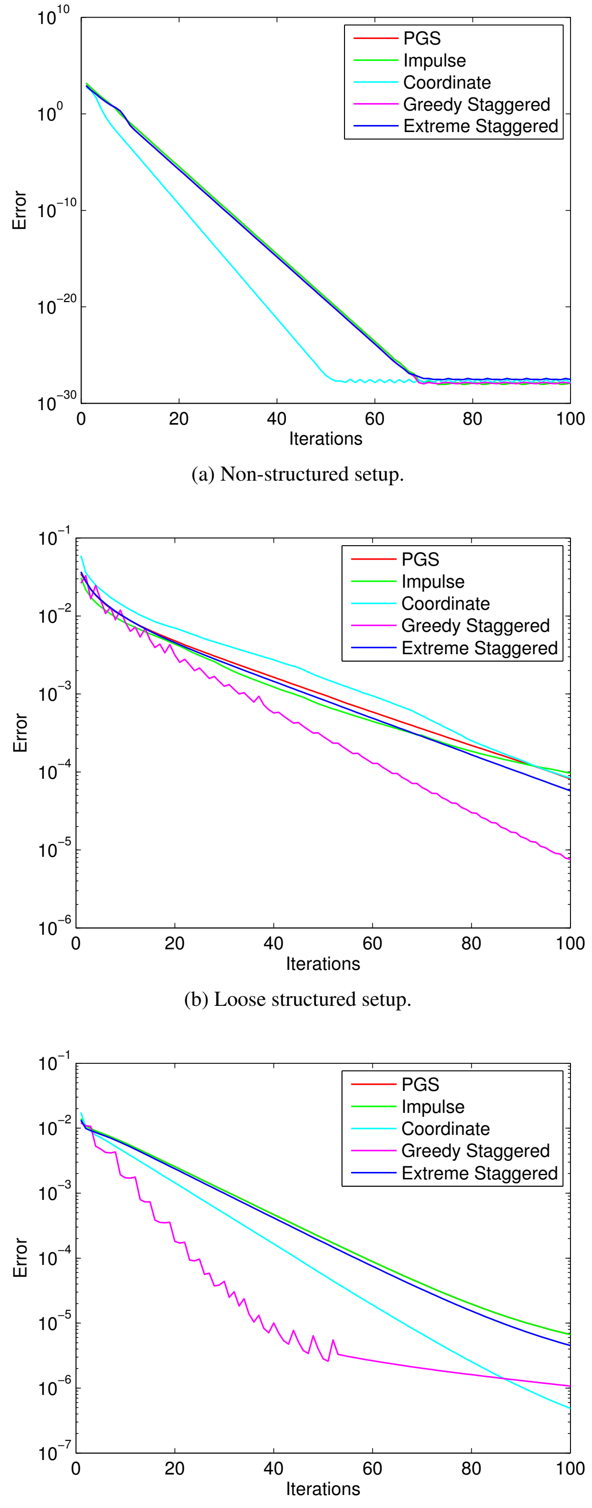
Coordinate Permutation: A number of CP heuristic variations have been investigated sorting by x, y, and z coordinates. The most interesting results arose when sorting descending by y coordinate, which is shown on Figure 2. This particular permutation performs well on dense and non-structured setups,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

Extreme Staggered Permutation: The ESP heuristic performs similar to the pure PGS method. However, it uses much less time per iteration than the pure PGS method, see Table 2. The speedup is more pronounced for the larger data sets than for the smaller data sets.

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

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

Greedy Staggered Permutation: As Figure 2 shows, the GSP heuristic performs at least as well as the pure PGS method, and often better. When changing from normal iterations to friction iterations (or vice versa), a short burst in the rate of convergence is experienced. This causes the jagged shape of the convergence plots. Another noticeable feature of the GSP heuristic is the speedup. As Table 3 shows, the GSP heuristic is roughly twice as fast as the pure PGS method.



(a) Non-structured setup.

(b) Loose structured setup.

(c) Dense structured setup.

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

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

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

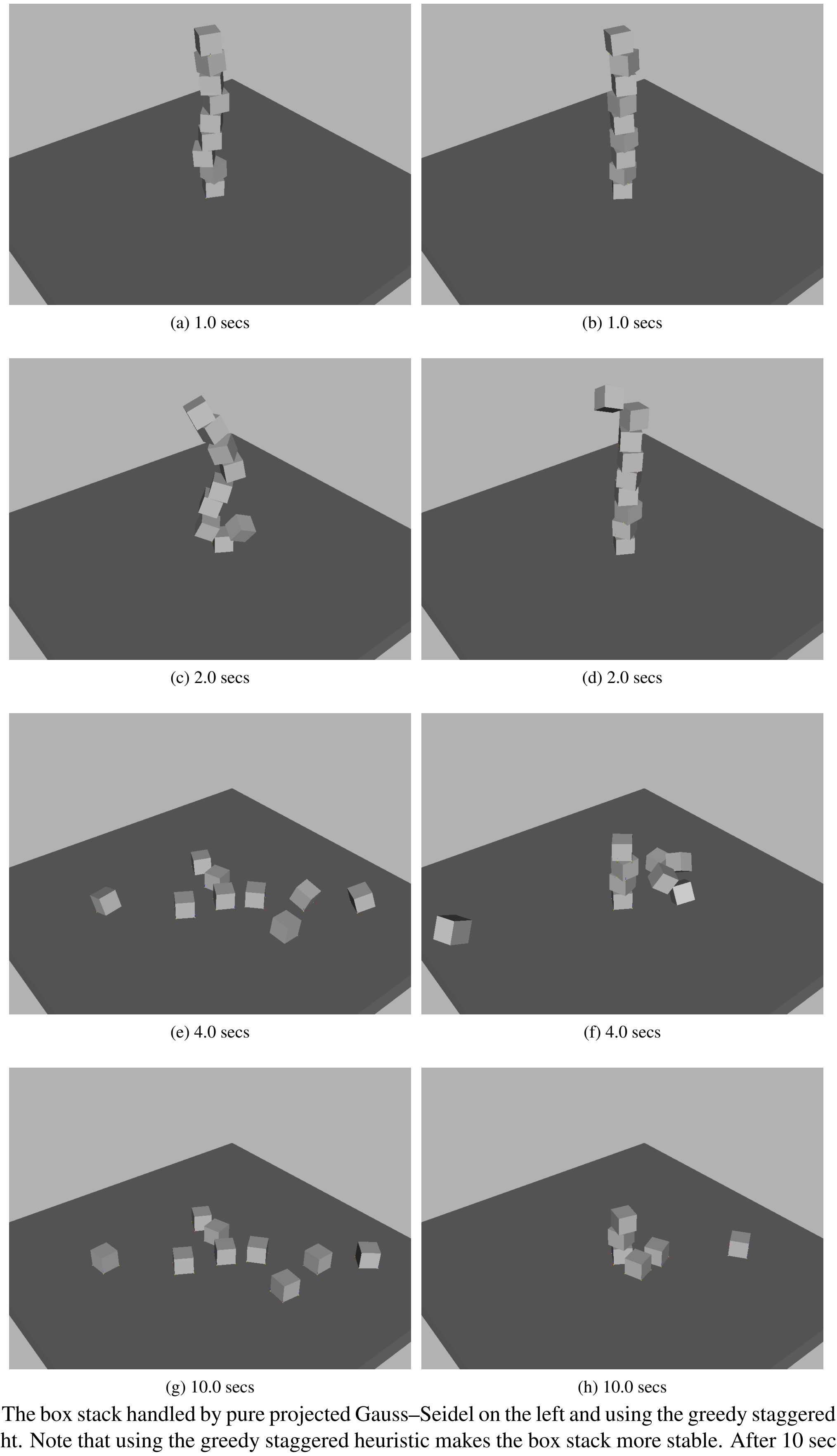
The GSP heuristic has been implemented in OpenTissue [14]. A comparison of the visual improvement over the PGS method, when used on the *box stack*-setup is shown in Figure 3. Note that the box stack is more stable when using the GSP heuristic, compared to using the pure PGS method.

7 CONCLUSION AND DISCUSSION

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

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

In our opinion, staggered permutations are the most promising heuristics. It is interesting to hypothesize on why this is so. Are their success the result of improved numerical behavior of the method when decoupling normal impulses from friction impulses? Normal impulses control the bounds of the friction impulses, whereas the friction impulses have an indirect, yet significant, effect on normal impulses through the dynamics of multiple contacts. Or is it because the friction model used in the nonlinear complementarity problem (NCP) formulation is too poor a model? Should future work focus on creating improved contact models for interactive simulation that has a better friction model component? Or should future numerical schemes be based on the splitting idea? The results by [17] suggest that convergence rate for the friction component of the NCP formulation is much slower than for the normal component, This indicates to us that the NCP formulation is a poor friction model. In [9] an accurate



(a) 1.0 secs

(b) 1.0 secs

(c) 2.0 secs

(d) 2.0 secs

(e) 4.0 secs

(f) 4.0 secs

(g) 10.0 secs

(h) 10.0 secs

Figure 3: The box stack handled by pure projected Gauss–Seidel on the left and using the greedy staggered heuristic on the right. Note that using the greedy staggered heuristic makes the box stack more stable. After 10 seconds, the bottom three boxes are still stacked when using the GSP heuristic.

linear complementarity problem (LCP) formulation is used to model friction, and they obtain nice results for friction. This again indicates that the current friction model widely used in interactive rigid body simulators is too poor. We will leave these questions open for future work on interactive rigid body simulation.

REFERENCES

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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