Oct, 2017

PARMES - technical report - 2 - revision - 1

A projected quasi-Newton method for dynamic multibody frictional contact problems

Summary

We present a simple projected quasi-Newton approach for dealing with frictional contact constraints in multibody dynamic simulations. A regularized linear subproblem is solved inexactly at every Newton step, rendering the method attractive in the context of parallel computations. Our approach performs comparably with the commonly used Gauss-Seidel scheme, while having a much lower communication cost in parallel. We include examples of rigid and deformable problems, run on a distributed memory cluster. An implementation of the presented method is available at https://parmes.github.io/solfec

Contributors

Tomasz Koziara, t.koziara@gmail.com

1. Introduction

The motivation behind the presented work is in modeling the dynamics of large collections of bodies interacting via contact and friction. Our point of departure is Contact Dynamics, an implicit method originated by Moreau [1] and Jean [2], facilitating time integration with relatively large time steps. In order to solve for contact reactions Contact Dynamics employs nonlinear Gauss-Seidel relaxation. Although this approach is quite robust, its distributed memory implementation requires much care [3], and can be inefficient when convergence is slow. In an earlier research [4] we have looked into a nonsmooth Newton approach, easier to parallelize, but representing potentially troublesome due to oscillatory behavior. The current work proposes an alternative formulation.

We use the work of De Saxcé and Feng [5] and modify the linearization by Fukushima et al. [6] in order to formulate the dynamic frictional contact problem as smooth root finding. In order to find approximate solutions, we subsequently develop a projected quasi-Newton method, which scales in parallel and converges sufficiently well to replace the Gauss-Seidel approach for the problem set of our interest. This approach may be useful in a broader sense and hence it is documented in the current report. We refer the reader to reviews [7, 8] for a perspective on finite-dimensional multibody contact/impact dynamics. The list of cross-references of review [7] may be helpful in tracking recent developments in the field. For example a solution method by Mazhar et al. [9] has a similar character to the one presented here.

A brief account of discrete multibody dynamics is given in Section 2. Section 3 formulates the frictional contact problem. Smoothing is discussed in Section 4. In Section 5 a convenient measure of accuracy is defined. Linearization is outlined in Section 6. The projected quasi-Newton method is detailed in Section 7, followed by examples in Section 8. The paper is concluded in Section 9. In addition, Appendix A includes derivation of a smoothed derivative, while Appendix B provides links to source code and examples files.

2. Discrete multibody dynamics

At time t a set of bodies has configuration $\mathbf{q}^t$ and velocity $\mathbf{u}^t$. Consecutive values $\mathbf{q}^{t+h}$, $\mathbf{u}^{t+h}$ for $h > 0$ are obtained by numerical integration of the momentum balance and the kinematic relation $\dot{\mathbf{q}} = \mathbf{u}$. In order to focus attention, we briefly outline a scheme originally presented in [1]. A half-step configuration is obtained as

$$\mathbf{q}^{t+h/2} = \mathbf{q}^t + \frac{h}{2}\mathbf{u}^t. \tag{1}$$

The velocity update reads

$$\mathbf{u}^{t+h} = \mathbf{b}^{t+h} + \mathbf{A}^{-1}\mathbf{H}^T\mathbf{R} \tag{2}$$

where

$$\mathbf{b}^{t+h} = \mathbf{u}^t + h\mathbf{A}^{-1}\mathbf{f} \tag{3}$$

is a free velocity, not accounting for an influence of constraints. When $\mathbf{A} = \mathbf{M}$, where $\mathbf{M}$ is an inertia matrix, (3) corresponds to an explicit scheme detailed in [1, 10]. For $\mathbf{A} = \mathbf{M} + \left(h^2/4\right)\mathbf{K}$, where $\mathbf{K} = -\{\partial\mathbf{f}/\partial\mathbf{q}\}\left(\mathbf{q}^{t+h/2}\right)$ is a stiffness matrix, (3) corresponds to the linearly-implicit scheme by Zhang and Skeel [11]. In both cases the out of balance force $\mathbf{f} = \mathbf{f}\left(\mathbf{q}^{t+h/2}, t + \frac{h}{2}\right)$. For smooth problems both integrators have good conservation properties and are of second order. Naturally, the linearly-implicit scheme allows for larger time steps. While (3) represents a regular velocity update for an unconstrained system, (2) accounts for the constraints by adding a contribution of their reactions $\mathbf{R}$. In (2) $\mathbf{A} = \mathbf{A}\left(\mathbf{q}^{t+h/2}\right)$ and $\mathbf{H} = \mathbf{H}\left(\mathbf{q}^{t+h/2}\right)$, while $\mathbf{R} = \int_t^{t+h} d\mathbf{R}$ is considered to be an impulse. $\mathbf{H}$ is a linear operator mapping the body space velocity $\mathbf{u}$ into a constraint space velocity $\mathbf{U}$

$$\mathbf{U} = \mathbf{H}\mathbf{u}. \tag{4}$$

For example, in case of finite elements, $\mathbf{H}$ is a global to local coordinates transformation of the shape functions values nonzero at constraint points. Once $\mathbf{R}$ is known, after obtaining $\mathbf{u}^{t+h}$ the final configuration update reads

$$\mathbf{q}^{t+h} = \mathbf{q}^{t+h/2} + \frac{h}{2}\mathbf{u}^{t+h}. \tag{5}$$

In the remaining part of the paper we discuss how to calculate $\mathbf{R}$ in case of frictional contact constraints.

3. Frictional contact formulation

At $\mathbf{q}^{t+h/2}$ we identify a number of points where pairs of bodies interact by contact. At those points, indexed by α, we introduce local coordinate systems. Each of those systems has an axis pointing into the normal direction. This is the direction along which bodies separate. The remaining two axes span a tangent plane. This is the plane in which bodies shear. At each contact point, using the local coordinate system, we consider a relative velocity $\mathbf{U}_\alpha$ and an impulse force $\mathbf{R}_\alpha$. Their relationship is linear

$$\mathbf{U}_\alpha = \sum_\beta \mathbf{W}_{\alpha\beta}\mathbf{R}_\beta + \mathbf{B}_\alpha \tag{6}$$

where $\mathbf{W}_{\alpha\beta}$ are 3×3 block matrices and $\mathbf{B}_\alpha$ are 3-vectors contributing to

$$\mathbf{W} = \mathbf{H}\mathbf{A}^{-1}\mathbf{H}^T \tag{7}$$

$$\mathbf{B} = \mathbf{H}\mathbf{b}^{t+h} \tag{8}$$

in a suitable manner. Because for $\mathbf{R} = \mathbf{0}$ there holds $\mathbf{U} = \mathbf{B}$, $\mathbf{B}$ is the free relative velocity at the constraints.

At contact points we consider a minimalistic interface law, reflecting the core difficulties related to the lack of smoothness and the non-associated character of friction. In terms of normal behaviour we employ the velocity Signorini condition, $U_{\alpha N} \geq 0$, $R_{\alpha N} \geq 0$, $U_{\alpha N} R_{\alpha N} = 0$. In terms of shearing behavior we utilize the Coulomb law, $\mathbf{R}_\alpha \in \text{interior}\,(K_\alpha) \Rightarrow \mathbf{U}_{\alpha T} = 0$ and $\mathbf{R}_\alpha \in \text{boundary}\,(K_\alpha) \Rightarrow \mathbf{U}_{\alpha T} = -\gamma \mathbf{R}_{\alpha T}$, where $\gamma > 0$. The friction cone K_α is defined as

$$K_\alpha = \{\mathbf{R}_\alpha : \|\mathbf{R}_{\alpha T}\| \leq \mu_\alpha R_{\alpha N}, R_{\alpha N} \geq 0\} \tag{9}$$

where μ_α is the coefficient of friction. It has been shown by De Saxcé and Feng [5], that the Signorini-Coulomb law can be expressed in a compact form

$$-\begin{bmatrix} \mathbf{U}_{\alpha T} \\ U_{\alpha N} + \mu_\alpha \|\mathbf{U}_{\alpha T}\| \end{bmatrix} \in N_{K_\alpha}(\mathbf{R}_\alpha) \tag{10}$$

where N_{K_α} stands for the normal cone of the set K_α. For a convex set A the normal cone $N_A(\mathbf{R})$ at point $\mathbf{R} \in A$ is defined as the set of all vectors $\mathbf{V}$ such that $\langle \mathbf{V}, \mathbf{S} - \mathbf{R} \rangle \leq 0$ for all $\mathbf{S} \in A$. Let us now define

$$\mathbf{F}(\mathbf{R}) = \begin{bmatrix} \dots \\ \mathbf{U}_{\alpha T}(\mathbf{R}) \\ U_{\alpha N}(\mathbf{R}) + \mu_\alpha \|\mathbf{U}_{\alpha T}(\mathbf{R})\| \\ \dots \end{bmatrix} \tag{11}$$

and

$$K = \prod_\alpha K_\alpha \tag{12}$$

where the dependence $\mathbf{U}_\alpha(\mathbf{R})$ is defined in (6), and $\mathbf{R}$ is a collection of $\mathbf{R}_\alpha$ for $\alpha \in \{1, ..., \nu\}$. Formula (10) states, that the frictional contact constraints are satisfied if $-\mathbf{F}(\mathbf{R})$ belongs to the normal cone of K at $\mathbf{R}$. Hence

$$-\mathbf{F}(\mathbf{R}) = \mathbf{R} - \mathbf{F}(\mathbf{R}) - \mathrm{proj}_K(\mathbf{R} - \mathbf{F}(\mathbf{R})) \tag{13}$$

which can be reduced to the standard projection formula $\mathbf{R} = \mathrm{proj}_K(\mathbf{R} - \mathbf{F}(\mathbf{R}))$. Let us not do it though, but rather define a vector field

$$\mathbf{m}(\mathbf{S}) = \mathbf{S} - \mathrm{proj}_K(\mathbf{S}). \tag{14}$$

We can rewrite (10) as

$$\mathbf{C}(\mathbf{R}) = \mathbf{F}(\mathbf{R}) + \mathbf{m}(\mathbf{R} - \mathbf{F}(\mathbf{R})) = \mathbf{0} \text{ and } \mathbf{R} \in K, \tag{15}$$

a root finding problem defined within the friction cone.

4. Smoothing

Projection is a nonsmooth operator and so is $\mathbf{m}(\mathbf{S})$. Similarly, $\mathbf{F}(\mathbf{R})$ is not smooth due to $\|\mathbf{U}_{\alpha T}(\mathbf{R})\|$. Since our aim to to develop a Newton-like approach for equation (15), some suitable notion of derivative needs to be put in place. Plethora of nonsmooth analytical tools have been developed in recent decades, many of which are summarized in the monograph [12]. This avenue has been investigated in our previous work [4]. In this paper we are interested in exploring the classical smooth toolbox and hence we adopt the work of Fukushima at al. [6] in the context of frictional contact. Since we shall need a gradient of $\mathbf{C}(\mathbf{R})$, our first step is to produce its smooth approximation $\mathbf{C}_\omega(\mathbf{R})$, such that $\nabla \mathbf{C}_\omega(\mathbf{R})$ is well defined for all $\omega > 0$.

4.1. Smoothed $\mathbf{m}(\mathbf{S})$

A polar cone K_α° of the friction cone K_α is defined as follows

$$K_\alpha^\circ = \{\mathbf{V} : \langle \mathbf{V}, \mathbf{S} \rangle \leq 0 \text{ for all } \mathbf{S} \in K_\alpha\} = N_{K_\alpha}(\mathbf{0}). \tag{16}$$

For any $\mathbf{S}_\alpha$ we have (Moreau)

$$\mathbf{S}_\alpha = \text{proj}_{K_\alpha}(\mathbf{S}_\alpha) + \text{proj}_{K_\alpha^\circ}(\mathbf{S}_\alpha) \tag{17}$$

and hence

$$\mathbf{m}_\alpha(\mathbf{S}_\alpha) = \text{proj}_{K_\alpha^\circ}(\mathbf{S}_\alpha). \tag{18}$$

We use the analysis from [6] in order to derive a smoothed $\mathbf{m}(\mathbf{S})$. In [6] only the self-dual case $\mu_\alpha = 1$ was considered. The formulas obtained here represent a slight generalization. In order to avoid clutter, we temporarily omit the α-indexing of contact point entities. For any $\mathbf{z} \in R^3$ we define the following spectral factorization with respect to the pair of cones $-K$ and K°

$$\mathbf{z} = \lambda_1 \mathbf{u}_1 + \lambda_2 \mathbf{u}_2 \tag{19}$$

where λ_1, λ_2 are spectral values

$$\lambda_1 = \frac{-z_N - \mu \|\mathbf{z}_T\|}{1+\mu^2}, \quad \lambda_2 = \frac{\|\mathbf{z}_T\| - \mu z_N}{1+\mu^2} \tag{20}$$

and $\mathbf{u}_1$, $\mathbf{u}_2$ are spectral vectors

$$\mathbf{u}_1 = \begin{cases} \begin{bmatrix} -\mu \frac{\mathbf{z}_T}{\|\mathbf{z}_T\|} \\ -1 \end{bmatrix} & \text{if} \quad \mathbf{z}_T \neq \mathbf{0} \\ \begin{bmatrix} -\mu \mathbf{w} \\ -1 \end{bmatrix} & \text{if} \quad \mathbf{z}_T = \mathbf{0} \end{cases} \tag{21}$$

$$\mathbf{u}_2 = \begin{cases} \begin{bmatrix} \frac{\mathbf{z}_T}{\|\mathbf{z}_T\|} \\ -\mu \end{bmatrix} & \text{if} \quad \mathbf{z}_T \neq \mathbf{0} \\ \begin{bmatrix} \mathbf{w} \\ -\mu \end{bmatrix} & \text{if} \quad \mathbf{z}_T = \mathbf{0} \end{cases} \tag{22}$$

where $\mathbf{w} \in R^2$ and $\|\mathbf{w}\| = 1$. For any $\mathbf{z} \in R^3$ we can write

$$\text{proj}_{K^\circ}(\mathbf{z}) = \max(0, \lambda_1)\mathbf{u}_1 + \max(0, \lambda_2)\mathbf{u}_2 \tag{23}$$

where λ_1, λ_2 and $\mathbf{u}_1, \mathbf{u}_2$ are defined in (20), (21) and (22). For any $\mathbf{z} \in R^3$ with $\mathbf{z}_T \neq \mathbf{0}$ we take a slice of R^3 and span a Cartesian x-y frame along $\mathbf{z}_T$ and $[0,0,1]^T$ respectively. Then $x_\mathbf{z} > 0$, $\mathbf{u}_1$ corresponds to a x-negative face of K° and $\mathbf{u}_2$ corresponds to a x-positive face of $-K$. Clearly, $\mathbf{u}_1 \perp \mathbf{u}_2$ and λ_1 and λ_2 are the normalized projections of $\mathbf{z}$ onto these two directions. Formula (23) follows from the geometry of $-K$ and K°.

The projection operator as defined in (23) is nonsmooth. Motivated by [6, 13] we employ the following smoothing

$$\mathrm{proj}^{\omega}_{K^\circ}(\mathbf{z}) = \omega g(\lambda_1/\omega)\,\mathbf{u}_1 + \mu\omega g(\lambda_2/(\mu\omega))\,\mathbf{u}_2 \tag{24}$$

where

$$g(\lambda) = \frac{1}{2}\left(\sqrt{\lambda^2+4}+\lambda\right). \tag{25}$$

The smoothing function $\omega g(\lambda/\omega)$ is differentiable for all $\omega > 0$ and it converges to $\max(0,\lambda)$ when $\omega \downarrow 0$. There also hold

$$\lim_{\lambda\to-\infty} g(\lambda) = 0, \quad \lim_{\lambda\to\infty}(g(\lambda)-\lambda) = 0, \quad 0 < \dot{g}(\lambda) < 1 \tag{26}$$

mimicking the behavior of $\max(0,\lambda)$. The gradient of the smoothed projection onto the polar cone reads

$$\nabla\mathrm{proj}^{\omega}_{K^\circ}(\mathbf{z}) = \begin{cases} \dot{g}(\lambda_1/\omega)\,\mathbf{I} & \text{if} \quad \mathbf{z}_T = \mathbf{0} \\ \begin{bmatrix} a\mathbf{I} + (b-a)\,\mathbf{z}_T\mathbf{z}_T^T/\|\mathbf{z}_T\|^2 & c\mathbf{z}_T/\|\mathbf{z}_T\| \\ c\mathbf{z}_T^T/\|\mathbf{z}_T\| & d \end{bmatrix} & \text{if} \quad \mathbf{z}_T \neq \mathbf{0} \end{cases} \tag{27}$$

where

$$a = \frac{g(\lambda_2/(\mu\omega)) - g(\lambda_1/\omega)}{\lambda_2/(\mu\omega) - \lambda_1/\omega}, b = \frac{\mu^2\dot{g}(\lambda_1/\omega) + \dot{g}(\lambda_2/(\mu\omega))}{1+\mu^2} \tag{28}$$

$$c = \frac{\mu(\dot{g}(\lambda_1/\omega) - \dot{g}(\lambda_2/(\mu\omega)))}{1+\mu^2}, d = \frac{\dot{g}(\lambda_1/\omega) + \mu^2\dot{g}(\lambda_2/(\mu\omega))}{1+\mu^2}. \tag{29}$$

A detailed derivation of the above formula can be found in Appendix 2, where after [6, 13] we also show that $\nabla\mathrm{proj}^{\omega}_{K^\circ}$ is positive definite for $\omega > 0$. Finally, the smoothed $\mathbf{m}(\mathbf{S})$ reads

$$\mathbf{m}_\omega(\mathbf{S}) = \mathrm{proj}^{\omega}_{K^\circ}(\mathbf{S}). \tag{30}$$

4.1.1. Zero friction

It is instructive to investigate the behavior of $\mathbf{m}_\omega(\mathbf{S})$ and its gradient in the limit case of $\mu = 0$. For $g_1 = \omega g(\lambda_1/\omega)$ and $g_2 = \mu\omega g(\lambda_2/(\mu\omega))$ we have

$$g_1 = \frac{1}{2}\left(\lambda_1 + \sqrt{\lambda_1^2+4\omega^2}\right), g_2 = \frac{1}{2}\left(\lambda_2 + \sqrt{\lambda_2^2+4\mu^2\omega^2}\right) \tag{31}$$

and hence

$$\mathbf{m}_\omega(\mathbf{S}) = g_1\mathbf{u}_1 + \max(0,\lambda_2)\,\mathbf{u}_2. \tag{32}$$

The tangential direction $\mathbf{u}_2$ is not active in the frictionless case and it is not smoothed. For $\mu = 0$ the tangential component of $\mathbf{C}(\mathbf{R})$ reads

$$\mathbf{C}_T(\mathbf{R}) = \mathbf{F}_T(\mathbf{R}) + \mathbf{m}_T(\mathbf{R} - \mathbf{F}(\mathbf{R})) = \mathbf{R}_T \tag{33}$$

and hence $\mathbf{C}_T(\mathbf{R}) = \mathbf{0}$ implies $\mathbf{R}_T = \mathbf{0}$. This is also reflected in $\nabla_\omega \mathbf{m}(\mathbf{S})$ and consequently in $d\mathbf{C}_\omega(\mathbf{U}, \mathbf{R}) = \nabla_\omega \mathbf{m}(\mathbf{R} - \mathbf{U})\, d\mathbf{R}$, since $\nabla \mathbf{F}(\mathbf{U}) = \mathbf{I}$ for $\mu = 0$. Let us produce practical versions of (28) and (29) first. We have

$$\dot{g}_1 = \frac{1}{2}\left(1 + \lambda_1 / \sqrt{\lambda_1^2 + 4\omega^2}\right), \dot{g}_2 = \frac{1}{2}\left(1 + \lambda_2 / \sqrt{\lambda_2^2 + 4\mu^2\omega^2}\right) \tag{34}$$

and

$$a = \frac{1}{2}\left(1 + (\lambda_2 + \mu\lambda_1) / \left(\sqrt{\lambda_2^2 + 4\mu^2\omega^2} + \mu\sqrt{\lambda_1^2 + 4\omega^2}\right)\right) \tag{35}$$

$$b = \left(\mu^2 \dot{g}_1 + \dot{g}_2\right) / \left(1 + \mu^2\right) \tag{36}$$

$$c = \mu\left(\dot{g}_1 - \dot{g}_2\right) / \left(1 + \mu^2\right) \tag{37}$$

$$d = \left(\dot{g}_1 + \mu^2 \dot{g}_2\right) / \left(1 + \mu^2\right). \tag{38}$$

In case of $\mathbf{z}_T = \mathbf{R}_T - \mathbf{F}(\mathbf{R})_T = \mathbf{0}$ the gradient is well defined in (27). Otherwise it is quite easy to see that $a = b = 1$, $c = 0$, $d = \dot{g}(\lambda_1/\omega)$ and again $\nabla \mathbf{m}_\omega(\mathbf{S})$ is well defined.

4.2. Smoothed $\mathbf{F}(\mathbf{R})$

The smoothed $\mathbf{F}(\mathbf{R})$ reads

$$\mathbf{F}_\omega(\mathbf{U}) = \begin{bmatrix} \dots \\ \mathbf{U}_{\alpha T} \\ U_{\alpha N} + \mu_\alpha\left(\sqrt{\langle \mathbf{U}_{\alpha T}, \mathbf{U}_{\alpha T}\rangle + \omega^2} - \omega\right) \\ \dots \end{bmatrix} \tag{39}$$

so that $\nabla \mathbf{F}_\omega$ is well defined for any $\omega > 0$ and the smoothed norm $\|\mathbf{U}_T\|_\omega = \sqrt{\langle \mathbf{U}_T, \mathbf{U}_T\rangle + \omega^2} - \omega = 0$ for $\mathbf{U}_T = \mathbf{0}$.

5. Constraints satisfaction measure

For $\mathbf{R} \in K$ the vector function $\mathbf{C}(\mathbf{R}) = \mathbf{F}(\mathbf{R}) + \mathbf{m}(\mathbf{R} - \mathbf{F}(\mathbf{R}))$ represents the relative velocity at contact points, which is not compatible with the Signorini-Coulomb frictional contact law. It is therefore possible to define a measure of constraints satisfaction, that has some physical interpretation. Because $\mathbf{C}$ is expressed in terms of velocity, the following quantity

$$g(\mathbf{R}) = \sum_{\alpha=1}^{\nu} \left\langle \mathbf{W}_{\alpha\alpha}^{-1} \mathbf{C}_\alpha(\mathbf{R}), \mathbf{C}_\alpha(\mathbf{R}) \right\rangle / \sum_{\alpha=1}^{\nu} \left\langle \mathbf{W}_{\alpha\alpha}^{-1} \mathbf{B}_\alpha, \mathbf{B}_\alpha \right\rangle \tag{40}$$

approximately measures the amount of spurious energy, due to an inexact satisfaction of constraints. The denominator corresponds to the kinetic energy of the relative free motion. $\mathbf{W}_{\alpha\alpha}$ are the diagonal, 3×3, positive definite blocks of $\mathbf{W}$. The above formula has already been used in [3]. It can be used to compare the accuracy of solutions obtained by means of different numerical approaches. The merit function $g(\mathbf{R})$ is used in Section 7 as a termination criterion for the projected quasi-Newton scheme. This is one among many possible choices (not necessarily the best). Selecting a good convergence criterion for ill-conditioned problems that are often generated by multibody frictional contact formulations remains an open issue. Other termination criteria has been discussed for example in [14, 15].

6. Linearization

When linearizing the problem it is useful to think of $\mathbf{U}$ and $\mathbf{R}$ as independent and write

$$\mathbf{C}_\omega(\mathbf{U}, \mathbf{R}) = \mathbf{F}_\omega(\mathbf{U}) + \mathbf{m}_\omega(\mathbf{R} - \mathbf{F}_\omega(\mathbf{U})), \mathbf{U} = \mathbf{W}\mathbf{R} + \mathbf{B}, \tag{41}$$

We then have

$$d\mathbf{C}_\omega = \mathbf{X}d\mathbf{U} + \mathbf{Y}d\mathbf{R} \tag{42}$$

where $\mathbf{X}$ and $\mathbf{Y}$ are 3×3-block diagonal matrices

$$\mathbf{X} = [\mathbf{I} - \partial\mathbf{m}_\omega/\partial\mathbf{S}]\,\partial\mathbf{F}_\omega/\partial\mathbf{U} \tag{43}$$

$$\mathbf{Y} = \partial\mathbf{m}_\omega/\partial\mathbf{S} \tag{44}$$

and $\mathbf{I}$ is identity. The linearization reads

$$\mathbf{U} = \mathbf{W}\mathbf{R} + \mathbf{B} \tag{45}$$

$$\mathbf{C}_\omega(\mathbf{U}, \mathbf{R}) + \mathbf{X}d\mathbf{U} + \mathbf{Y}d\mathbf{R} = 0 \tag{46}$$

$$d\mathbf{U} = \mathbf{W}d\mathbf{R}. \tag{47}$$

By eliminating $\mathbf{U}$ and $d\mathbf{U}$ we come back to $\mathbf{C}_\omega(\mathbf{R}) = \mathbf{C}_\omega(\mathbf{W}\mathbf{R} + \mathbf{B}, \mathbf{R})$ and see that in the linearization of $\mathbf{C}_\omega(\mathbf{R}) = \mathbf{0}$,

$$\mathbf{C}_\omega(\mathbf{R}) + \nabla\mathbf{C}_\omega(\mathbf{R})\,d\mathbf{R} = 0, \tag{48}$$

the gradient reads

$$\nabla\mathbf{C}_\omega(\mathbf{R}) = \mathbf{X}\mathbf{W} + \mathbf{Y}. \tag{49}$$

It is noteworthy, that when using an iterative solver $\nabla\mathbf{C}_\omega$ does not need to be fully assembled. On the other hand it is clear that $\nabla\mathbf{C}_\omega$ and $\mathbf{W}$ have the same pattern of sparsity.

7. Projected quasi-Newton method

Our idea is to employ the following projected quasi-Newton step

$$\mathbf{R}^{k+1} = \mathrm{proj}_K \left[\mathbf{R}^k - \mathbf{A}^{-1} \mathbf{C}_\omega \left(\mathbf{R}^k \right) \right] \tag{50}$$

where

$$\mathbf{A} \simeq \nabla \mathbf{C}_\omega \left(\mathbf{R}^k \right) \tag{51}$$

is an approximation of $\nabla \mathbf{C}_\omega$. Since in many practical situations $\nabla \mathbf{C}_\omega$ is singular (locally or globally there may be more constraints than kinematic freedom), we cannot hope for directly employing $\nabla \mathbf{C}_\omega$. The projection is required by (15), so that the iterates remain within the friction cone. For the sake of simplicity the only strategy that we wish to use to achieve a global convergence is the combination of the projection and a suitable choice of $\mathbf{A}$.

Perhaps the simplest regularization of $\nabla \mathbf{C}_\omega$ is a diagonal perturbation

$$\mathbf{A} = \nabla \mathbf{C}_\omega + \delta \mathbf{I} \tag{52}$$

where $\delta > 0$. After combining it with an approximate, iterative inversion of $\mathbf{A}$, we end up with the following projected quasi-Newton step

$$\mathbf{R}^{k+1} = \mathrm{proj}_K \left[\mathbf{R}^k - \left(\nabla \mathbf{C}_\omega^k + \delta \mathbf{I} \right)^{-1}_{\mathrm{GMRES}(\epsilon \| \mathbf{C}_\omega^k \|, m)} \mathbf{C}_\omega^k \right] \tag{53}$$

where by $\mathbf{C}_\omega^k$ and $\nabla \mathbf{C}_\omega^k$ we mean the values taken at $\mathbf{R}^k$. In more detail the above formula is presented as Algorithm 1. In step (a) we update the velocity $\mathbf{U}^k$. This is needed in order to compute $\mathbf{C}_\omega^k$ and $\nabla \mathbf{C}_\omega^k$ in step (b). The GMRES solver runs with the absolute accuracy $\epsilon \left\| \mathbf{C}_\omega^k \right\|$ and the iterations bound m in step (c). Importantly, GMRES is preconditioned with inverted diagonal 3×3 blocks of $\mathbf{A}$. Finally the projection step follows in (d). We note, that the parallel communication is required in steps (a) and (c) for the matrix-vector products, and in the last line of the algorithm to add up values of $g\left(\mathbf{R}^k\right)$ from different processors.

No numerical scheme is 100% robust and in the last line of Algorithm 1 we fall back on the Gauss-Seidel scheme [3] when Newton iterations fail. Quality of contact solution often depends on the quality of contact points and it is not uncommon that ill-conditioned or contradictory contact points are produced in multibody simulations. In such cases the Gauss-Seidel scheme generates a good approximate solution, while the Newton solver may have terminated rather far off.

We do not attempt to analyze the convergence of the presented scheme. From our experience, proofs of global convergence for nonmonotone problems ($\mu > 0$) rarely bear relevance to numerical practice. On the other hand, for well behaved problems ($\mu = 0$) almost all methods work, be it smooth or nonsmooth. For completeness we mention that the globalization strategy adopted here has been analyzed under the name of pseudo-transient continuation [16].

Algorithm 1 The projected quasi-Newton method.

PQN($\mathbf{R}, \gamma, n, \omega, \delta, m, \epsilon$):

1. $\triangle\mathbf{R}^0 = \mathbf{0}$, $k = 0$.
2. Do
   a) $\mathbf{U}^k = \mathbf{W}\mathbf{R}^k + \mathbf{B}$.
   b) Compute $\mathbf{C}_\omega^k$ and $\mathbf{A}^k = \nabla\mathbf{C}_\omega^k + \delta\mathbf{I}$.
   c) $\triangle\mathbf{R}^k = \left(\mathbf{A}^k\right)^{-1}_{\mathrm{GMRES}(\epsilon\|\mathbf{C}_\omega^k\|,m)} \mathbf{C}_\omega^k$.
   d) $\mathbf{R}^{k+1} = \mathrm{proj}_K\left[\mathbf{R}^k - \triangle\mathbf{R}^k\right]$.
   e) $k = k + 1$.

   while $g\left(\mathbf{R}^k\right) \geq \gamma$ and $k < n$.
3. If $k = n$ then use the Gauss-Seidel solver from [3].

γ	n	ω	δ	m	ϵ	GMRES restart	MV products
1E-8	1000	1E-3 $\cdot\ \gamma$	0	10	0.25	20	100000

Table 1: Default parameter values for Algorithm 1. GMRES restart corresponds to the fixed size of the Krylov subspace used by the GMRES solver. MV products is the upper bound on the total number of matrix-vector products performed by GMRES.

8. Examples

Two examples are presented: a rotating drum of rigid or pseudo-rigid ellipsoids and a sine sweep excitation of stacked finite element cubes. In the examples, when not mentioned otherwise, default parameters from Table 1 are employed in Algorithm 1.

8.1. Rotating drum of rigid and pseudo-rigid ellipsoids

This example illustrates a motion of particles inside of a rotating drum. The drum geometry is depicted in Figure 1 (a): the drum is made out of a hollow pipe with inner radius of 1.0m and thickness 0.05m, spanned between points $(0, 0, 0)$ and $(0, 0.5, 0)$, and subdivided into 32 flat pieces. There are two rectilinear blocks, centered at points $(-0.9, 0.25, 0)$ and $(0.9, 0.25, 0)$, of size $0.2\times0.5\times0.2$ along the x, y, z directions. The drum is modeled as an obstacle - meaning that its mass is not contributing to the simulated physics of the problem. Two side walls are used to prevent falling out of particles (disk shaped obstacles, not shown in the figure). There are two stages of the simulation: dropping down the particles and rotating the drum. Figure 1 (b) illustrates the particle dropping stage: the user specifies the number of particles N and the radii of the particles

is determined as follows:

$$r_0 = (0.1/N)^{1/3},\ r_x = 0.9r_0, r_y = 0.5r_0, r_z = r_0,\ r_1 = \frac{r_x + r_y + r_z}{3}$$

where r_x, r_y, r_z are the radii of ellipsoidal particles and r_1 is a radii of spherical particles (when used below). Particles are first dropped down until all N of them are inserted into the simulation. The drum begins to rotate next. The total simulation time is 10 seconds and the time step is 0.001 second (unless specified otherwise). The particle material parameters are $E = 1\text{E}6\text{Pa}$, $\mu = 0.25$, $\rho = 100\text{kg/m}^3$ for the Young modulus, Poisson ratio and mass density, respectively. Gravity is $(0, 0, -10)$. The angular velocity of rotation is $(0, 1, 0)$ rad/s (unless specified otherwise). The Signoroni-Coulomb material model [2] is used and the coefficient of friction is 0.3. Two kinematic models are used in this example: rigid and pseudo-rigid [17]. In the latter case the rotation matrix of the rigid model is replaced by a 3×3 linear transformation (per-body deformation gradient) and hence the ellipsoids deform linearly (develop constant strain/stress). When labeling results reported below by *ell. rigid* we mean simulations involving rigid ellipsoids, while by *ell. defo* we mean simulations involving pseudo-rigid ellipsoids; *sph. rigid* and *sph. defo* are used analogously when spherical particles are used instead. The elliptical pseudo-rigid particles (*ell. defo*) actually deform: their geometry is updated based on the current deformation gradient. The spherical pseudo-rigid particles (*shp. defo*) are updated in a simplified manner: their initial spherical shape is retained.

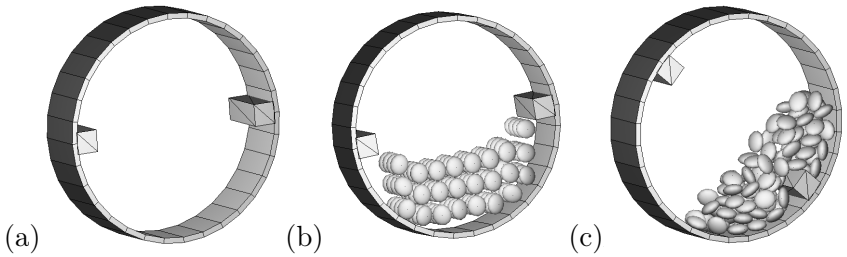


Figure 1: Rotating drum: drum geometry and simulation stages: (a) empty drum, (b) dropping down the particles, (c) rotating the drum.

8.1.1. Sensitivity study: absolute δ values

In this section the number of particles $N = 100$. Figures 2 and 3 depict sensitivity of the solver iteration counts to the choice of δ, m, and ϵ. In case (a) the linear solver iterations bound $m = 1000$ and hence the linear solve accuracy is dictated by the choice of ϵ. In this case generally, the more inexact the linear solve is the more robust the solver behavior becomes: both the total and average numbers of iterations are lowest for $\epsilon \in \{0.1, 0.25\}$; for the rigid body case, with $\delta = 0.001$, the are no cases of divergence for these two values of ϵ; in the deformable case the entire vertical strip at $\delta = 0.01$ has this property and also there is no divergence for several other combinations of δ and ϵ. In case (b) the absolute accuracy of the linear solver is kept constant at $\epsilon = 0.001$

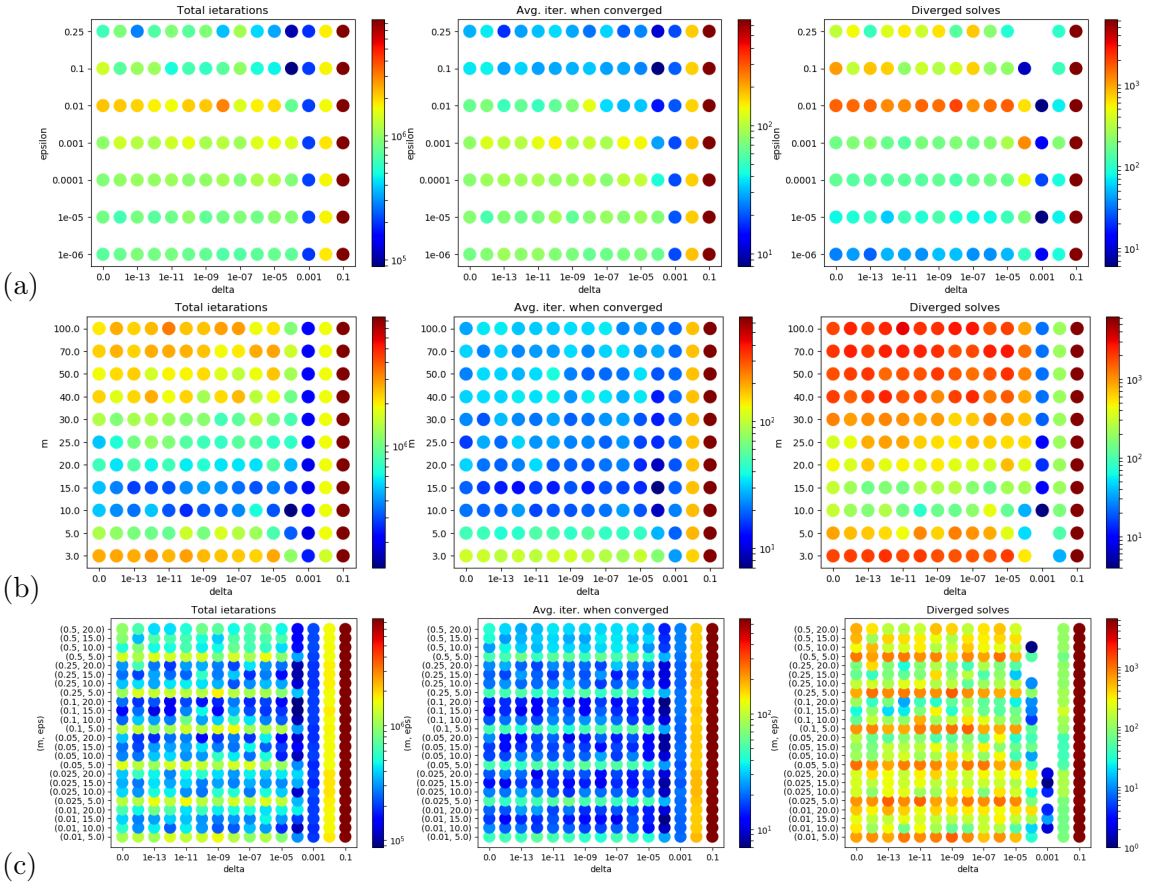


Figure 2: Rotating drum ($N = 100$, *ell. rigid*): sensitivity to the choice of δ, m, and ϵ; case (a) is for $m = 1000$ and varying δ, ϵ; case (b) is for $\epsilon = 10^{-3}$ and varying m, δ; case (c) is for varying (ϵ, m) pairs and δ; left to right we have: total iterations counts across the entire simulation, average numbers of solver iterations per time step when iterations are converged, and total (per simulation) number of diverged Newton solver runs (when the Gauss-Seidel solver was invoked upon reaching the iteration limit); missing dots in the rightmost figures correspond to no cases of divergence.

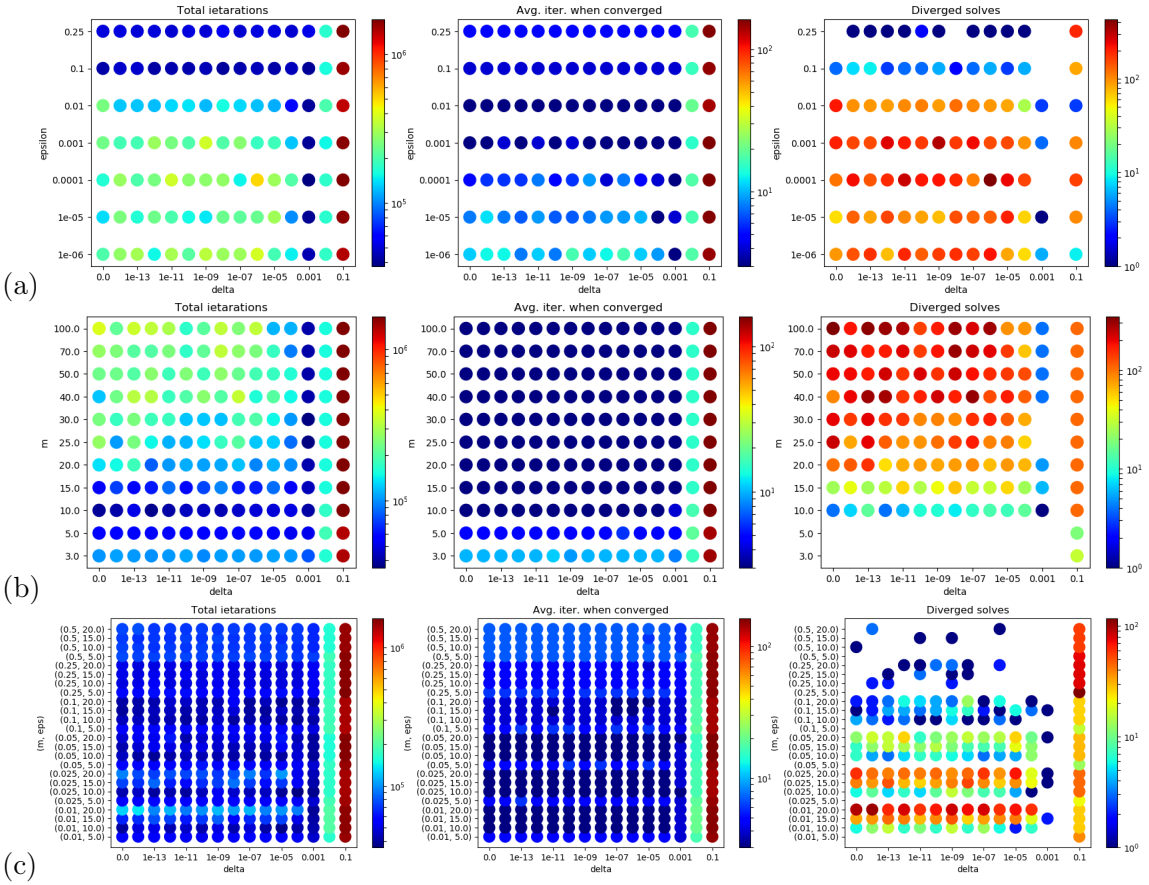


Figure 3: Rotating drum ($N = 100$, *ell. defo*): sensitivity to the choice of δ, m, and ϵ; sensitivity to the choice of δ, m, and ϵ; case (a) is for $m = 1000$ and varying δ, ϵ; case (b) is for $\epsilon = 10^{-3}$ and varying m, δ; case (c) is for varying (ϵ, m) pairs and δ; left to right we have: total iterations counts across the entire simulation, average numbers of solver iterations per time step when iterations are converged, and total (per simulation) number of diverged Newton solver runs (when the Gauss-Seidel solver was invoked upon reaching the iteration limit); missing dots in the rightmost figures correspond to no cases of divergence.

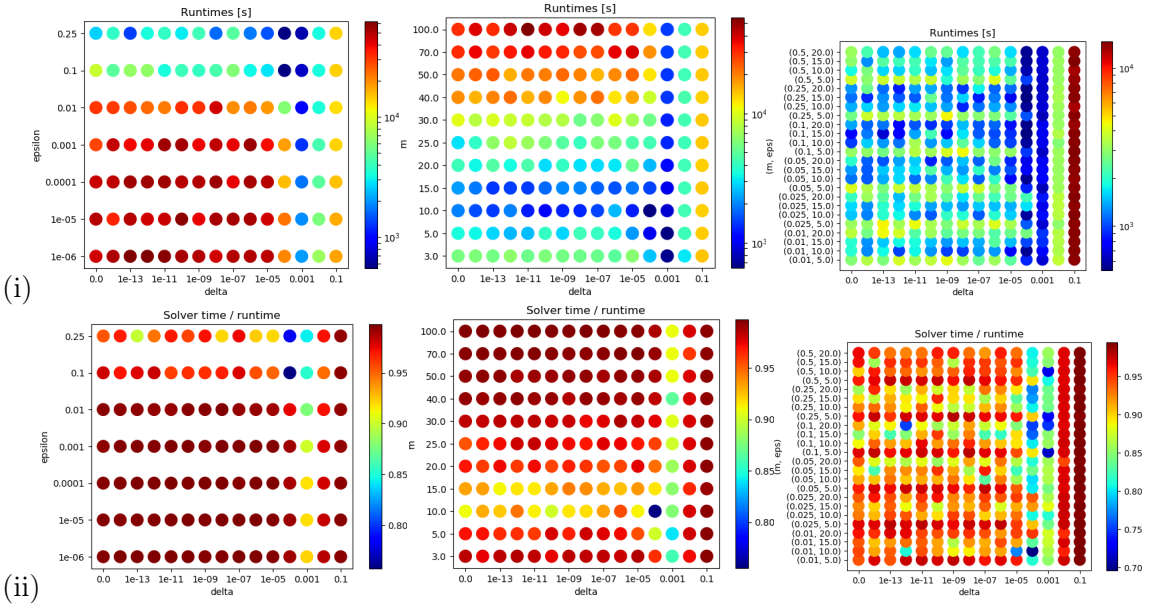


Figure 4: Rotating drum ($N = 100$, *ell. rigid*): runtimes (i) and solver time to runtime ratios (ii); left to right figures correspond to cases (a-c) in Figure 2.

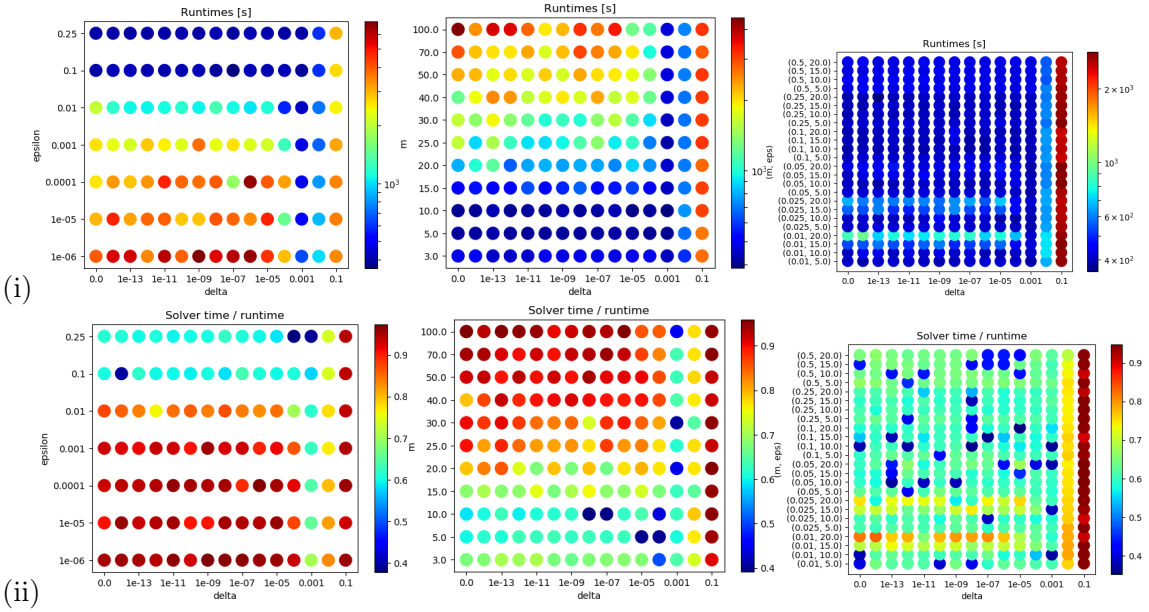


Figure 5: Rotating drum ($N = 100$, *ell. defo*): runtimes (i) and solver time to runtime ratios (ii); left to right figures correspond to cases (a-c) in Figure 3.

	Newton solver based				Gauss-Seidel solver based			
	runtime	ratio	it/step	divs	runtime	ratio	it/step	divs
ell. rigid	525s	0.78	10	0	467s	0.84	112	20
ell. defo	356s	0.64	5	0	463s	0.48	18	0
sph. rigid	257s	0.93	12	1	440s	0.96	174	330
sph. defo	217s	0.89	6	44	66s	0.63	21	0

Table 2: Rotating drum ($N = 100$): comparison of solution statistics for (δ, ϵ, m) choice of (0.0001, 0.25, 15) for the rigid body case and (1E-13, 0.25, 20) for the deformable body case; respective columns stand for total runtime in seconds (runtime), solver time to runtime ratio (ratio), average number of iterations per step when converged (it/step), and number of diverged steps (divs).

while the number of linear solver iterations m is varied. In this case it is also clear that inexact solves are favorable, with a lowest total number of iterations for $m \in \{10, 15\}$ for both the rigid and deformable case. Quite clearly: less exact (but not too inexact) linear solves are favored in terms of producing a lower overall iteration counts. Case (c) samples sensitivity in the space of various (ϵ, m) choices and the varying regularization parameter δ. In the rigid case (Figure 2 (c), right) it is seen that the entire $\delta = 0.001$ is nearly free of solver divergence; a significantly large parameter space has this property in the deformable case (Figure 3 (c), right). We can also see that for the rigid case the average number of iterations per time step is in the range 10-100, while in the deformable case it is in the range 1-10. Consequently, there is a similar correspondence between the total number of iterations (scaled up by 10000: the number of time steps).

The runtimes and solver time to runtime ratios are depicted in Figure 4 and 5. Shortest runtimes and lowest timing ratios correspond to the same parameter set choices as those transpiring in Figures 2 and 3. For the rigid body case, the choice of (0.0001, 0.25, 15) for the (δ, ϵ, m) triad produces the shortest total runtime of 525s (with zero solver failures). For the pseudo-rigid kinematic case, the choice of (1E-13, 0.25, 20) for the (δ, ϵ, m) triad produces the shortest total runtime of 356s (with zero solver failures). These runtimes are well comparable with those produced by the Gauss-Seidel solver based runs and reported in Table 2.

8.1.2. Sensitivity study: relative δ values

The choice of the regularization parameter δ as an absolute value may not always be practical. For simulation setups with many particles it may not be possible to run many large scale tests in order to tune δ. It may be more practical to study solver performance on a smaller example, just like in this section, and then attempt to reuse the set of recommended parameters for larger scale runs. This is why investigating a choice of δ as relative to a norm of the $\mathbf{W}$ operator (7) is relevant. Three such choices are investigated here:

$$\delta_1 = \delta \cdot \min_i W_{ii} \text{ or } \delta_1 = \delta \cdot \text{avg}_i W_{ii} \text{ or } \delta_1 = \delta \cdot \max_i W_{ii}$$

where W_{ii} are the diagonal entries of $\mathbf{W}$. Consequently, δ_1 is used in line 2b of Algorithm 1 instead of δ. The same studies as in Section 8.1.1 are repeated in the current case. The results are juxtaposed in Figures 6-11 for the rigid and deformable cases and for the three variants of δ_1 choice.

By browsing Figures 6-11 we notice that cases of fully robust runs (without divergence) are significantly more common when the choice of δ_1 is relative to $\mathbf{W}$. As we move from the $\min W_{ii}$ norm, through the average W_{ii} norm, to the $\max W_{ii}$ norm, the region of no solver failures gradually moves from right to left – within the unchanging range of the relative span of δ. This reflects the fact that the absolute range of values of δ_1 for which no failures occur becomes better represented (within the chosen and fixed range of δ) as we shift between $\min W_{ii} \rightarrow \text{avg} W_{ii} \rightarrow \max W_{ii}$. Perhaps it does not matter which of those norms we use as long as in our investigation of an optimum value of δ we cover a sufficient range. We can also see, that for rigid ellipsoids this range is narrower when compared with the deformable counterpart: the no failure zone is significantly broader in case of pseudo-rigid ellipsoids. This corresponds to the fact that it is easier to resolve contact reactions with the extra kinematic freedom allowing for the particles themselves to deform.

In all of the Figures 6-11 the areas of shortest runtimes do not necessarily strictly correspond to the areas of no solver failure (at least visually). There are overlap zones where both properties meet, but in general shortest runtimes occur in areas where both, the average number of solver iterations per time step, and the number of solver failures, are simultaneously low. Having only few instances of a fallback on the Gauss-Seidel solver during the entire simulation does not significantly affect the runtime. Nonetheless, it is possible to select parameter sets for which there are no solver failures and the runtime is short. For the rigid body case, the choice of $(0.01_{\text{avg}W_{ii}}, 0.25, 10)$ for the $(\delta_{\text{avg}W_{ii}}, \epsilon, m)$ triad produces the shortest total runtime of 467s (with zero solver failures). For the pseudo-rigid kinematic case, the choice of $(0.01_{\text{avg}W_{ii}}, 0.01, 5)$ for the $(\delta_{\text{avg}W_{ii}}, \epsilon, m)$ triad produces the shortest total runtime of 352s (with zero solver failures). These runtimes are well comparable with those produced by the Gauss-Seidel solver based runs and reported in Table 3. Table 4 juxtaposes shortest runtimes for all three choices of the relative δ norms.

8.1.3. Sensitivity study: material density vs. relative δ choices

Small scale simulations like in Sections 8.1.1 and 8.1.2 are suitable for exploring and tuning solver parameters. Nonetheless, in case of large scale simulations, such direct tuning would not be practical - consuming too much resources and time. When modifying aspects of a simulation setup (e.g. material parameters, geometrical dimensions, number of particles) it is useful to have an orientation whether a set of optimal parameters - for a tuned simulation setup - remains approximately optimal for a model that has undergone change. Here, we explore this by varying material density versus the choice of delta for

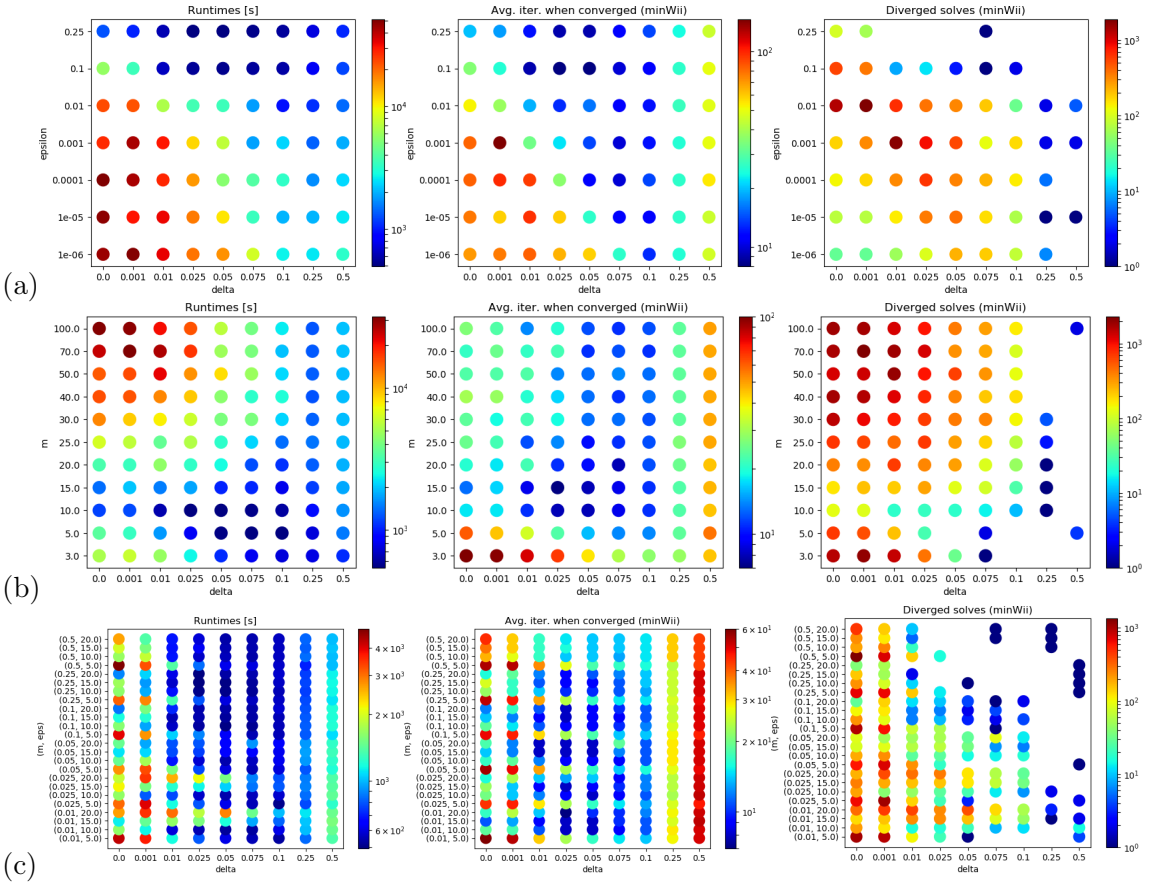


Figure 6: Rotating drum ($N = 100$, *ell. rigid*, $\delta_1 = \delta \cdot \min_i W_{ii}$): sensitivity to the choice of δ, m, and ϵ; case (a) is for $m = 1000$ and varying δ, ϵ; case (b) is for $\epsilon = 10^{-3}$ and varying m, δ; case (c) is for varying (ϵ, m) pairs and δ; left to right we have: total runtimes, average numbers of solver iterations per time step when iterations are converged, and total (per simulation) number of diverged Newton solver runs (when the Gauss-Seidel solver was invoked upon reaching the iteration limit); missing dots in the rightmost figures correspond to no cases of divergence.

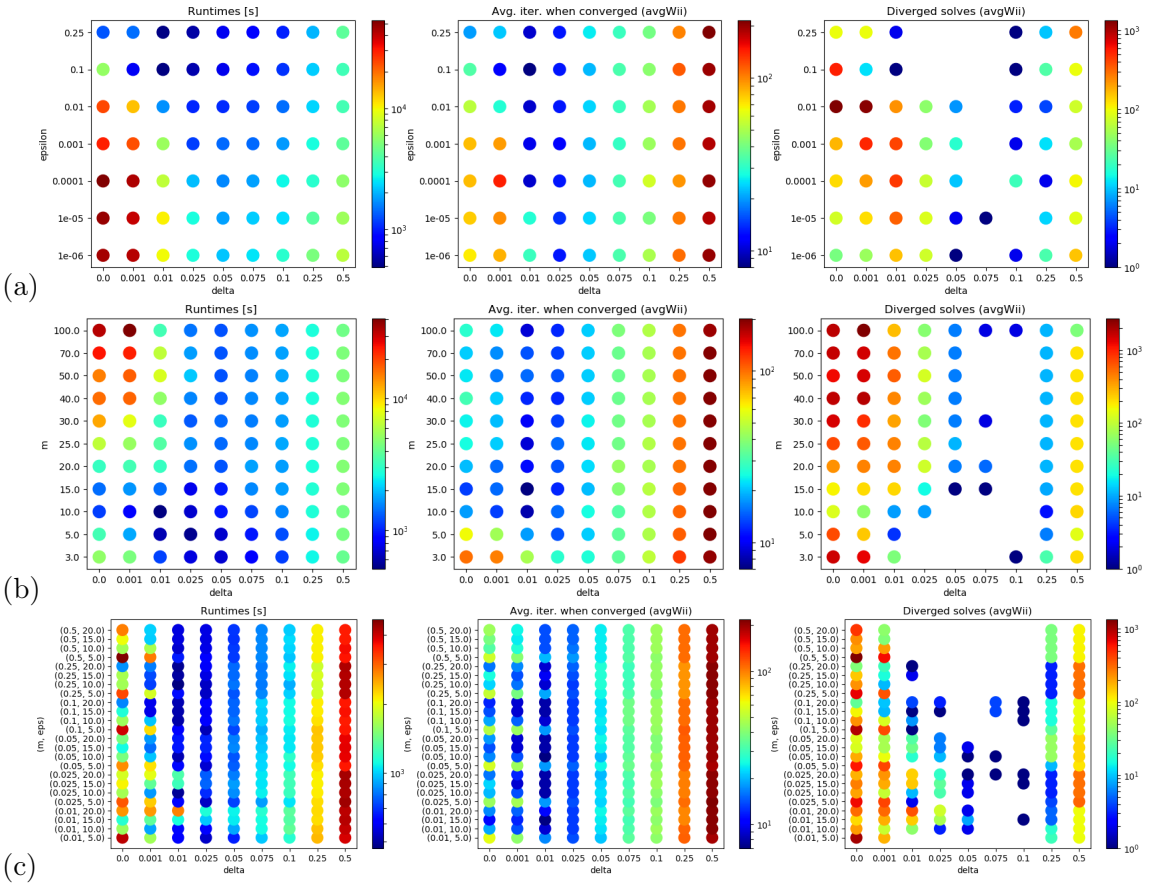


Figure 7: Rotating drum ($N = 100$, *ell. rigid*, $\delta_1 = \delta \cdot \underset{i}{\text{avg}} W_{ii}$): sensitivity to the choice of δ, m, and ϵ; case (a) is for $m = 1000$ and varying δ, ϵ; case (b) is for $\epsilon = 10^{-3}$ and varying m, δ; case (c) is for varying (ϵ, m) pairs and δ; left to right we have: total runtimes, average numbers of solver iterations per time step when iterations are converged, and total (per simulation) number of diverged Newton solver runs (when the Gauss-Seidel solver was invoked upon reaching the iteration limit); missing dots in the rightmost figures correspond to no cases of divergence.

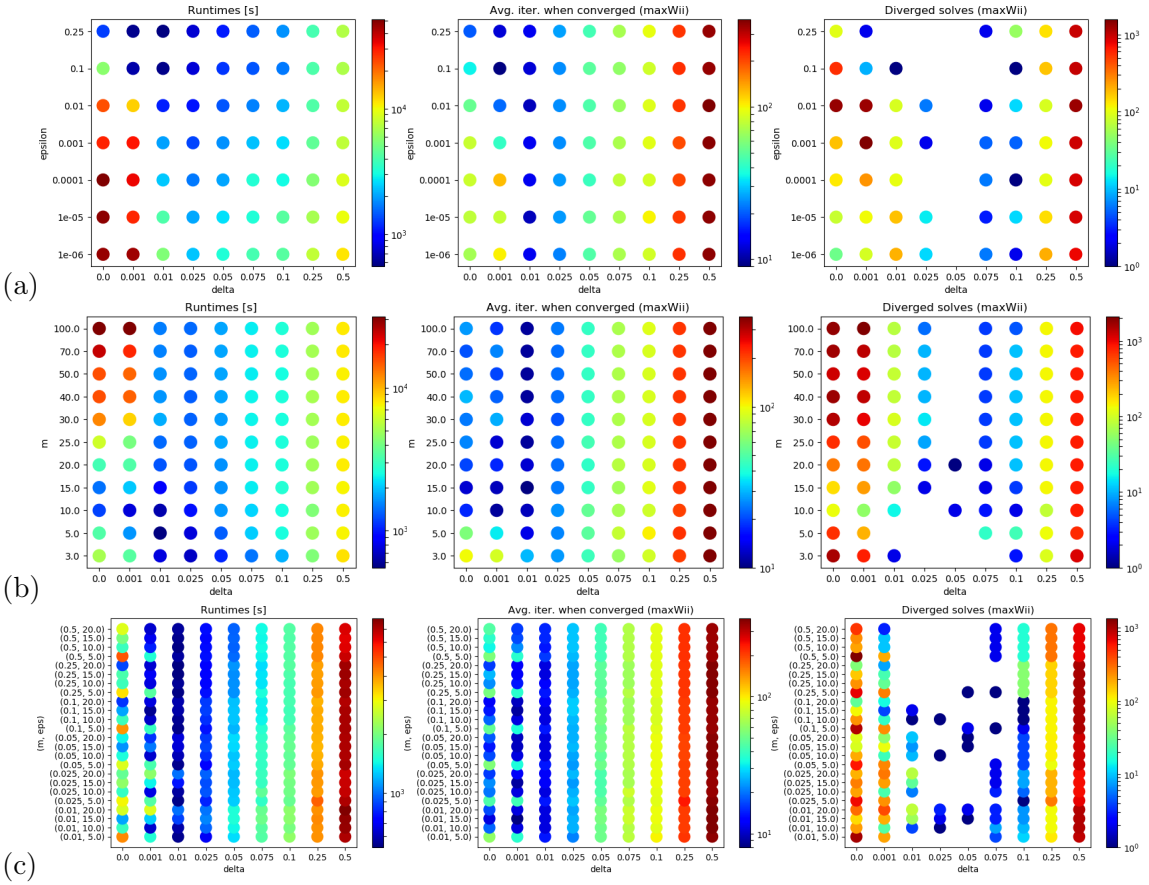


Figure 8: Rotating drum ($N = 100$, *ell. rigid*, $\delta_1 = \delta \cdot \max_i W_{ii}$): sensitivity to the choice of δ, m, and ϵ; case (a) is for $m = 1000$ and varying δ, ϵ; case (b) is for $\epsilon = 10^{-3}$ and varying m, δ; case (c) is for varying (ϵ, m) pairs and δ; left to right we have: total runtimes, average numbers of solver iterations per time step when iterations are converged, and total (per simulation) number of diverged Newton solver runs (when the Gauss-Seidel solver was invoked upon reaching the iteration limit); missing dots in the rightmost figures correspond to no cases of divergence.

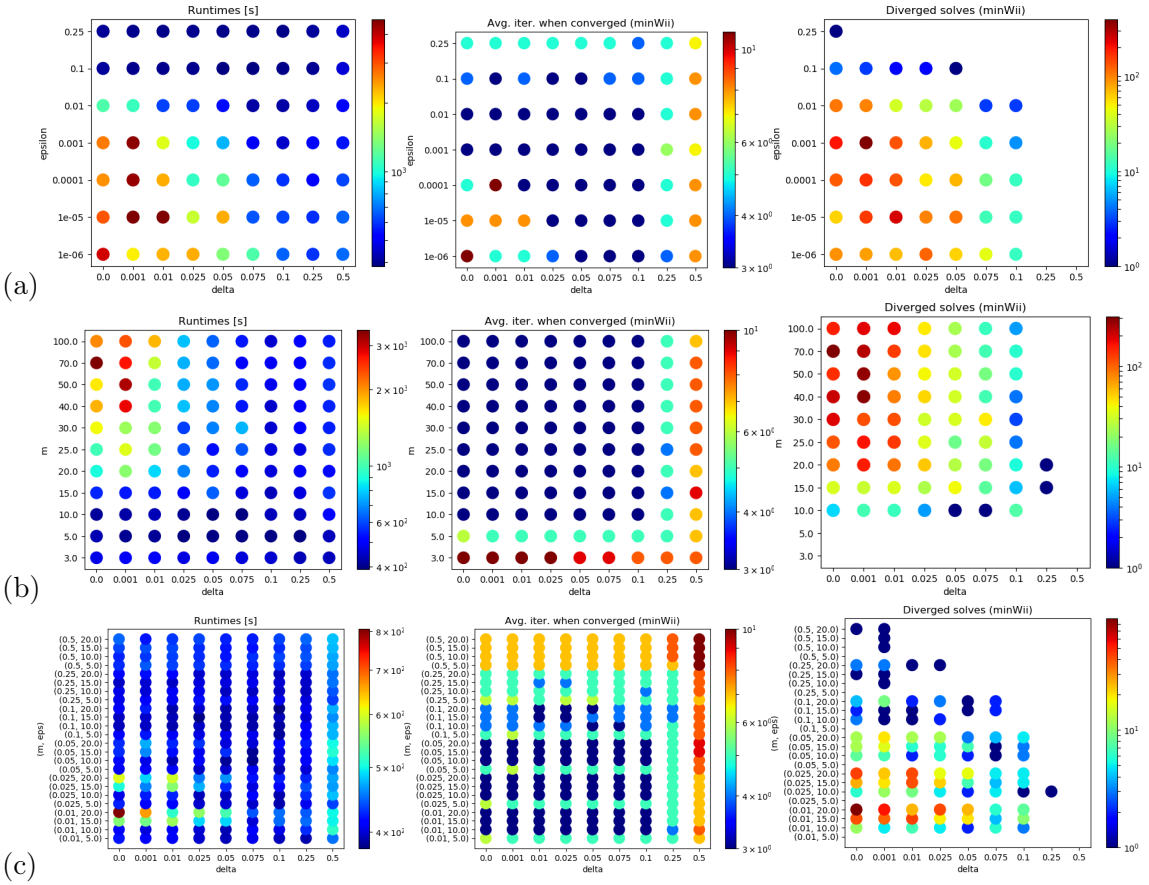


Figure 9: Rotating drum ($N = 100$, *ell. defo*, $\delta_1 = \delta \cdot \min_i W_{ii}$): sensitivity to the choice of δ, m, and ϵ; case (a) is for $m = 1000$ and varying δ, ϵ; case (b) is for $\epsilon = 10^{-3}$ and varying m, δ; case (c) is for varying (ϵ, m) pairs and δ; left to right we have: total runtimes, average numbers of solver iterations per time step when iterations are converged, and total (per simulation) number of diverged Newton solver runs (when the Gauss-Seidel solver was invoked upon reaching the iteration limit); missing dots in the rightmost figures correspond to no cases of divergence.

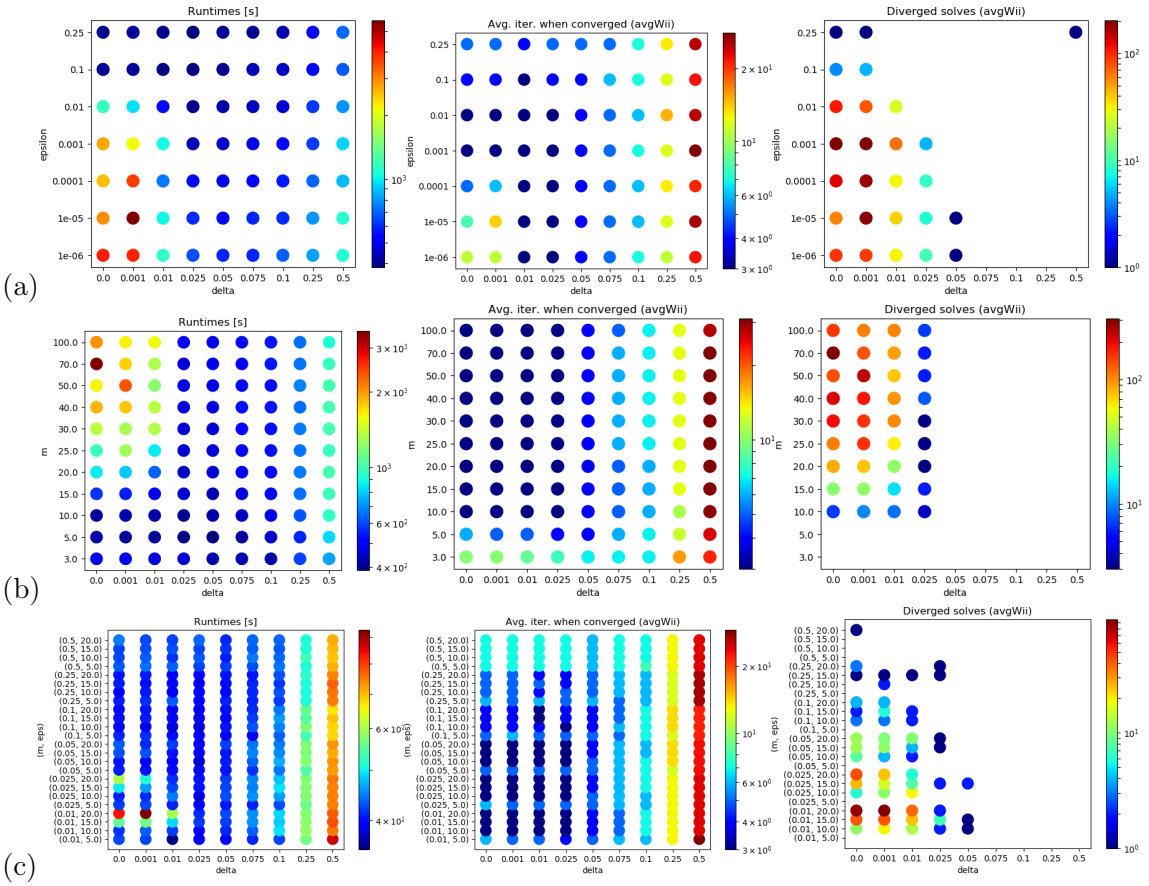


Figure 10: Rotating drum ($N = 100$, *ell. defo*, $\delta_1 = \delta \cdot \underset{i}{\text{avg}} W_{ii}$): sensitivity to the choice of δ, m, and ϵ; case (a) is for $m = 1000$ and varying δ, ϵ; case (b) is for $\epsilon = 10^{-3}$ and varying m, δ; case (c) is for varying (ϵ, m) pairs and δ; left to right we have: total runtimes, average numbers of solver iterations per time step when iterations are converged, and total (per simulation) number of diverged Newton solver runs (when the Gauss-Seidel solver was invoked upon reaching the iteration limit); missing dots in the rightmost figures correspond to no cases of divergence.

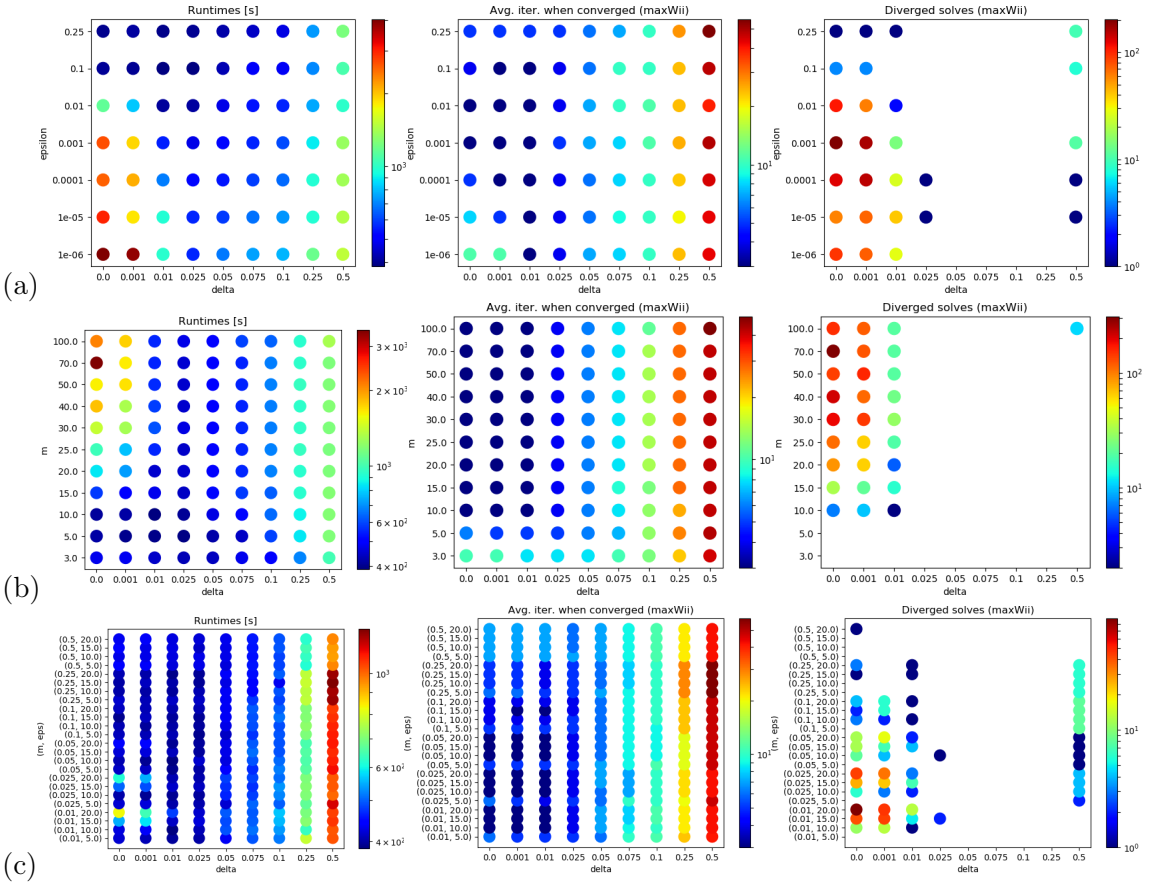


Figure 11: Rotating drum ($N = 100$, *ell. defo*, $\delta_1 = \delta \cdot \max_i W_{ii}$): sensitivity to the choice of δ, m, and ϵ; case (a) is for $m = 1000$ and varying δ, ϵ; case (b) is for $\epsilon = 10^{-3}$ and varying m, δ; case (c) is for varying (ϵ, m) pairs and δ; left to right we have: total runtimes, average numbers of solver iterations per time step when iterations are converged, and total (per simulation) number of diverged Newton solver runs (when the Gauss-Seidel solver was invoked upon reaching the iteration limit); missing dots in the rightmost figures correspond to no cases of divergence.

	Newton solver based				Gauss-Seidel solver based			
	runtime	ratio	it/step	divs	runtime	ratio	it/step	divs
ell. rigid	467s	0.77	9	0	495s	0.85	112	20
ell. defo	352s	0.65	5	0	482s	0.45	18	0
sph. rigid	272s	0.93	14	0	462s	0.96	174	330
sph. defo	135s	0.82	4	0	64s	0.64	21	0

Table 3: Rotating drum ($N = 100$): comparison of solution statistics for (δ, ϵ, m) choice of $(0.01_{\text{avg}W_{ii}}, 0.25, 10)$ for the rigid body case and $(0.01_{\text{avg}W_{ii}}, 0.01, 5)$ for the deformable body case; respective columns stand for total runtime in seconds (runtime), solver time to runtime ratio (ratio), average number of iterations per step when converged (it/step), and number of diverged steps (divs).

		runtime	δ	ϵ	m
rigid	$\min W_{ii}$	495s	0.025	0.25	15
	$\text{avg}W_{ii}$	467s	0.01	0.25	10
	$\max W_{ii}$	525s	0.01	0.25	20
defo	$\min W_{ii}$	378s	0.1	0.1	20
	$\text{avg}W_{ii}$	352s	0.01	0.01	5
	$\max W_{ii}$	385s	0.01	0.05	10

Table 4: Rotating drum ($N = 100$): juxtaposition of shortest runtimes and parameter choices for all three relative δ norms.

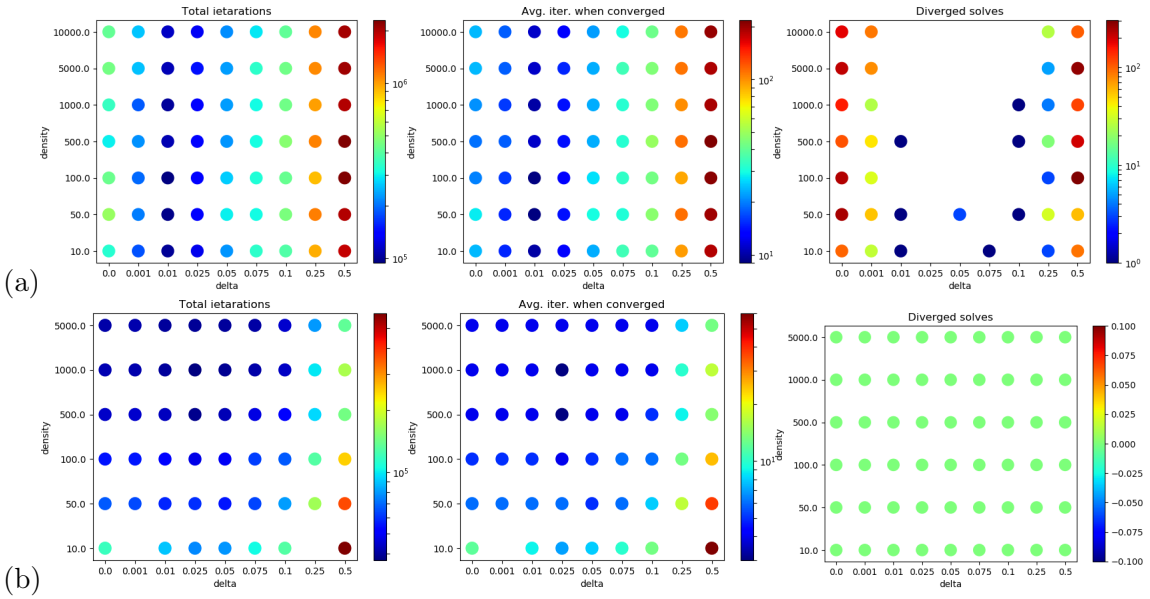
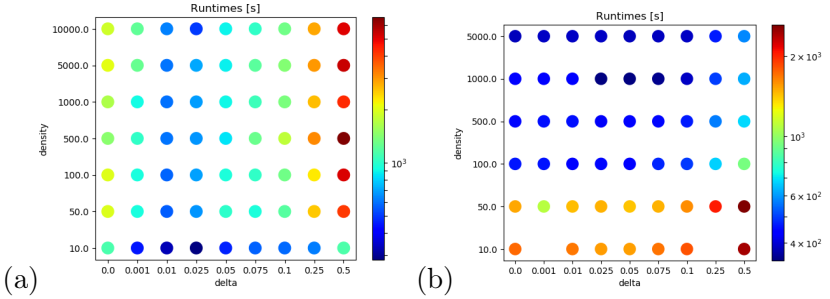


Figure 12: Rotating drum ($N = 100$, $(\delta_1, \epsilon, m) = (\delta_{\text{avg}W_{ii}}, 0.25, 10)$ for *ell. rigid*, $(\delta_1, \epsilon, m) = (\delta_{\text{avg}W_{ii}}, 0.01, 5)$ for *ell. defo*): sensitivity to the choice of material density and δ; case (a) is for rigid ellipsoids; case (b) is for pseudo-rigid ellipsoids; left to right we have: total runtimes, average numbers of solver iterations per time step when iterations are converged, and total (per simulation) number of diverged Newton solver runs (when the Gauss-Seidel solver was invoked upon reaching the iteration limit); missing dots in the rightmost figures correspond to no cases of divergence.

the same model with 100 particles. We hope to verify whether the relative choice of delta is robust enough to be applied for large scale simulations of the rotating drum: in our setup the dimensions of the drum remain unchanged, but particles become smaller as their number grows. Varying the density of the 100 particles algebraically approximates this effect.

Figure 12 illustrates solver statistics in the parameter space of material density and δ. For the rigid body case (a) it is clear that the optimum $\delta = 0.01$ remains constant across the range of tested density values. There is also a relatively large space of divergence free runs, as visible in the rightmost subfigure. For the deformable case on the other hand, there is some sensitivity of the optimum choice of δ to material density. This is because in case of pseudo-rigid ellipsoids there is the effect of interaction between deformability and inertia, which affects the conditioning of the problem and influences the behavior of the solver. Subfigures (b) indicate that a range of δs could be used while the optimum choices might be closer to 0.025 for larger density values. Also, there are no solver failures (as seen in the rightmost subfigure) in this case.

An analogous sensitivity of total runtimes is pictured in Figure 13. Similar observations apply in this case: $\delta = 0.01$ is clearly most suitable in the rigid case, while in the



(a) (b)

Figure 13: Rotating drum ($N = 100$, $(\delta_1, \epsilon, m) = (\delta_{\text{avg}W_{ii}}, 0.25, 10)$ for *ell. rigid*, $(\delta_1, \epsilon, m) = (\delta_{\text{avg}W_{ii}}, 0.01, 5)$ for *ell. defo*): sensitivity of the total runtime to the choice of material density and δ; case (a) is for rigid ellipsoids; case (b) is for pseudo-rigid ellipsoids.

Figure 14: Rotating drum ($N = 100k$): example snapshots of rigid body simulation with linear velocity magnitude field at times 1s, 5s and 10s.

deformable space the upper left corner of the parameter space is optimal. Interestingly, there is quite a steep ramping up of runtimes for lower material densities in the pseudo-rigid case. This illustrates the total effects of the interaction between deformation and inertia, as mentioned above.

8.1.4. Larger scale parallel runs

100k ellipsoids are now simulated using the same example in four modes: rigid and deformable, either using the Newton solver with $(\delta_1, \epsilon, m) = (0.01_{\text{avg}W_{ii}}, 0.25, 10)$ for rigid and $(\delta_1, \epsilon, m) = (0.01_{\text{avg}W_{ii}}, 0.01, 5)$ for pseudo-rigid ellipsoids, or using the Gauss-Seidel solver. In each of those four cases 1, 2, 4, 8, and 16 nodes of a parallel cluster are used, utilizing 24 CPU cores per cluster node[1] (this produces a range of MPI jobs spanning between 24 and 384 ranks). Figure 14 illustrates simulation state snapshots at times 1s, 5s and 10s, while Tables 5 and 6 summarize runtimes and solver statistics across the range of 1 to 16 cluster nodes.

[1]Cluster nodes use Intel Xeon E5-2600 V2 Ivy Bridge 2.7GHz CPUs and have 64GB of RAM. They are interconnected via 1x56Gb/s FDR InfiniBand network.

		Newton solver based					Gauss-Seidel solver based				
	Cluster nodes	1	2	4	8	16	1	2	4	8	16
Run hours	ell. rigid total										
	ell. defo total										
	ell. rigid solver										
	ell. defo solver										

Table 5: Rotating drum ($N = 100k$): comparison of total and solver runtimes and parallel scaling.

		Newton solver based				Gauss-Seidel solver based			
	Nodes	ratio	tot it	it/step	divs	ratio	tot it	it/step	divs
ell. rigid	1								
	2								
	4								
	8								
	16								
ell. defo	1								
	2								
	4								
	8								
	16								

Table 6: Rotating drum ($N = 100k$): comparison of solution statistics for varying numbers of cluster nodes; respective columns stand for solver time to runtime ratio (ratio), total iterations count (tot it), average number of iterations per step when converged (it/step), and number of diverged steps (divs).

8.2. Sine sweep excitation of stacked finite element cubes

8.2.1. Small scale serial runs

8.2.2. Larger scale parallel runs

9. Conclusions

A. Appendix - Derivative of the smoothed projection onto the polar cone

The projection operator as defined in (23) is nonsmooth. Motivated by [6, 13] we employ the following smoothing

$$\mathrm{proj}^{\omega}_{K^\circ}(\mathbf{z}) = \kappa_1 \omega g\left(\lambda_1/(\kappa_1\omega)\right)\mathbf{u}_1 + \kappa_2 \omega g\left(\lambda_2/(\kappa_2\omega)\right)\mathbf{u}_2 \tag{54}$$

where $\kappa_1 > 0$ and $\kappa_2 > 0$ and

$$g(\lambda) = \frac{1}{2}\left(\sqrt{\lambda^2+4}+\lambda\right). \tag{55}$$

The smoothing function $\omega g(\lambda/\omega)$ is differentiable for all $\omega > 0$ and it converges to $\max(0,\lambda)$ when $\omega \downarrow 0$. We derive gradient of $\mathrm{proj}^{\omega}_{K^\circ}(\mathbf{z})$ as defined in (54). This is a repetition of the derivation provided by Fukushima et al. [6] (Proposition 5.2), adjusted to the case of a non self-dual cone. To compute $\nabla \mathrm{proj}^{\omega}_{K^\circ}(\mathbf{z})$ we need to consider two cases, $\mathbf{z}_T \neq \mathbf{0}$ and $\mathbf{z}_T = \mathbf{0}$. Since there holds

$$\mathrm{proj}^{\omega}_{K^\circ}(\mathbf{z}) = \omega h(\mathbf{z}/\omega) \;\text{ and thus }\; \nabla \mathrm{proj}^{\omega}_{K^\circ}(\mathbf{z}) = \nabla h(\mathbf{z}/\omega) \tag{56}$$

where

$$h(\mathbf{z}) = \kappa_1 g\left(\lambda_1/\kappa_1\right)\mathbf{u}_1 + \kappa_2 g\left(\lambda_2/\kappa_2\right)\mathbf{u}_2, \tag{57}$$

it is enough to compute ∇h.

Case $\mathbf{z}_T \neq \mathbf{0}$

We have

$$\nabla_{\mathbf{z}}\lambda_i = \frac{1}{1+\mu^2}\mathbf{u}_1, \;\; i = 1,2 \tag{58}$$

$$\nabla_{\mathbf{z}}\mathbf{u}_1 = \frac{-\mu}{\|\mathbf{z}_T\|}\begin{bmatrix} \mathbf{I} - \mathbf{z}_T\mathbf{z}_T^T/\|\mathbf{z}_T\|^2 & 0 \\ 0 & 0 \end{bmatrix} \tag{59}$$

$$\nabla_{\mathbf{z}}\mathbf{u}_2 = \frac{1}{\|\mathbf{z}_T\|}\begin{bmatrix} \mathbf{I} - \mathbf{z}_T\mathbf{z}_T^T/\|\mathbf{z}_T\|^2 & 0 \\ 0 & 0 \end{bmatrix} \tag{60}$$

and hence

$$\begin{aligned}\nabla h\left(\mathbf{z}\right) &= \kappa_1 g\left(\lambda_1/\kappa_1\right)\nabla_{\mathbf{z}}\mathbf{u}_1 + \dot{g}\left(\lambda_1/\kappa_2\right)\mathbf{u}_1\left(\nabla_{\mathbf{z}}\lambda_1\right)^T \\ &+ \kappa_2 g\left(\lambda_2/\kappa_2\right)\nabla_{\mathbf{z}}\mathbf{u}_2 + \dot{g}\left(\lambda_2/\kappa_2\right)\mathbf{u}_2\left(\nabla_{\mathbf{z}}\lambda_2\right)^T \\ &= \frac{\kappa_2 g\left(\lambda_2/\kappa_2\right) - \mu\kappa_1 g\left(\lambda_1/\kappa_1\right)}{\left\|\mathbf{z}_T\right\|}\begin{bmatrix}\mathbf{I} - \mathbf{z}_T\mathbf{z}_T^T/\left\|\mathbf{z}_T\right\| & 0\\ 0 & 0\end{bmatrix} \\ &+ \frac{1}{1+\mu^2}\left[\dot{g}\left(\lambda_1/\kappa_1\right)\mathbf{u}_1\mathbf{u}_1^T + \dot{g}\left(\lambda_2/\kappa_2\right)\mathbf{u}_2\mathbf{u}_2^T\right].\end{aligned}$$

There hold

$$\mathbf{u}_1\mathbf{u}_1^T = \begin{bmatrix}\mu^2\mathbf{z}_T\mathbf{z}_T^T/\left\|\mathbf{z}_T\right\|^2 & \mu\mathbf{z}_T/\left\|\mathbf{z}_T\right\| \\ \mu\mathbf{z}_T^T/\left\|\mathbf{z}_T\right\| & 1\end{bmatrix} \tag{61}$$

$$\mathbf{u}_2\mathbf{u}_2^T = \begin{bmatrix}\mathbf{z}_T\mathbf{z}_T^T/\left\|\mathbf{z}_T\right\|^2 & -\mu\mathbf{z}_T/\left\|\mathbf{z}_T\right\| \\ -\mu\mathbf{z}_T^T/\left\|\mathbf{z}_T\right\| & \mu^2\end{bmatrix} \tag{62}$$

so that

$$\begin{aligned}\nabla h &= \frac{\kappa_2 g\left(\lambda_2/\kappa_2\right) - \mu\kappa_1 g\left(\lambda_1/\kappa_1\right)}{\left\|\mathbf{z}_T\right\|}\begin{bmatrix}\mathbf{I} - \mathbf{z}_T\mathbf{z}_T^T/\left\|\mathbf{z}_T\right\|^2 & 0\\ 0 & 0\end{bmatrix} \\ &\frac{1}{1+\mu^2}\begin{bmatrix}\left(\mu^2\dot{g}\left(\lambda_1/\kappa_1\right) + \dot{g}\left(\lambda_2/\kappa_2\right)\right)\mathbf{z}_T\mathbf{z}_T^T/\left\|\mathbf{z}_T\right\|^2 & \mu\left(\dot{g}\left(\lambda_1/\kappa_1\right) - \dot{g}\left(\lambda_2/\kappa_2\right)\right)\mathbf{z}_T/\left\|\mathbf{z}_T\right\| \\ \mu\left(\dot{g}\left(\lambda_1/\kappa_1\right) - \dot{g}\left(\lambda_2/\kappa_2\right)\right)\mathbf{z}_T^T/\left\|\mathbf{z}_T\right\| & \dot{g}\left(\lambda_1/\kappa_1\right) + \mu^2\dot{g}\left(\lambda_2/\kappa_2\right)\end{bmatrix}\end{aligned}$$

and by taking

$$a = \frac{\kappa_2 g\left(\lambda_2/\kappa_2\right) - \mu\kappa_1 g\left(\lambda_1/\kappa_1\right)}{\left\|\mathbf{z}_T\right\|}, b = \frac{\mu^2\dot{g}\left(\lambda_1/\kappa_1\right) + \dot{g}\left(\lambda_2/\kappa_2\right)}{1+\mu^2} \tag{63}$$

$$c = \frac{\mu\left(\dot{g}\left(\lambda_1/\kappa_1\right) - \dot{g}\left(\lambda_2/\kappa_2\right)\right)}{1+\mu^2}, d = \frac{\dot{g}\left(\lambda_1/\kappa_1\right) + \mu^2\dot{g}\left(\lambda_2/\kappa_2\right)}{1+\mu^2} \tag{64}$$

we obtain

$$\nabla h\left(\mathbf{z}\right) = \begin{bmatrix}a\mathbf{I} + \left(b-a\right)\mathbf{z}_T\mathbf{z}_T^T/\left\|\mathbf{z}_T\right\|^2 & c\mathbf{z}_T/\left\|\mathbf{z}_T\right\| \\ c\mathbf{z}_T^T/\left\|\mathbf{z}_T\right\| & d\end{bmatrix}. \tag{65}$$

We now need to ensure that ∇h is invertible, which is done along Proposition 6.1 in [6] or Proposition 3.1 in [13]. Since d is positive it is enough to show that the Schur complement of ∇h with respect to d is also positive definite. The Schur complement with respect to d reads

$$\left\{a\mathbf{I} + \left(b-a\right)\mathbf{z}_T\mathbf{z}_T^T/\left\|\mathbf{z}_T\right\|^2\right\} - c^2\mathbf{z}_T\mathbf{z}_T^T/d\left\|\mathbf{z}_T\right\|^2 = a\left(\mathbf{I} - \mathbf{z}_T\mathbf{z}_T^T/\left\|\mathbf{z}_T\right\|^2\right) + \frac{bd-c^2}{d}\mathbf{z}_T\mathbf{z}_T^T/\left\|\mathbf{z}_T\right\|^2.$$

Both $\mathbf{I}-\mathbf{z}_T\mathbf{z}_T^T/\left\|\mathbf{z}_T\right\|^2$ and $\mathbf{z}_T\mathbf{z}_T^T/\left\|\mathbf{z}_T\right\|^2$ are positive semidefinite and their sum is identity. Hence any positive linear combination of these two matrices is itself positive definite. We now need to show that $a>0$ and that $bd-c^2>0$. For $a>0$ it is necessary that

$$\kappa_2 g\left(\lambda_2/\kappa_2\right)-\mu\kappa_1 g\left(\lambda_1/\kappa_1\right)>0$$

which is the case if only

$$\lambda_2/\kappa_2>\lambda_1/\kappa_1 \text{ and } \kappa_2\geq\mu\kappa_1.$$

From (20) we notice that $\left\|\mathbf{z}_T\right\|=\lambda_2-\mu\lambda_1$. Hence $\lambda_2>\mu\lambda_1$ and above conditions are most naturally fulfilled for

$$\kappa_1=1 \text{ and } \kappa_2=\mu$$

or

$$\kappa_1=\mu^{-1} \text{ and } \kappa_2=1.$$

Let us now check how these choices affect the condition $bd-c^2>0$. We have

$$\begin{aligned}
\left(bd-c^2\right)\left(1+\mu^2\right)^2 &= \left(\mu^2\dot{g}\left(\lambda_1/\kappa_1\right)+\dot{g}\left(\lambda_2/\kappa_2\right)\right)\left(\dot{g}\left(\lambda_1/\kappa_1\right)+\mu^2\dot{g}\left(\lambda_2/\kappa_2\right)\right)\\
&- \left(\mu\left(\dot{g}\left(\lambda_1/\kappa_1\right)-\dot{g}\left(\lambda_2/\kappa_2\right)\right)\right)^2\\
&= \mu^2\dot{g}^2\left(\lambda_1/\kappa_1\right)+\mu^4\dot{g}\left(\lambda_1/\kappa_1\right)\dot{g}\left(\lambda_2/\kappa_2\right)+\dot{g}\left(\lambda_1/\kappa_1\right)\dot{g}\left(\lambda_2/\kappa_2\right)+\mu^2\dot{g}^2\left(\lambda_2/\kappa_2\right)\\
&- \left(\mu^2\dot{g}^2\left(\lambda_1/\kappa_1\right)-2\mu^2\dot{g}\left(\lambda_1/\kappa_1\right)\dot{g}\left(\lambda_2/\kappa_2\right)+\mu^2\dot{g}^2\left(\lambda_2/\kappa_2\right)\right)\\
&= \left(\mu^4+2\mu^2+1\right)\dot{g}\left(\lambda_1/\kappa_1\right)\dot{g}\left(\lambda_2/\kappa_2\right)
\end{aligned}$$

and

$$\left(bd-c^2\right)=\dot{g}\left(\lambda_1/\kappa_1\right)\dot{g}\left(\lambda_2/\kappa_2\right)>0$$

regardless of the choice of κ_1 and κ_2. Finally, we chose $\kappa_1=1$ and $\kappa_2=\mu$ in order to include the case $\mu=0$.

Case $\mathbf{z}_T=\mathbf{0}$

For $\kappa_1=1$ and $\kappa_2=\mu$ we have

$$a=\frac{\mu g\left(\lambda_2/\mu\right)-\mu g\left(\lambda_1\right)}{\left\|\mathbf{z}_T\right\|}, b=\frac{\mu^2\dot{g}\left(\lambda_1\right)+\dot{g}\left(\lambda_2/\mu\right)}{1+\mu^2}$$

$$c=\frac{\mu\left(\dot{g}\left(\lambda_1\right)-\dot{g}\left(\lambda_2/\mu\right)\right)}{1+\mu^2}, d=\frac{\dot{g}\left(\lambda_1\right)+\mu^2\dot{g}\left(\lambda_2/\mu\right)}{1+\mu^2}.$$

From (20) we notice that $\left\|\mathbf{z}_T\right\|=\lambda_2-\mu\lambda_1$ and hence for $\mathbf{z}_T\to\mathbf{0}$ there holds $\lambda_2\to\mu\lambda_1$. By using the above formulas we can see

$$\lim_{\mathbf{z}_T \to \mathbf{0}} a = \lim_{\lambda_2 \to \mu\lambda_1} \frac{\mu g\left(\lambda_2/\mu\right) - \mu g\left(\lambda_1\right)}{\lambda_2 - \mu\lambda_1} = \dot{g}\left(\lambda_1\right)$$

$$\lim_{\mathbf{z}_T \to \mathbf{0}} b = \lim_{\lambda_2 \to \mu\lambda_1} \frac{\mu^2 \dot{g}\left(\lambda_1\right) + \dot{g}\left(\lambda_2/\mu\right)}{1+\mu^2} = \dot{g}\left(\lambda_1\right)$$

$$\lim_{\mathbf{z}_T \to \mathbf{0}} c = \lim_{\lambda_2 \to \mu\lambda_1} \frac{\mu\left(\dot{g}\left(\lambda_1\right) - \dot{g}\left(\lambda_2/\mu\right)\right)}{1+\mu^2} = 0$$

$$\lim_{\mathbf{z}_T \to \mathbf{0}} d = \lim_{\lambda_2 \to \mu\lambda_1} \frac{\dot{g}\left(\lambda_1\right) + \mu^2 \dot{g}\left(\lambda_2/\mu\right)}{1+\mu^2} = \dot{g}\left(\lambda_1\right)$$

so that

$$\lim_{\mathbf{z}_T \to \mathbf{0}} \nabla h\left(\mathbf{z}\right) = \dot{g}\left(\lambda_1\right)\mathbf{I}. \tag{66}$$

It remains to linearize $h\left(\mathbf{z}\right)$ around $\mathbf{z}_T = 0$ and show, that it results in the right hand side of (66). We have

$$h\left(\mathbf{z}+\triangle\mathbf{z}\right) = g\left(\lambda_1+\triangle\lambda_1\right)\mathbf{u}_1 + \mu g\left(\left(\lambda_2+\triangle\lambda_2\right)/\mu\right)\mathbf{u}_2, \;\; h\left(\mathbf{z}\right) = g\left(\lambda_1\right)\mathbf{u}_1 + \mu g\left(\lambda_2/\mu\right)\mathbf{u}_2$$

where

$$\lambda_1 = \frac{-z_N}{1+\mu^2}, \;\; \lambda_2 = \frac{-\mu z_N}{1+\mu^2}, \triangle\lambda_1 = \frac{-\triangle z_N - \mu\left\|\triangle\mathbf{z}_T\right\|}{1+\mu^2}, \;\; \triangle\lambda_2 = \frac{\left\|\triangle\mathbf{z}_T\right\| - \mu\triangle z_N}{1+\mu^2}$$

$$\mathbf{u}_1 = \begin{cases} \begin{bmatrix} -\mu\frac{\triangle\mathbf{z}_T}{\left\|\triangle\mathbf{z}_T\right\|} \\ -1 \end{bmatrix} & \text{if} \quad \triangle\mathbf{z}_T \neq \mathbf{0} \\ \begin{bmatrix} -\mu\mathbf{w} \\ -1 \end{bmatrix} & \text{if} \quad \triangle\mathbf{z}_T = \mathbf{0} \end{cases}, \;\; \mathbf{u}_2 = \begin{cases} \begin{bmatrix} \frac{\triangle\mathbf{z}_T}{\left\|\triangle\mathbf{z}_T\right\|} \\ -\mu \end{bmatrix} & \text{if} \quad \triangle\mathbf{z}_T \neq \mathbf{0} \\ \begin{bmatrix} \mathbf{w} \\ -\mu \end{bmatrix} & \text{if} \quad \triangle\mathbf{z}_T = \mathbf{0} \end{cases}$$

and where $\mathbf{w} \in R^2$ and $\left\|\mathbf{w}\right\| = 1$. There also holds

$$g\left(\lambda_1+\triangle\lambda_1\right) - g\left(\lambda_1\right) = \dot{g}\left(\lambda_1\right)\triangle\lambda_1 + o\left(\triangle\lambda_1\right) = \dot{g}\left(\lambda_1\right)\triangle\lambda_1 + o\left(\left\|\triangle\mathbf{z}\right\|\right)$$

$$\mu g\left(\left(\lambda_2+\triangle\lambda_2\right)/\mu\right) - \mu g\left(\lambda_2/\mu\right) = \dot{g}\left(\lambda_2/\mu\right)\triangle\lambda_2 + o\left(\left\|\triangle\mathbf{z}\right\|\right)$$

where $\beta = o\left(\alpha\right)$ if $\beta/\alpha \to 0$ as $\alpha \to 0$. Tentatively assuming that $\triangle\mathbf{z}_T \neq \mathbf{0}$ and keeping in mind that $\left\|\triangle\mathbf{z}_T\right\| = \triangle\lambda_2 - \mu\triangle\lambda_1$, $0 = \lambda_2 - \mu\lambda_1$ and $\triangle\lambda_1 + \mu\triangle\lambda_2 = -\triangle z_N$, we can write

$$
\begin{aligned}
h(\mathbf{z}+\triangle\mathbf{z})-h(\mathbf{z}) &= (g(\lambda_1+\triangle\lambda_1)-g(\lambda_1))\,\mathbf{u}_1+\mu\,(g((\lambda_2+\triangle\lambda_2)/\mu)-g(\lambda_2/\mu))\,\mathbf{u}_2\\
&= \dot{g}(\lambda_1)\triangle\lambda_1\mathbf{u}_1+\dot{g}(\lambda_2/\mu)\triangle\lambda_2\mathbf{u}_2+o(\|\triangle\mathbf{z}\|)\\
&= \dot{g}(\lambda_1)\begin{bmatrix}(\triangle\lambda_2-\mu\triangle\lambda_1)\frac{\triangle\mathbf{z}_T}{\|\triangle\mathbf{z}_T\|}\\ -\triangle\lambda_1-\mu\triangle\lambda_2\end{bmatrix}+o(\|\triangle\mathbf{z}\|)\\
&= \dot{g}(\lambda_1)\triangle\mathbf{z}+o(\|\triangle\mathbf{z}\|).
\end{aligned}
$$

Hence, as required

$$\nabla h(\mathbf{z})=\dot{g}(\lambda_1)\mathbf{I}.$$

B. Appendix - Links

Acknowledgements

The support from EDF Energy is gratefully acknowledged.

Revision notes

Rev. 1 Initial revision;

References

[1] J. J. Moreau. Numerical aspects of the sweeping process. *Computer Methods in Applied Mechanics and Engineering*, 177(3-4):329–349, 1999.

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

[3] Tomasz Koziara and Nenad Bićanić. A distributed memory parallel multibody contact dynamics code. *International Journal for Numerical Methods in Engineering*, 87(1-5):437–456, 2011.

[4] Tomasz Koziara and Nenad Bićanić. Semismooth Newton method for frictional contact between pseudo-rigid bodies. *Computer Methods in Applied Mechanics and Engineering*, 197:2763–2777, 2008.

[5] G. De Saxcé and Z. Q. Feng. The bipotential method: a constructive approach to design the complete contact law with friction and improved numerical algorithms. *Mathematical and Computer Modelling*, 28:225–245, 1998.

[6] Masao Fukushima, Zhi-Quan Luo, and Paul Tseng. Smoothing functions for second-order-cone complementarity problems. *SIAM Journal on Optimization*, 12(2):436–460, 2002.

[7] D. E. Stewart. Rigid-body dynamics with friction and impact. *SIAM Review*, 42:3–39, 2000.

[8] B Brogliato, AA ten Dam, L Paoli, F Genot, and M Abadie. Numerical simulation of finite dimensional multibody nonsmooth mechanical systems. *Applied Mechanics Reviews*, 55(2):107–150, 2002.

[9] Hammad Mazhar, Toby Heyn, Dan Negrut, and Alessandro Tasora. Using nesterov's method to accelerate multibody dynamics with friction and contact. *ACM Trans. Graph.*, 34(3):32:1–32:14, May 2015.

[10] Tomasz Koziara. *Aspects of computational contact dynamics*. PhD thesis, University of Glasgow, http://theses.gla.ac.uk/429/, 2008.

[11] M. Zhang and R.D. Skeel. Cheap implicit symplectic integrators. *Applied Numerical Mathematics*, 25:297–302(6), 1997.

[12] R. Tyrrell Rockafellar and Roger J-B. Wets. *Variational analysis*. Springer, 1998.

[13] Shunsuke Hayashi, Nobuo Yamashita, and Masao Fukushima. A combined smoothing and regularization method for monotone second-order cone complementarity problems. *SIAM J. on Optimization*, 15(2):593–615, 2005.

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

[15] Farhang Radjai and Vincent Richefeu. Contact dynamics as a nonsmooth discrete element method. *Mechanics of Materials*, 41(6):715–728, 2009.

[16] C. T. Kelley and D. E. Keyes. Convergence analysis of pseudo-transient continuation. *SIAM J. Numer. Anal.*, 35(2):508–523, 1998.

[17] H. Cohen and R. G. Muncaster. *The Theory of Pseudo-rigid Bodies*. Springer, New York, 1988.