

The definitive version is available at <http://diglib.eg.org/> and <http://onlinelibrary.wiley.com/>.

Deformable Objects Collision Handling with Fast Convergence

Siwang Li Zherong Pan Jin Huang[†] Hujun Bao[†] Xiaogang Jin[†]

State Key Lab of CAD&CG, Zhejiang University

Abstract

We present a stable and efficient simulator for deformable objects with collisions and contacts. For stability, an optimization derived from the implicit time integrator is solved in each timestep under the inequality constraints coming from collisions. To achieve fast convergence, we extend the MPRGP based solver from handling box constraints only to handling general linear constraints and prove its convergence. This generalization introduces a cost of solving dense linear systems in each step, but these systems can be reduced into diagonal ones for efficiency without affecting the general stability via pruning redundant collisions. Our solver is an order of magnitude faster, especially for elastic objects under large deformation compared with iterative constraint anticipation method (ICA), a typical method for stability. The efficiency, robustness and stability are further verified by our results.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

Collision handling for deformable objects has been an active area of research in computer graphics. For rigid objects, collision response can be treated as a separate pass after time integration. However, this does not apply for deformable objects since their internal forces could be stiff, leading to stability issues when a time-split solver is used. A widely adopted solution here is to use the implicit penalty forces, i.e., solving an unconstrained optimization with terms to penalize the penetration depth. However, they again lead to issues of additional artificial parameters, inefficient time stepping or sticking.

A more stable, but usually less efficient strategy is to formulate collisions as a set of inequality constraints. This formulation when coupled with an implicit time-stepper leads to a QP problem:

$$\min_x \frac{1}{2} x^T A x - x^T b \quad \text{s.t. } Jx \geq c, \quad (1)$$

which robustly reproduces non-sticky behaviour at the end of each timestep. To solve such problem efficiently, active set [JN06] and Projected Gauss-Seidel (PGS) [Cot09, OTSG09] are two commonly used approaches for physics based modelling. However, active set approaches use direct solvers to

determine active constraints, and thus is limited to dynamic systems with moderate degree of freedoms (DOFs). PGS adopts iterative solver, but usually suffers from poor convergence, especially for models with large number of DOFs. Moreover, being based on the dual problem:

$$\min_{\lambda} \frac{1}{2} \lambda^T J A^{-1} J^T \lambda - \lambda^T (c - J A^{-1} b) \quad \text{s.t. } \lambda \geq 0, \quad (2)$$

a coupled PGS formulation is still costly because of the term $J A^{-1} J^T$.

To address the stability and efficiency issues simultaneously, we propose a novel algorithm to solve the above specific QP problem arising from deformable body contacts and collisions. Our method is enlightened by the recently proposed “Modified Proportioning with Reduced Gradient Projections” algorithm (MPRGP) [Dos09], an iterative QP solver using gradient projection for box constraints. We generalize it in several aspects to handle general linear constraints. Unfortunately, the performance of this basic extension is limited by the gradient projection subproblem, which involves solving dense linear systems. To tackle this issue, we propose to diagonalize them by decoupling the constraints via discarding redundant collisions. By virtue of the three key ingredients, implicit strategy, extended MPRGP and constraints decoupling, our method is not only stable and robust, but achieves an order of magnitude faster conver-

[†] Corresponding Author: {hj.bao,jin}@cad.zju.edu.cn

gence over ICA, a typical implicit method [OTSG09] with PGS solver, for models with hundreds of thousands of DOFs.

2. Related Works

Deformable Body Simulation In computer graphics, since the pioneering work of [TPBF87], much effort has been devoted to the simulation of deformable objects. The related techniques cover deformable objects modelling, collision detection and contact handling etc.. While model reduction are usually adopted to accelerate the simulation in real-time applications [PW89, BJ05, CK05, AKJ08], dynamic systems in full space are usually preferred for high quality animation, especially for simulating highly detailed deformation caused by collisions and contacts [ZSTB10, MZS*11]. This work focuses on resolving collisions between large objects with hundreds of thousands DOFs.

Collision Detection Collision detection for deformable objects is widely studied in computer graphics. Readers can refer to [TKZ*04] for more details. Collision resolution techniques can be roughly divided into discrete methods (DCD) and continuous methods (CCD) [BFA02, RKLMO4, ORC07, TCYMO8, HVTG08]. While being much slower, CCD methods are more robust than DCD methods because all collisions are guaranteed to be resolved before next timestep. In this work, we focus on efficient simulation of volumetric deformable bodies with moderate timestep size, where DCD method is an approximate choice.

Collision Handling For volumetric deformable bodies, the most widely used DCD methods are based on penalty force [BW98, CK02, HFL00, TMOT12] due to its simplicity. [BW98, CK02] resolves each collision locally by applying an impulse, which requires small timestep size for stability. The continuous force model proposed in [TMOT12] can alleviate the problem but the stability is not guaranteed. on the other hand, [HFL00] introduces a spring energy for each collision which can then be considered by a implicit time integrator. Although this method is much more stable, it results in sticky contacts. Similar problem also arises in shell modelling [HVTG08]. Most recently, [TOK14] used the idea of cubature [AKJ08] to accelerate the computation of the penalty forces. A time consuming training process is required in their approach, and more importantly, it is limited to skeleton driven objects.

On the other hand, our approach is based on implicit framework which is widely used for rigid bodies [ST96] and thus avoids the stability problems even for large timestep size. For rigid object animation, the dual formulation can be solved via LCP solvers [Cot09], and has gained popularity over the last decade [Bar94, RKC02, KEP05, Erl07]. It has also been adopted to solve collisions between deformable objects since [BW92]. However, solving LCP for deformable bodies is excessively expensive because of the large number of contacts, DOFs and non-trivial energy Hessian. There has been works trying to simplify deformable models by leveraging pre-computations [PPG04, DDKA06].

For the dynamic systems with small DOFs, active set methods [JN06] are exploited to solve the corresponding LCP problem [KSJP08]. Most recently, [OTSG09] proposed an iterative constraint anticipation method (ICA) for this problem. However, this method uses an approximate Hessian in their dual formulation leading to slow convergence. Instead, we model the collision handling problem as a QP problem in its primal variables and propose a novel solver which converges much faster than ICA as demonstrated in our results.

MPRGP Solver Our method is based on the MPRGP method [Dos09] which provides the advantage of fast convergence. It has previously been used for modelling fluid flow [GB13, NGL10], but the MPRGP method has a limitation that only box constraints can be applied. In this work, we extend this method to efficiently handle a subset of general linear constraints which is enough to handle collisions between deformable objects.

3. Background

3.1. Motion Equation

The equation of motion for an elastic object with n vertices is

$$M\ddot{x} + f(x, \dot{x}) = f_{ext}, \quad (3)$$

where $x \in R^{3n}$, $M \in R^{3n \times 3n}$, $f(x, \dot{x}) \in R^{3n}$ and $f_{ext} \in R^{3n}$ represent the shape of the object, the mass matrix, the internal (elastic and damping) forces, as well as the external forces respectively. By using an implicit integrator with timestep size h to discretize Eq. (3), we obtain

$$\frac{1}{h^2}M(x^{i+1} - 2x^i + x^{i-1}) + f\left(x^{i+1}, \frac{1}{h}(x^{i+1} - x^i)\right) = f_{ext}^i, \quad (4)$$

where x^k represents the shape of the object at time $t^k = kh$ for $k = 0, 1, 2, \dots$. Eq. (4) is usually a nonlinear system with respect to x^{i+1} , and can be solved by using Newton's method. In what follows, we use x instead of x^{i+1} for clarity. At each iteration of Newton's method, we need to solve a quadratic system:

$$\min_x \frac{1}{2}x^T Ax - b^T x, \quad (5)$$

where A is usually a symmetric positive definite matrix

$$A = \frac{1}{h^2}M + \left.\frac{\partial f}{\partial x}\right|_{x=x^i}, \quad (6)$$

and b is given by

$$b = f_{ext}^i - f(x^i) + \left(\left.\frac{\partial f}{\partial x}\right|_{x=x^i}\right)x^i + \frac{1}{h^2}M(2x^i - x^{i-1}). \quad (7)$$

It is a common strategy to limit the Newton's iterations within only a few iterations to reduce the simulation cost, and the results are usually still stable and plausible.

3.2. Non-penetration Constraints

Considering the collisions and contacts, the vertex positions x of a deformable object is limited to some complex feasi-

ble domain Ω , which can be locally sampled as a set of linear constraints using conventional collision detectors. Hereafter, we assume that x are under a set of collision constraints $\{C_i\}$, where each constraint C_i asks the point x_i penetrating triangle (x_j, x_k, x_l) with normal n_i to be moved into a collision free status:

$$n_i^T (x_i - w_j x_j - w_k x_k - w_l x_l) \geq 0, \quad (8)$$

where w_j, w_k, w_l are the barycentric coordinates of the contact point on the triangle, which is returned by a collision detector. After grouping all the constraints in a matrix format as $Jx \geq c$, at each Newton's step, the following constrained quadratic programming problem needs to be solved :

$$\min_x \frac{1}{2} x^T A x - b^T x \quad \text{s.t.} \quad Jx \geq c. \quad (9)$$

Our work focuses on solving this problem efficiently and robustly.

3.3. MPRGP

Our solver is based on the MPRGP algorithm, which is proposed for solving the following QP problem with box constraints:

$$\min_x \frac{1}{2} x^T A x - x^T b \quad \text{s.t.} \quad x \in \Omega_B, \quad (10)$$

where matrix A is symmetric positive definite, and $\Omega_B = \{x : x \geq L\}$ defines the box constraints, where vector L is the lower bound of x . There are several variants of the traditional MPRGP algorithm as introduced in [Dos09]. We adopt the monotonic MPRGP in our current implementation, but our extension can be directly applied to other variants.

The outline of the traditional monotonic MPRGP is shown in Algorithm 1. The constants $\bar{\alpha} \in (0, 2\|A\|_2^{-1}]$ and $\Gamma > 0$ are adjustable parameters. We always choose $\bar{\alpha} = 2\|A\|_2^{-1}$, and $\Gamma = 1$. At each iteration, MPRGP computes the free gradient $\phi(x)$, the chopped gradient $\beta(x)$, and the reduced free gradient $\tilde{\phi}_{\bar{\alpha}}(x)$:

$$\phi_i(x) = \begin{cases} g_i(x) & \text{for } i \in \mathcal{F}(x) \\ 0 & \text{for } i \in \mathcal{A}(x), \end{cases} \quad (11)$$

$$\beta_i(x) = \begin{cases} 0 & \text{for } i \in \mathcal{F}(x) \\ \min(g_i(x), 0) & \text{for } i \in \mathcal{A}(x), \end{cases} \quad (12)$$

$$\tilde{\phi}_{\bar{\alpha}}(x) = \frac{1}{\bar{\alpha}}(x - P_{\Omega_B}(x - \bar{\alpha}\phi(x))). \quad (13)$$

Here, $g(x) = Ax - b$ is the gradient of the objective function, $\mathcal{F}(x) = \{i : x_i > L_i\}$, $\mathcal{A}(x) = \{i : x_i = L_i\}$ are the free and active set respectively, and $P_{\Omega_B}(\cdot)$ is a projection operator with the entries defined by

$$[P_{\Omega_B}(x)]_i = \max(x_i, L_i), \quad (14)$$

which projects any vector into the feasible domain Ω_B . Fig. 1 gives an illustration of the free and chopped gradient. The free gradient is responsible for functional value minimization, keeping the active set fixed, while the chopped gradient is responsible for active set change.

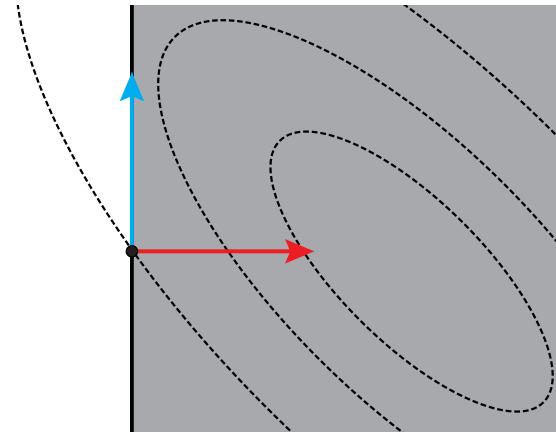


Figure 1: Illustration of the two types of modified gradients used in MPRGP algorithm on a 2D toy example. Here, the gray region is Ω_B , the dotted lines are iso-lines of objective function. The black dot is our current solution. To derive a better solution, we can either keep the active set fixed and go along the inverse free gradient (blue). Or we can leave the active set and go along the inverse chopped gradient (red).

Now with a definition of free and chopped gradient, we can present the MPRGP algorithm. It works by choosing between several types of steps: the CG step (Algorithm 2) keeps the active set fixed and always goes along the free gradient. This is just the conventional conjugate gradient method [JN06] projected into the active set. While the proportioning step is responsible for active set change by going along the chopped gradient. MPRGP algorithm determines the type of next step according to the expected function value decrease. However, if a CG step is chosen, it also runs the risk of leaving the feasible domain. In that case, MPRGP uses a third type of step, the expansion step, where a steepest descendant direction is taken with fixed step size and the solution is then pulled back to a closest feasible point in Ω_B by the projection operator. Finally, after a non-CG step is performed, the conjugate property of the previous CG series becomes violated, so the entire CG iterations need to be restarted.

4. MPRGP Extension

The traditional MPRGP solver introduced above only supports QP problems with box constraints, and can not be applied to solve the QP problem Eq. (1) that appears in collision handling. Although the dual formulation of Eq. (1), i.e. Eq. (2), can be solved by traditional MPRGP solver, it involves the term $JA^{-1}J^T$ and definitely leads to low performance for deformable bodies with large DOFs even without explicitly expressing it as a matrix.

In this section, we will extend the MPRGP solver to support the primal formulation of Eq. (1) with more general linear constraints. We found that there are three key operations

Algorithm 1: Monotonic MPRGP outline

Given a SPD matrix A , vectors b, L and initial value $x^0 \geq L$, choose $\Gamma > 0$, $\tilde{\alpha} \in (0, 2\|A\|_2^{-1}]$,
set $k = 0, p = \phi(x^0)$;
while not convergent **do**
 Step 1. {compute $g(x^k), \phi(x^k), \beta(x^k), \tilde{\phi}_{\tilde{\alpha}}(x^k)$ };
 if $\|\beta(x^k)\|^2 \leq \Gamma^2 \tilde{\phi}_{\tilde{\alpha}}(x^k)^T \phi(x^k)$ **then**
 Step 2. {CG steps};
 compute p, x^{k+1}, y using Algorithm 2;
 if not convergent and $y \notin \Omega_B$ **then**
 Step 3. {expansion step};
 $y = P_{\Omega_B}(x^k), x^{k+1} = P_{\Omega_B}(y - \tilde{\alpha}\phi(y)),$
 $g = Ax^{k+1} - b, p = \phi(x^{k+1}), k = k + 1$
 end
 else
 Step 4. {proportioning step};
 $\alpha_p = \beta^T(x^k)g(x^k)/\beta^T(x^k)A\beta(x^k),$
 $x^{k+1} = x^k - \alpha_p\beta(x^k), g = g - \alpha_pA\beta(x^k),$
 $p = \phi(x^{k+1}), k = k + 1;$
 end
 end
 return x^k ;

Algorithm 2: CG steps

$\alpha_{cg} = g^T p / p^T A p, y = x^k - \alpha_{cg} p$;
while not convergent and $f(P_{\Omega_B}(y)) \leq f(P_{\Omega_B}(x^k))$ and
 $\|\beta(x^k)\|^2 \leq \Gamma^2 \tilde{\phi}_{\tilde{\alpha}}(x^k)^T \phi(x^k)$ **do**
 $x^{k+1} = y, g = g - \alpha_{cg} A p,$
 $s = \phi(y)^T A p / p^T A p, p = \phi(y) - s p, k = k + 1,$
 $\alpha_{cg} = g^T p / p^T A p, y = x^k - \alpha_{cg} p$;
end
 return x^{k+1}, y, p ;

of the traditional MPRGP algorithm introduced above depending on the formulation of constraints: the free gradient Eq. (11), chopped gradient Eq. (12) and projection operator Eq. (14). We can adopt the outline of MPRGP shown in Algorithm 1, and extend it to support more general linear constraints in Eq. (1) by properly redefining these operations. In the following, we use \hat{J} and $\hat{\epsilon}$ to represent the active constraints set, i.e. $\hat{J}x = \hat{\epsilon}$, where \hat{J} is the sub-matrix of J , and $\hat{\epsilon}$ is the corresponding sub-vector of c .

4.1. Free Gradient

In MPRGP algorithm, the free gradient ϕ is supposed to be an increasing direction (thus its inverse would be a decreasing direction). What's more, it is required that, an infinitesimal move along the free gradient should not change the active set, which can be satisfied by enforcing $\hat{J}\phi = 0$. Thus, we define the free gradient ϕ as a projection of the gradient

g onto the null space of matrix \hat{J} :

$$\phi = \min_{\phi} \frac{1}{2} \|\phi - g\|_2^2, \quad \text{s.t. } \hat{J}\phi = 0. \quad (15)$$

After substituting $\phi = g + \hat{J}^T \hat{\lambda}_\phi$ into $\hat{J}\phi = 0$, the Lagrange multipliers $\hat{\lambda}_\phi$ can be solved as:

$$(\hat{J}\hat{J}^T)\hat{\lambda}_\phi = -\hat{J}g, \quad (16)$$

and then the free gradient can be obtained by $\phi = g + \hat{J}^T \hat{\lambda}_\phi$.

4.2. Chopped Gradient

The chopped gradient should be complement to the free gradient, and its inverse direction should be a feasible decreasing direction, i.e it should satisfy $-\hat{J}\beta \geq 0$. Thus we compute the chopped gradient $\beta(x)$ by solving

$$\beta = \min_{\beta} \frac{1}{2} \|\beta - g_\beta\|_2^2, \quad \text{s.t. } -\hat{J}\beta \geq 0, \quad (17)$$

where $g_\beta = g - \phi$. The LCP equivalence of the above problem is

$$0 \leq \hat{\lambda}_\beta \perp \hat{J}\hat{J}^T \hat{\lambda}_\beta \geq \hat{J}g_\beta, \quad (18)$$

which can be solved by using traditional Active Set method. Finally we can compute the chopped gradient as $\beta = g_\beta - \hat{J}^T \hat{\lambda}_\beta$.

4.3. Projection

In order to enforce the final solution to be feasible, we need to project any infeasible solution x into the feasible domain $\Omega_c = \{x : Jx \geq c\}$ in each iteration. We define the projection operation as

$$P_{\Omega_c}(x) = \min_y \frac{1}{2} \|y - x\|_2^2, \quad \text{s.t. } Jy \geq c, \quad (19)$$

which returns a feasible vector y that is closest to x . Similar to the way of solving β , we start from its LCP correspondence

$$0 \leq \lambda_P \perp JJ^T \lambda_P \geq c - Jx, \quad (20)$$

which again is solved by traditional Active Set method for λ_P . The projected point is then $y = x + J^T \lambda_P$.

4.4. Convergence

Our extension on the three key operations follows the idea of original MPRGP algorithm, so that we adopt the same stopping criteria: $\|\phi_i(x) + \beta_i(x)\| < \epsilon$ for both Algorithm 1 and Algorithm 3. Unfortunately, the convergence proof of the original MPRGP algorithm cannot be directly applied to our method because it heavily depends on the fact that the operator P_{Ω_c} is separable (see [Dos09]), which does not hold for the general linear constraints in our case. As a consequence, simply replacing the operators in Algorithm 1 cannot guarantee the convergence. However, we noticed that two desirable properties of Algorithm 1 still hold for our extended operators:

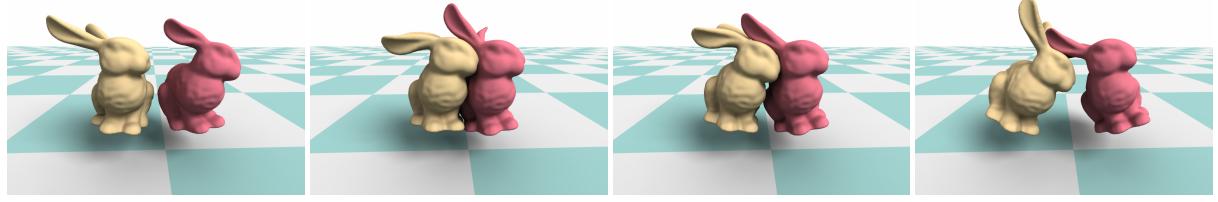


Figure 2: Collision decoupling is necessary especially when there are a large amount of interbody collisions. We observed no instability using simply random strategy to filter the constraints.

1. Function values would strictly reduce after either CG or proportioning step. This is because both ϕ and β are decreasing directions as shown in Appendix A.
2. The series $\{x^k\}$ generated by repeatedly applying the expansion step is convergent when $\alpha \in (0, 2\|A\|_2^{-1}]$. This is justified by Proposition 5.6 of [Dos09] because our projection operator maps any point to a closest point in Ω_B which is a convex set.

Based on these two facts, we can construct a convergent variant of Algorithm 1 by repeatedly applying the expansion step in an inner while loop until the function value is strictly decreasing. In that way, all three types of steps give strictly decreasing function values and the algorithm is thus convergent. One additional advantage of this technique is that the rate of convergence by the expansion step is linear to the spectral radius of A . Therefore, the inner while loop can take the advantage of an effective preconditioner.

In the original MPRGP algorithm, where Ω_B is a unilaterally bounded separable domain, x^{k+1} is guaranteed to be feasible after the proportioning step. But this is not the case for our general convex domain, therefore x^{k+1} need to be guarded against infeasibility by another while loop.

The while loops work as a final resort against divergence. However, these fail-safe strategies are actually not necessary when some collision constrains are excluded from consideration, see Section 5. Therefore these will not impair the efficiency of our algorithm. A typical convergence history is illustrated in Fig. 8.

5. Collision Decoupling

After combining the operations introduced above with the traditional MPRGP algorithm framework, we develop a novel solver for the QP problem Eq. (1) with general linear constraints. However, the computational cost of the method would be higher than the traditional version because we need to solve problems Eq. (15), Eq. (17) and Eq. (19) for the corresponding Lagrange multipliers at each iteration. When there is a large number of simultaneous interbody collisions, this problem would be more serious, see Fig. 2.

In order to reduce the computational cost, we propose to simplify the structure of the matrix JJ^T and $\hat{J}\hat{J}^T$ by modifying the collision constraints. For large models with hundreds of thousands DOFs, we found that it is practical to preserve

Algorithm 3: Modified Monotonic MPRGP outline

Given a SPD matrix A , vectors b, L and initial value $x^0 \geq L$, choose $\Gamma > 0$, $\tilde{\alpha} \in (0, 2\|A\|_2^{-1}]$,
set $k = 0, p = \phi(x^0)$;
while not convergent **do**

- Step 1. {compute $g(x^k), \phi(x^k), \beta(x^k), \tilde{\phi}_{\tilde{\alpha}}(x^k)\}$;
- if** $\|\beta(x^k)\|^2 \leq \Gamma^2 \tilde{\phi}_{\tilde{\alpha}}(x^k)^T \phi(x^k)$ **then**

 - Step 2. {CG steps};
 - compute p, x^{k+1}, y using Algorithm 2;
 - if** not convergent and $y \notin \Omega_B$ **then**

 - Step 3. {expansion step};
 - $y = P_{\Omega_B}(x^k), x^{k+1} = P_{\Omega_B}(y - \tilde{\alpha}\phi(y))$,
 - while** higher function value **do**

 - $| x^{k+1} = P_{\Omega_B}(x^{k+1} - \tilde{\alpha}g(x^{k+1}))$;

 - end**
 - $g = Ax^{k+1} - b, p = \phi(x^{k+1}), k = k + 1$

 - end**
 - else**

 - Step 4. {proportioning step};
 - $\alpha_p = \beta^T(x^k)g(x^k)/\beta^T(x^k)A\beta(x^k)$,
 - $x^{k+1} = P_{\Omega_B}(x^k - \alpha_p\beta(x^k))$,
 - while** higher function value **do**

 - $| x^{k+1} = P_{\Omega_B}(x^{k+1} - \tilde{\alpha}g(x^{k+1}))$;

 - end**
 - $g = Ax^{k+1} - b, p = \phi(x^{k+1}), k = k + 1$;

 - end**

return x^k ;

only one collision pair for each participating vertex, and ignore other collisions related to this vertex. That is, if a vertex appears in several non-penetration constraints with the formulation of Eq. (8), we only preserve one of these constraints (random selected) and ignore the others. Then no two constraints will share the same vertex, and all the constraints are decoupled. By using this scheme, each column of J will contain one non-zero element at most. Then matrix JJ^T and $\hat{J}\hat{J}^T$ would be diagonal. Thus the corresponding problems Eq. (15) Eq. (17) and Eq. (19) can be solved very efficiently. The computational advantage of this strategy is illustrated in Fig. 6. Besides, a stronger result supporting the use of collision decoupling is given in Appendix B, where

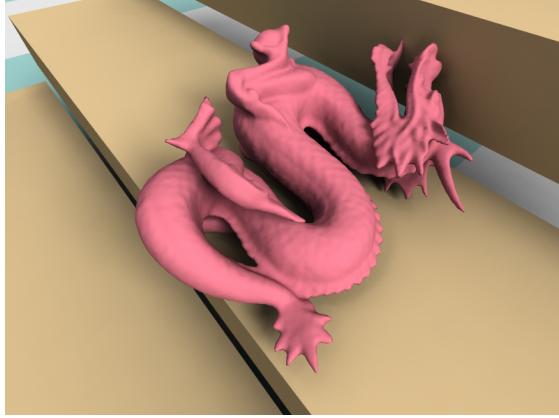


Figure 3: Simulation results for a large dragon model rolling down a stair. Non-penetration constraints of large QP problems are handled efficiently and robustly.

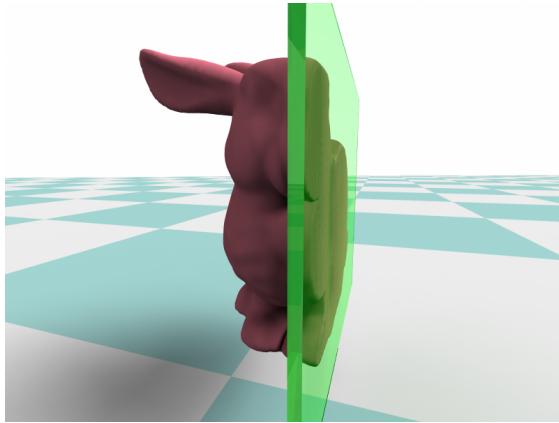


Figure 4: A large bunny model collides heavily with a fixed glass and produces 18342 non-penetration constraints at this timestep.

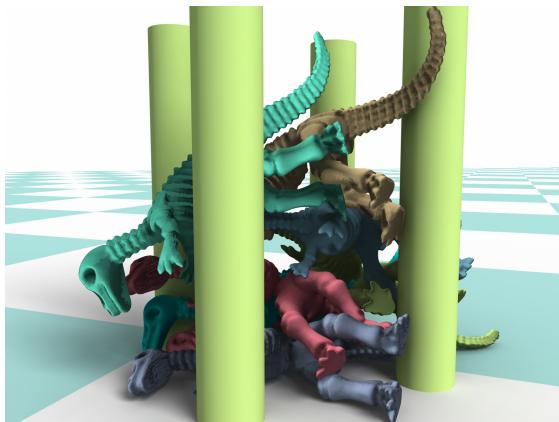


Figure 5: Eight dinosaurs fall together. Collision decoupling scheme is adopted to handle the complex constraints efficiently, while stability and robustness are still preserved.

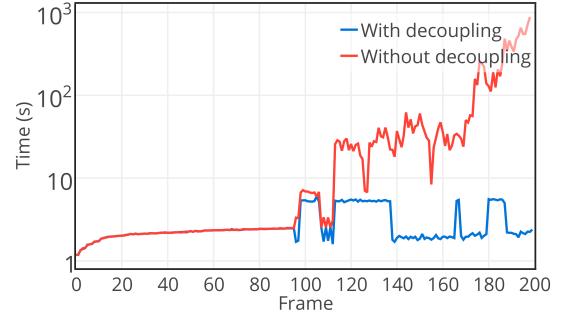


Figure 6: The overhead of our solver in each timestep with and without collision decoupling. A frame of this example is shown in Fig. 5.

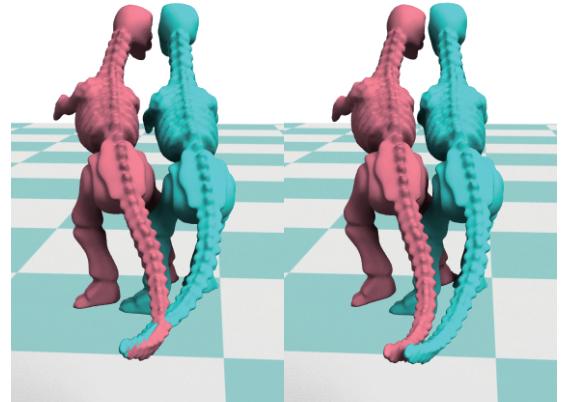


Figure 7: Collisions between some thin regions on the dinosaur mesh cannot be resolved using large timestep size $h = 0.01s$ (left). This problem is alleviated under $h = 0.005s$ (right).

it is proved that the objective function value would strictly decrease after the expansion step so that the additional while loops in Algorithm 3 are indeed unnecessary.

However efficient it is, collision decoupling would introduce some accuracy loss. This may even result in the tunnelling artifacts near thin regions of the mesh as shown in Fig. 7. This may be resolved by adaptive or asynchronous timestepping schemes. Another problem introduced by decoupling is the noise due to randomized selection. A potentially more sophisticated approach is to sort the collisions according to penetration depth and select in that order.

6. Results

We make several experiments to demonstrate the robustness and efficiency of our approach as shown in Fig. 3, Fig. 4, Fig. 5 as well as in the corresponding video. We adopt implicit integration to solve the equation of motion with one Newton's iteration at each timestep. DCD collision detector is adopted, and after the collision decoupling, the con-

strained QP problems are solved by our extended MPRGP algorithm.

In Fig. 3, we simulate a dragon model by solving large QP problems with DOFs= 140208. As our extended MPRGP is an iterative QP solver, it can handle such large QP problem in just 36.4ms on average, which is impossible for traditional active set method [JN06] using direct solvers. In Fig. 4, we show a result with a huge number of collision constraints on the bunny model with more than 40k vertices, which collides heavily to a fixed glass. Because we adopt the implicit strategy, our method is robust and stable in handling such a seriously deformed configuration. Relatively small timestep size is used to show the details of impact (see Table 1), much larger timestep size can be used as long as DCD detector returns correct collision constraints. For the dinosaur example (see Fig. 5), complex collisions need to be resolved. Due to the nature of DCD methods, we need to use small timestep size in this example to avoid undetected collisions, especially for the thin parts of the dinosaur (i.e the tails, foots and hands). Though collision decoupling scheme is adopted to accelerate the MPRGP solver, we have not observed any visual artifacts as shown in our video, which demonstrates the robustness and efficiency of our decoupling scheme for handling complex collisions.

6.1. Performance

The performance of our QP solver is profiled in Table 1. In order to obtain $\bar{\alpha}$, we use the power method to compute $\|A\|_2^{-1}$. We use $\|A\|_2^{-1}$ from the last step as an initial value for computing a new $\|A\|_2^{-1}$. This warm-started power method can be convergent within 40 iterations in our experiments and the time for the $\bar{\alpha}$ estimation has been included in t of this table.

As shown in Fig. 8, our MPRGP solver presents similar convergence speed to the unconstrained CG solver even for a large linear system with 35832 DOFs and 691 non-penetration constraints. We use the same coefficient matrix and right-hand-side vector in both approaches, and just ignore the constraints for the CG method in this experiment. There are different modifications of the outline of the traditional MPRGP Algorithm 1. Theorem 3.2 in [Dos09] indicates that MPRGP converges fast when it generates long chains of CG iterations. Thus in our implementation, we choose the Monotonic MPRGP [Dos09], which inserts in between the last feasible iteration and the expansion step a finite number of unfeasible CG iterations as long as the objective function decreases. Table 1 demonstrates that, in our Monotonic MPRGP solver, the CG steps dominate the iterations. This is why our solver presents similar convergence to unconstrained CG solver in Fig. 8.

Currently, we adopt Jacobi preconditioner in our solver to improve the convergence of the CG iterations. As the CG steps dominate the iterations, this simple precondition-

scene	vertices	tets	h	cg/it	t	T
Fig. 3	46736	160553	10	99.4%	36.4	400
Fig. 4	40107	164419	3	99.1%	24.5	600
Fig. 5	11944	41992	5	99.0%	5.13	400
Fig. 12	9537	49152	10	99.8%	1.72	200

Table 1: Performance measured on a PC with Intel CPU E7500, 2.93GHz, with 8 GB of memory. From left to right, number of vertices (vertices), tetrahedron (tets), timestep size (h , in millisecond), percentage of CG steps in the iterations with the Monotonic MPRGP outline (cg/it), average time to solve the QP for each Newton’s step (t, in second), and the total frames (T).

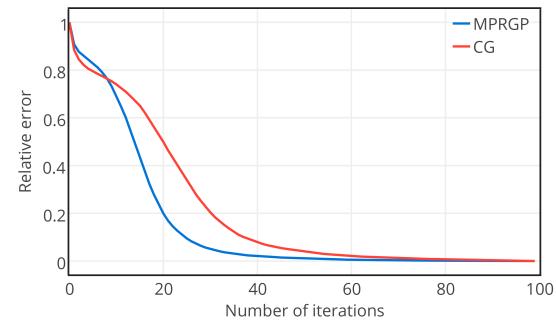


Figure 8: Convergence history of relative error $|F^k - F^*| / |F^0 - F^*|$, where F^k is the objective function value at iteration k and F^* is the optimum. Comparison of the unconstrained CG method with our extended MPRGP under 35832 DOFs and 691 non-penetration constraints.

ing scheme can significantly improve the convergence of our solver as shown in Fig. 9.

6.2. Comparison with ICA

Fig. 10 compares the convergence rate and computational time between our solver and the ICA solver introduced in [Cot09], which is based on PGS. In this comparison, the same set of collisions after applying the constraint decoupling scheme can significantly improve the convergence of our solver as shown in Fig. 9.

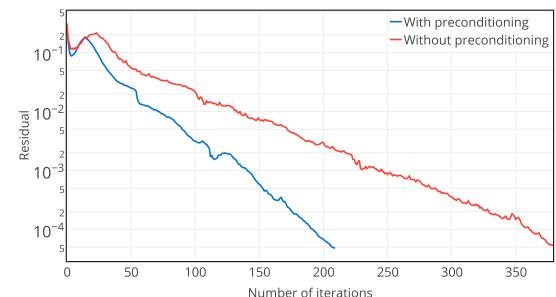


Figure 9: Convergence history of: $\|\phi_i(x) + \beta_i(x)\|$. Comparison of our extended MPRGP solver with and without preconditioning. The example problem is same as that of Fig. 8.

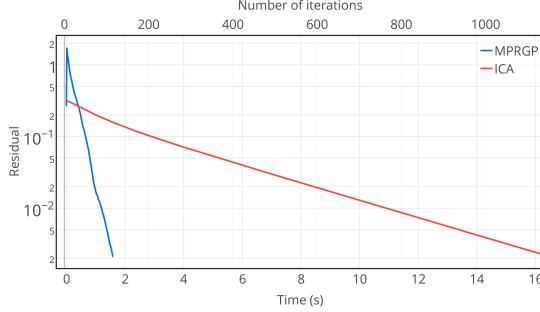


Figure 10: Convergence and time comparison of ICA with our extended MPRGP solver.

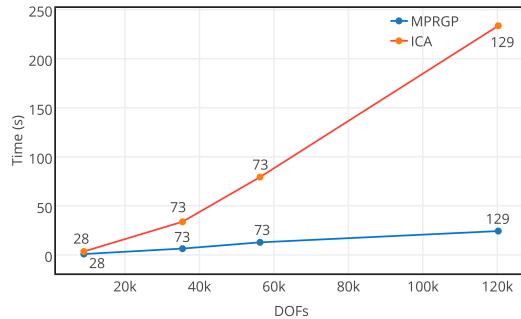


Figure 11: The scaling of ICA and our method against the number of DOFs (number of vertices $\times 3$) used for the dino model. The number of decoupled constraints are marked on data points and models of different resolution are generated via local remeshing. One frame of this example is shown in Fig. 7.

pling is passed to both methods. ICA converges faster initially, but the rate drops quickly after first several iterations. This is consistent with the fact that Jacobian method (and its PGS variant) is effective only for high frequency residuals. The corresponding animation for Fig. 10 is shown in Fig. 12 (as well as in the video). In this example, we can not obtain physically plausible results until the residual (absolute error of the KKT condition) is smaller than 10^{-2} . Under such an accuracy requirement, our method converges more than one order of magnitude faster than ICA.

Because our method provides better convergence rate, the performance advantage over ICA will be more significant on models with higher resolution, which has been verified in our experiments. However, on lower resolutions, ICA should be used since overhead of the sparse solve in $JA^{-1}J^T$ becomes marginal. Fig. 11 illustrates the scaling of both algorithms against the number of vertices.

7. Conclusion and Limitation

In this paper, we introduce a novel collision handler for volumetric deformable objects, which is robust, stable and efficient in handling deformable bodies with hundreds of thou-

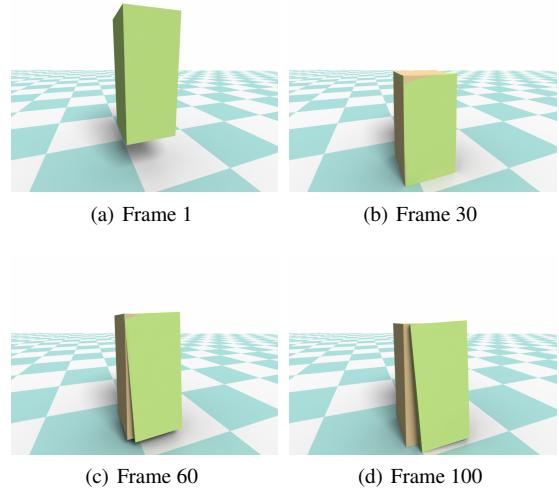


Figure 12: Non-plausible (in yellow) and plausible (in green) results generated by using relative error tolerance as 10^{-1} and 10^{-2} respectively.

sands of DOFs. Our solver extends the traditional MPRGP algorithm to support general linear constraints arising in collision handling while preserving its high convergence rate. The constraint decoupling scheme further improves the efficiency by ignoring a subset of collisions. Besides, our method naturally inherits the typical advantages of PCG method: it can be easily implemented on a massive parallel processor, and better convergence can be achieved by incorporating a more sophisticated preconditioner such as a geometric multigrid [ZSTB10]. We leave these as potential future works. On the other hand, the convergence of such solver is also dominated by the conditioning of A , see [Dos09] for a proof. A good preconditioner is required for acceptable convergence rate if bad meshing or high stiffness is used for the elastic body.

However, we have made some non-trivial assumption in our development that worth further exploration. The collision decoupling ignores some collision constraints without noticeable artifacts, but it nevertheless leads to accuracy loss, which is a common limitation of many methods simplifying the collision constraints [BJ08]. Besides, we have assumed frictionless contacts in all results. Unfortunately, the LCP problem resulted from the frictional cone constraints cannot be properly decoupled to be handled efficiently by our method. If they are desired, one can combine our solver with previous methods such as [OTSG09] taking the Lagrangian multipliers computed by our method as input or consider each frictional contact locally as [BMF03].

Acknowledgements

This work was partially supported by NSFC (No.61170139 and No.61210007), Fundamental Research Funds for the

Central Universities (No. 2015FZA5018). Xiaogang Jin was supported by NSFC (Grant no. 61272298).

References

- [AKJ08] AN S. S., KIM T., JAMES D. L.: Optimizing cubature for efficient integration of subspace deformations. *ACM Trans. Graph.* 27, 5 (2008), 165:1–165:10. [2](#)
- [Bar94] BARAFF D.: Fast contact force computation for nonpenetrating rigid bodies. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1994), SIGGRAPH '94, ACM, pp. 23–34. [2](#)
- [Bar96] BARAFF D.: Linear-time dynamics using lagrange multipliers. In *In SIGGRAPH 96 Conference Proceedings, Computer Graphics Proceedings, Annual Conference Series* (1996), ACM SIGGRAPH, Addison Wesley, pp. 137–146.
- [BFA02] BRIDSON R., FEDKIW R., ANDERSON J.: Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph.* 21, 3 (July 2002), 594–603. [2](#)
- [BJ05] BARBIĆ J., JAMES D. L.: Real-time subspace integration for St. Venant-Kirchhoff deformable models. *ACM Trans. Graph.* 24, 3 (2005), 982–990. [2](#)
- [BJ08] BARBIĆ J., JAMES D. L.: Six-dof haptic rendering of contact between geometrically complex reduced deformable models. *IEEE Trans. Haptics* 1, 1 (Jan. 2008), 39–52. [8](#)
- [BMF03] BRIDSON R., MARINO S., FEDKIW R.: Simulation of clothing with folds and wrinkles. In *Symposium on Computer Animation* (2003), pp. 28–36. [8](#)
- [BW92] BARAFF D., WITKIN A.: Dynamic simulation of non-penetrating flexible bodies. In *Computer Graphics* (1992), ACM, pp. 303–308. [2](#)
- [BW98] BARAFF D., WITKIN A.: Large steps in cloth simulation. In *Proceedings of ACM SIGGRAPH* (1998), pp. 43–54. [2](#)
- [CK02] CHOI K.-J., KO H.-S.: Stable but responsive cloth. *ACM Trans. Graph.* 21, 3 (July 2002), 604–611. [2](#)
- [CK05] CHOI M. G., KO H.-S.: Modal warping: Real-time simulation of large rotational deformation and manipulation. *IEEE Transactions on Visualization and Computer Graphics* 11, 1 (2005), 91–101. [2](#)
- [Cot09] COTTLE R.: Linear complementarity problem linear complementarity problem. In *Encyclopedia of Optimization*, Floudas C. A., Pardalos P. M., (Eds.). Springer US, 2009, pp. 1873–1878. [1, 2, 7](#)
- [DDKA06] DURIEZ C., DUBOIS F., KHEDDAR A., ANDRIOT C.: Realistic haptic rendering of interacting deformable objects in virtual environments. *IEEE Transactions on Visualization and Computer Graphics* 12, 1 (Jan. 2006), 36–47. [2](#)
- [Dos09] DOSTÁL Z.: *Optimal Quadratic Programming Algorithms With Applications to Variational Inequalities*. Springer-Verlag US, 2009. [1, 2, 3, 4, 5, 7, 8, 10](#)
- [Erl07] ERLEBEN K.: Velocity-based shock propagation for multibody dynamics animation. *ACM Trans. Graph.* 26, 2 (June 2007), 12. [2](#)
- [GB13] GERSZEWSKI D., BARGEIL A. W.: Physics-based animation of large-scale splashing liquids. *ACM Trans. Graph.* 32, 6 (2013), 185. [2](#)
- [HFL00] HIROTA G., FISHER S., LIN M. C.: Simulation of non-penetrating elastic bodies using distance fields. *Tech. rep., University of North Carolina at Chapel Hill* (2000). [2](#)
- [HVTG08] HARMON D., VOUGA E., TAMSTORF R., GRINSPIUN E.: Robust treatment of simultaneous collisions. *SIGGRAPH (ACM Transactions on Graphics)* 27, 3 (2008), 1–4. [2](#)
- [JN06] JORGE NOCEDAL S. J. W.: *Numerical Optimization*. Springer-Verlag US, 2006. [1, 2, 3, 7](#)
- [KEP05] KAUFMAN D. M., EDMUND S., PAI D. K.: Fast frictional dynamics for rigid bodies. *ACM Trans. Graph.* 24, 3 (2005), 946–956. [2](#)
- [KSJP08] KAUFMAN D. M., SUEDA S., JAMES D. L., PAI D. K.: Staggered projections for frictional contact in multibody systems. *ACM Trans. Graph.* 27, 5 (Dec. 2008), 164:1–164:11. [2](#)
- [MZS*11] MCADAMS A., ZHU Y., SELLE A., EMPEY M., TAMSTORF R., TERAN J., SIFAKIS E.: Efficient elasticity for character skinning with contact and collisions. *ACM Trans. Graph.* 30, 4 (July 2011), 37:1–37:12. [2](#)
- [NGL10] NARAIN R., GOLAS A., LIN M. C.: Free-flowing granular materials with two-way solid coupling. vol. 29, ACM, p. 173. [2](#)
- [ORC07] ORTEGA M., REDON S., COQUILLART S.: A six degree-of-freedom god-object method for haptic display of rigid bodies with surface properties. *IEEE Transactions on Visualization and Computer Graphics* 13, 3 (May 2007), 458–469. [2](#)
- [OTSG09] OTADUY M. A., TAMSTORF R., STEINEMANN D., GROSS M.: Implicit contact handling for deformable objects. *Computer Graphics Forum (Proc. of Eurographics)* 28, 2 (apr 2009). [1, 2, 8](#)
- [PPG04] PAULY M., PAI D. K., GUIBAS L. J.: Quasi-rigid objects in contact. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2004), Eurographics Association, pp. 109–119. [2](#)
- [PW89] PENTLAND A., WILLIAMS J.: Good vibrations: modal dynamics for graphics and animation. *Proceedings of ACM SIGGRAPH* 23, 3 (1989), 207–214. [2](#)
- [RKC02] REDON S., KHEDDAR A., COQUILLART S.: Gauss' least constraints principle and rigid body simulations. In *In proceedings of IEEE International Conference on Robotics and Automation* (2002), pp. 11–15. [2](#)
- [RKLM04] REDON S., KIM Y. J., LIN M. C., MANOCHA D.: Fast continuous collision detection for articulated models. In *Proceedings of the Ninth ACM Symposium on Solid Modeling and Applications* (Aire-la-Ville, Switzerland, Switzerland, 2004), SM '04, Eurographics Association, pp. 145–156. [2](#)
- [ST96] STEWART D. E., TRINKLE J. C.: An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *International Journal for Numerical Methods in Engineering* 39 (1996), 2673–2691. [2](#)
- [TCYM08] TANG M., CURTIS S., YOON S.-E., MANOCHA D.: Interactive continuous collision detection between deformable models using connectivity-based culling. In *Proceedings of the 2008 ACM Symposium on Solid and Physical Modeling* (New York, NY, USA, 2008), SPM '08, ACM, pp. 25–36. [2](#)
- [TKZ*04] TESCHNER M., KIMMERLE S., ZACHMANN G., HEIDELBERGER B., RAGHUPATHI L., FUHRMANN A., CANI M.-P., FAURE F., MAGNETAT-THALMANN N., STRASSER W.: Collision detection for deformable objects. In *Eurographics State-of-the-Art Report (EG-STAR)* (2004), Eurographics Association, Eurographics Association, pp. 119–139. [2](#)
- [TMOT12] TANG M., MANOCHA D., OTADUY M. A., TONG R.: Continuous penalty forces. *ACM Trans. Graph.* 31, 4 (2012), 107. [2](#)
- [TOK14] TENG Y., OTADUY M. A., KIM T.: Simulating articulated subspace self-contact. *ACM Trans. Graph.* 33, 4 (July 2014), 106:1–106:9. [2](#)
- [TPBF87] TERZOPOULOS D., PLATT J., BARR A., FLEISCHER K.: Elastically deformable models. *SIGGRAPH Comput. Graph.* 21, 4 (Aug. 1987), 205–214. [2](#)

[ZSTB10] ZHU Y., SIFAKIS E., TERAN J., BRANDT A.: An efficient multigrid method for the simulation of high-resolution elastic solids. *ACM Trans. Graph.* 29, 2 (Apr. 2010), 16:1–16:18. 2, 8

Appendix A: Guaranteed Decreasing Direction

To prove the convergence of Algorithm 3, we need to show that $\phi^T g \geq 0$ and $\beta^T g \geq 0$. These just follow from fundamental linear algebra, but we present the details here for completeness.

Lemma 1 The solution of Eq. (15) satisfies $\phi^T g \geq 0$ and $\phi^T g_\beta = 0$.

Proof The first assertion can be seen from the following expansion:

$$\frac{1}{2} \|\phi - g\|_2^2 = \frac{1}{2} \|\phi\|_2^2 - 2\phi^T g + \frac{1}{2} \|g\|_2^2.$$

Thus, if $\phi^T g < 0$ then $-\phi$, which is also feasible, is a better solution. The second assertion can be proved by solving the KKT system Eq. (15) analytically for ϕ :

$$\phi = g - \tilde{J}^T (\tilde{J}\tilde{J}^T)^{-1} \tilde{J}g = g - Pg,$$

which in turn reveals that $\phi^T g_\beta = g^T Pg - g^T P^2 g = 0$. \square

We then show that $\beta^T \phi = 0$ and $\beta^T g_\beta \geq 0$, so that: $\beta^T g \geq 0$.

Lemma 2 The solution of Eq. (17) satisfies $\beta^T \phi = 0$.

Proof This can be proved similarly, i.e., if $\beta^T \phi \neq 0$, we show that: $\hat{\beta} = \beta - p_\phi \phi$, where $p_\phi = \beta^T \phi / \phi^T \phi$, is a better solution. Clearly, this is a feasible solution since $\hat{J}\phi = 0$. By plugging $\hat{\beta}$ into Eq. (17), we get:

$$\begin{aligned} \frac{1}{2} \|\beta - g_\beta - p_\phi \phi\|_2^2 &= \frac{1}{2} \|\beta - g_\beta\|_2^2 + \frac{1}{2} p_\phi^2 \phi^T \phi - p_\phi \phi^T \beta \\ &= \frac{1}{2} (\|\beta - g_\beta\|_2^2 - \frac{1}{2} p_\phi^2 \phi^T \phi). \end{aligned}$$

Where we have used the property $\phi^T g_\beta = 0$ from Lemma 1.

\square

Lemma 3 The solution of Eq. (17) satisfies $\beta^T g_\beta \geq 0$.

Proof If we denote by \hat{J}_A the set of active constraints of Eq. (17), then the following system gives the same solution:

$$\frac{1}{2} \|\beta - g_\beta\|_2^2 \quad \hat{J}_A \beta = 0.$$

The result thus follows immediately from Lemma 1. \square

Appendix B: Guaranteed Function Value Decrease

Here, we want to prove the following lemma:

Lemma 4 Assuming $F(x) = \frac{1}{2}x^T Ax - x^T b$ and J is derived from a decoupled set of collisions, we have:

$$F(P_{\Omega_B}(x - \bar{\alpha}g(x))) \leq \eta_f F(x),$$

where $\eta_f = 1 - \min(\bar{\alpha}, 2\|A\|_2^{-1} - \bar{\alpha})\lambda_{\min}(A)$.

Proof Without loss of generality, for the matrix $J^{m \times n}$, we assume $m < n$ and each row of J is normalized so that $JJ^T = I$. We can then augment J with $n - m$ mutually orthogonal basis to get:

$$\tilde{J}^{n \times n} \triangleq \begin{pmatrix} J^{m \times n} \\ J_{aug}^{(n-m) \times n} \end{pmatrix},$$

so that $\tilde{J}^T \tilde{J} = \tilde{J} \tilde{J}^T = I$. Since \tilde{J} is invertible, we can introduce a new set of variables s such that $s = \tilde{J}x$ and $F_s(s) \triangleq F(x)$. In this way, the Projection operator under s becomes:

$$P_{\Omega_c}^s(s) = \min_{s^*} \frac{1}{2} \|s^* - s\|_2^2, \quad \text{s.t. } s^* \geq \tilde{c},$$

which is derived by substituting x, y with $\tilde{J}^T s, \tilde{J}^T s^*$ in $P_{\Omega_c}(x)$ where \tilde{c} is defined by augmenting c with $-\infty$. This is a separable problem so that the Proposition 5.10 of [Dos09] applies, giving:

$$F_s(P_{\Omega_c}^s(s - \bar{\alpha}g_s)) \leq \eta_f F_s(s) = \eta_f F(x),$$

where $g_s = \tilde{J}(AJ^T s - b) = \tilde{J}g$. Note that $\bar{\alpha}$ and η_f will not be changed by our variable substitution since eigenvalues are invariant under similarity transform. To reach our final conclusion, we note that:

$$\begin{aligned} F_s(P_{\Omega_c}^s(s - \bar{\alpha}g_s)) &= F(\tilde{J}^T P_{\Omega_c}^s(s - \bar{\alpha}g_s)) = \\ F(\tilde{J}^T P_{\Omega_c}^s(\tilde{J}(x - \bar{\alpha}g))) &= F(P_{\Omega_c}(x - \bar{\alpha}g)). \end{aligned}$$

\square