TR-2014-11

An Analysis of Several Methods for Handling Hard-Sphere Frictional Contact in Rigid Multibody Dynamics

Hammad Mazhar
Daniel Melanz
Michael C. Ferris
Dan Negrut

September 30, 2014

Abstract

This technical report provides a brief introduction into different ways that the equations of motion associated with the Differential Variational Inequality (DVI) approach can be posed. Both the primal and dual forms are outlined along with a discussion on the relaxed formulations. Using the dual relaxed formulation a static analysis was performed to understand how the Accelerated Projected Gradient Descent (APGD) method compares to several commercial solvers such as Mosek and PATH. A dynamic analysis was used to demonstrate how relaxing the model influences the dynamics of several simple tests problems.

Contents

1 Introduction

The goal of this technical report is to analyze several different solution methods to solve complementarity problems dealing with friction and contact. To this end the General Algebraic Modeling System (GAMS) [1] will be used as the test bed for different modeling and solution methods. Simulation data will be generated using Chrono [2] with Matlab used as the intermediate software to convert data formats.

The framework consists of a C++ library, Chrono [2] which is used for time integration and model specification, data is passed to Matlab in memory using the Matlab C++ API. A custom solver was written using this interface that passes the matrices associated with the problem into Matlab, solves the problem and then returns the solution back to Chrono. Within Matlab, a GAMS Data Exchange (GDX) [3] input file was generated for the GAMS model. Once GAMS has solved the problem it writes the solution to a GDX file which is read back into Matlab and subsequently passed to Chrono. This process repeats for each time integration step of a simulation.

2 Nomenclature

$\mathbf{M} \in \mathbb{R}^{N_{dof} \times N_{dof}}$ Mass Matrix

$\mathbf{D} \in \mathbb{R}^{N_{dof} \times (3N_c)}$ Contact Jacobian Matrix

$\mathbf{G} \in \mathbb{R}^{N_{dof} \times N_b}$ Joint Jacobian Matrix

$\mathbf{C} \in \mathbb{R}^{N_{dof} \times (3N_c+N_b)}$ Jacobian Matrix

$\mathbf{q} \in \mathbb{R}^{N_{dof} \times 1}$ Positions

$\mathbf{v} \in \mathbb{R}^{N_{dof} \times 1}$ Velocities

$\widehat{\gamma} \in \mathbb{R}^{3N_c \times 1}$ Contact Reaction Forces

$\gamma \in \mathbb{R}^{3N_c \times 1}$ Contact Reaction Impulses

$\widehat{\lambda} \in \mathbb{R}^{N_b \times 1}$ Joint Reaction Forces

$\lambda \in \mathbb{R}^{N_b \times 1}$ Joint Reaction Impulses

$\nu \in \mathbb{R}^{(3N_c+N_b) \times 1}$ Reaction Impulses

$\mathbf{N} \in \mathbb{R}^{(3N_c+N_b) \times (3N_c+N_b)}$ Shur Matrix

$\mathbf{r} \in \mathbb{R}^{(3N_c+N_b) \times 1}$ Shur RHS

$\mathbf{f} \in \mathbb{R}^{N_{dof} \times 1}$ External Forces

h Time Step

μ Friction Constant

Υ Friction Cone

Υ° Polar Friction Cone

Φ Gap Function

N_c Number of Contacts

N_b Number of Joints

N_{dof} Number of Degrees of Freedom

3 Formulations

3.1 Force-Acceleration Form

Below are the equations for motion for a system of rigid bodies that have contact (unilateral) and joint (bilateral) constraints. Eq. 1b is a simple force balance stating $\mathbf{M}\dot{\mathbf{v}} = \mathbf{f}$ where the forces comprise of applied forces such as gravity, reaction forces from joints and frictional contact forces. Eq. 1c states that the joints must be satisfied at the position level (No drift). The complementarity condition, Eq. 1d, states that the Lagrange multiplier $\widehat{\gamma}_{i,n}$ is positive when the gap Φ_i is zero and vice versa, a zero reaction force means that the gap is positive. Finally, using the maximum dissipation energy principle, Coulomb friction model can be posed as a minimization problem, Eq. 1e.

$$\dot{\mathbf{q}} = \underbrace{\mathbf{L}}_{\text{Velocity transformation matrix}}(\mathbf{q})\overbrace{\mathbf{v}}^{\text{Generalized velocities}} \tag{1a}$$

$$\underbrace{\mathbf{M}}_{\text{Mass matrix}}\overbrace{(\mathbf{q})}^{\text{Generalized Positions}}\dot{\mathbf{v}} = \underbrace{\mathbf{f}\,(t,\mathbf{q},\mathbf{v})}_{\text{Applied force}} - \overbrace{\mathbf{G}^{\mathrm{T}}(\mathbf{q},t)\widehat{\lambda}}^{\text{Reaction force}} + \sum_{i\in\mathcal{A}(\mathbf{q},\delta)} \underbrace{\left(\widehat{\gamma}_{i,n}\,\mathbf{D}_{i,n} + \widehat{\gamma}_{i,u}\,\mathbf{D}_{i,u} + \widehat{\gamma}_{i,w}\,\mathbf{D}_{i,w}\right)}_{\text{Frictional contact force}} \tag{1b}$$

$$\mathbf{0} = \mathbf{g}(\mathbf{q},t) \tag{1c}$$

$$i \in \mathcal{A}(\mathbf{q}(t),\delta) : 0 \leq \widehat{\gamma}_{i,n} \;\perp\; \overbrace{\Phi_i}^{\text{Gap function}}(\mathbf{q}) \geq 0 \tag{1d}$$

$$(\widehat{\gamma}_{i,u},\widehat{\gamma}_{i,w}) = \underset{\sqrt{(\widehat{\gamma}_{i,u})^2+(\widehat{\gamma}_{i,w})^2}\leq\mu_i\widehat{\gamma}_{i,n}}{\operatorname{argmin}} \underbrace{\mathbf{v}^T\left(\widehat{\gamma}_{i,u}\,\mathbf{D}_{i,u} + \widehat{\gamma}_{i,w}\,\mathbf{D}_{i,w}\right)}_{\text{Friction dissipation energy}} \tag{1e}$$

This model can be thought of as a "gap-force" model, where the complementarity condition is posed between the gap (or penetration distance) Φ_i and the contact force between the two objects in contact $\widehat{\gamma}_{i,n}$. The alternative to this model is the "impulse-velocity" model where the normal velocity is kept strictly positive, the downside of this is that drift may occur over time that causes small penetrations. This can become a problem with large models when the solver doesn't fully converge to a correct solution. Using a gap term accounts for drift between time steps and keeps the overall simulation more stable. However, as the amount of penetration is related to the size of the time step, large step sizes will cause the simulation to become unstable, therefore an appropriate time step should be determined for each simulation.

3.2 Discretized Form

When the equations in Eq. 1 are discretized with respect to time, the Lagrange multipliers become reaction impulses rather than forces. Stabilization terms are added for both the contacts and the joints that correct for violations at the position level. The relaxation term

in 2d is added so that the problem can be posed as the CCP in Eq. 13 it is possible to solve the equations of motion without this term using a fixed point iterations scheme and solving multiple optimization problems.

$$\overbrace{\mathbf{q}^{(l+1)}}^{\text{Generalized positions}} = \mathbf{q}^{(l)} + \overbrace{h}^{\text{Step size}} \underbrace{\mathbf{L}}_{\text{Velocity transformation matrix}}(\mathbf{q}^{(l)})\mathbf{v}^{(l+1)} \tag{2a}$$

$$\mathbf{M}(\overbrace{\mathbf{v}^{(l+1)}}^{\text{Generalized speeds}} - \mathbf{v}^{(l)}) = \underbrace{h\mathbf{f}(t^{(l)}, \mathbf{q}^{(l)}, \mathbf{v}^{(l)})}_{\text{Applied impulse}} - \overbrace{\mathbf{G}(\mathbf{q}^{(l)}, t)\lambda}^{\text{Reaction impulse}} + \sum_{i \in \mathcal{A}(q^{(l)}, \delta)} \underbrace{(\gamma_{i,n}\,\mathbf{D}_{i,n} + \gamma_{i,u}\,\mathbf{D}_{i,u} + \gamma_{i,w}\,\mathbf{D}_{i,w})}_{\text{Frictional contact reaction impulses}} \tag{2b}$$

$$0 = \underbrace{\frac{1}{h}\mathbf{g}(\mathbf{q}^{(l)}, t)}_{\text{Stabilization term}} + \mathbf{G}^{\mathrm{T}}\mathbf{v}^{(l+1)} + \frac{1}{h}\mathbf{g}_t \tag{2c}$$

$$i \in \mathcal{A}(q^{(l)}, \delta) : 0 \leq \overbrace{\frac{1}{h}\Phi_i(\mathbf{q}^{(l)})}^{\text{Stabilization term}} + \mathbf{D}_{i,n}^{T}\mathbf{v}^{(l+1)} - \overbrace{\mu_i \sqrt{\left(\mathbf{D}_{i,v}^{T}\mathbf{v}^{(l+1)}\right)^2 + \left(\mathbf{D}_{i,w}^{T}\mathbf{v}^{(l+1)}\right)^2}}^{\text{Relaxation/Tilting Term}} \perp \gamma_{i,n} \geq 0 \tag{2d}$$

$$(\gamma_{i,u}, \gamma_{i,w}) = \underset{\sqrt{{\gamma_{i,u}}^2 + {\gamma_{i,w}}^2} \leq \mu_i \gamma_{i,n}}{\operatorname{argmin}} \quad \mathbf{v}^{\mathrm{T}}\left(\gamma_{i,u}\,\mathbf{D}_{i,u} + \gamma_{i,w}\,\mathbf{D}_{i,w}\right) \tag{2e}$$

3.2.1 Description of equations

Eq. 2a Time integration to get new positions from new velocities

Eq. 2b Taking Eq. 1b and multiplying through by the timestep, unknowns are now impulses.

Eq. 2c Taylor series expansion of Eq. 1c where constraints are enforced at the velocity level. The stabilization term corrects for any drift between the current and actual solutions.

Eq. 2d Complementarity condition between the contact impulse and the Taylor series expansion of the gap term from Eq. 1d. The stabilization term corrects for any drift/penetration which can occur over time. The relaxation condition is added so that the KKT conditions for Eq. 2d and Eq. 2e can be written as shown in Eq. 13.

Eq. 2e Conic constraint based on friction model.

3.3 Forming the Cone Complementarity Problem (CCP)

In order to get to the Cone Complementarity Problem defined in 13 first write the KKT conditions for the optimization problem in Eq. 2e, where λ_i is a slack variable.

$$\mathbf{D}_{i,u}^{\mathrm{T}}\mathbf{v} + \lambda_i \frac{\gamma_{i,u}}{\sqrt{\gamma_{i,u}^2 + \gamma_{i,w}^2}} = 0 \tag{3a}$$

$$\mathbf{D}_{i,w}^{\mathrm{T}}\mathbf{v} + \lambda_i \frac{\gamma_{i,w}}{\sqrt{\gamma_{i,u}^2 + \gamma_{i,w}^2}} = 0 \tag{3b}$$

$$\mu_i \gamma_{i,n} - \sqrt{\gamma_{i,u}^2 + \gamma_{i,w}^2} \geq 0 \tag{3c}$$

$$\lambda_i \geq 0 \tag{3d}$$

$$\lambda_i \left(\mu_i \gamma_{i,n} - \sqrt{\gamma_{i,u}^2 + \gamma_{i,w}^2} \right) = 0 \tag{3e}$$

Then combine Eq. 3a and 3b to obtain the following expression for λ_i:

$$\lambda_i = \sqrt{\left(\mathbf{D}_{i,u}^{\mathrm{T}}\mathbf{v}\right)^2 + \left(\mathbf{D}_{i,w}^{\mathrm{T}}\mathbf{v}\right)^2} \tag{4}$$

The following expression can be obtained from Eq. 2d for the case where $\gamma_{i,n} > 0$:

$$\frac{1}{h}\Phi_i + \mathbf{D}_{i,n}^{\mathrm{T}}\mathbf{v} = \mu_i \sqrt{\left(\mathbf{D}_{i,u}^{\mathrm{T}}\mathbf{v}\right)^2 + \left(\mathbf{D}_{i,w}^{\mathrm{T}}\mathbf{v}\right)^2} = \mu_i \lambda_i \tag{5}$$

Now, define $\mathbf{g}_i = \left[\frac{1}{h}\Phi_i + \mathbf{D}_{i,n}^{\mathrm{T}}\mathbf{v}, \mathbf{D}_{i,u}^{\mathrm{T}}\mathbf{v}, \mathbf{D}_{i,w}^{\mathrm{T}}\mathbf{v}\right]^{\mathrm{T}}$ and consider the following:

$$\mathbf{g}_i^{\mathrm{T}}\gamma_i = \gamma_{i,n}\left(\frac{1}{h}\Phi_i + \mathbf{D}_{i,n}^{\mathrm{T}}\mathbf{v}\right) + \gamma_{i,u}\mathbf{D}_{i,u}^{\mathrm{T}}\mathbf{v} + \gamma_{i,w}\mathbf{D}_{i,w}^{\mathrm{T}}\mathbf{v} \tag{6}$$

Substitute Eq. 5, 3a, and 3b into Eq. 6:

$$\mathbf{g}_i^{\mathrm{T}}\gamma_i = \gamma_{i,n}\mu_i\lambda_i - \gamma_{i,u}\lambda_i \frac{\gamma_{i,u}}{\sqrt{\gamma_{i,u}^2 + \gamma_{i,w}^2}} - \gamma_{i,w}\frac{\gamma_{i,u}}{\sqrt{\gamma_{i,w}^2 + \gamma_{i,w}^2}} = \lambda_i \left(\gamma_{i,n}\mu_i - \sqrt{\gamma_{i,w}^2 + \gamma_{i,w}^2}\right) = 0 \tag{7}$$

where the last equality in Eq. 7 is based on Eq. 3e.

Using Eq. 3c the condition that $\gamma_i \in \Upsilon_i$ can be easily verified, and by Eq. 2d that $-\mathbf{g}_i \in \Upsilon_i^\circ$. Additionally, Eq. 7 shows that $-\mathbf{g}_i \perp \gamma_i$, so the complementarity condition $\Upsilon_i \ni \gamma \perp -\mathbf{g}_i \in \Upsilon_i^\circ$ holds.

By defining $\mathbf{b}_i = \left[\frac{1}{h}\Phi_i, 0, 0\right]^{\mathrm{T}}$ and $\mathbf{k} = \mathbf{M}\mathbf{v}^{(l)} + h\mathbf{f}\left(t^{(l)}, \mathbf{q}^{(l)}, \mathbf{v}^{(l)}\right)$, Eq. 2b can be expressed as

$$\mathbf{M}\mathbf{v}^{(l+1)} = \mathbf{k} + \mathbf{D}\gamma^{(l+1)}, \tag{8}$$

which leads to the following expression for $\mathbf{v}^{(l+1)}$ in terms of $\gamma^{(l+1)}$:

$$\mathbf{v}^{(l+1)} = \mathbf{M}^{-1}\left(\mathbf{k} + \mathbf{D}\gamma^{(l+1)}\right). \tag{9}$$

Define the matrix $\mathbf{N}$ and vector $\mathbf{r}$ as follows:

$$\mathbf{N} = \mathbf{D}^{\mathrm{T}}\mathbf{M}^{-1}\mathbf{D}, \tag{10}$$
$$\mathbf{r} = \mathbf{b} + \mathbf{D}^{\mathrm{T}}\mathbf{M}^{-1}\mathbf{k}. \tag{11}$$

With these definitions, it is easy to show that

$$\mathbf{g}_i = \mathbf{D}_i^{\mathrm{T}}\mathbf{v}^{(l+1)} + \mathbf{b}_i \tag{12a}$$
$$= \mathbf{D}_i^{\mathrm{T}}\left(\mathbf{M}^{-1}\mathbf{k} + \mathbf{M}^{-1}\mathbf{D}\gamma^{(l+1)}\right) + \mathbf{b}_i \tag{12b}$$
$$= \mathbf{D}_i^{\mathrm{T}}\mathbf{M}^{-1}\mathbf{D}\gamma^{(l+1)} + \mathbf{D}_i^{\mathrm{T}}\mathbf{M}^{-1}\mathbf{k} + \mathbf{b}_i \tag{12c}$$
$$= \left(\mathbf{N}\gamma^{(l+1)} + \mathbf{r}\right)_i. \tag{12d}$$

Therefore, it has been shown that solving the relaxed form of the equations of motion (Equations 2a-2e) is equivalent to solving the CCP in Equation 13.

$$\text{Find } \gamma_i^{(l+1)}, \text{ for } i = 1, \ldots, N_c$$
$$\text{such that } \Upsilon_i \ni \gamma_i^{(l+1)} \perp -\left(\mathbf{N}\gamma^{(l+1)} + \mathbf{r}\right)_i \in \Upsilon_i^{\circ} \tag{13a}$$
$$\text{where } \Upsilon_i = \{[x, y, z]^T \in \mathbb{R}^3 | \sqrt{y^2 + z^2} \leq \mu_i x\} \tag{13b}$$
$$\text{and } \Upsilon_i^{\circ} = \{[x, y, z]^T \in \mathbb{R}^3 | x \leq -\mu_i \sqrt{y^2 + z^2}\}, \tag{13c}$$

3.4 Bilateral Contraints

The formulation above is valid when dealing with contact constraints. Some modifications need to be made so that bilateral constraints can be solved in the same formulation. Modify 8, and 9 to include the bilateral constraint term

$$\mathbf{M}\mathbf{v}^{(l+1)} = \mathbf{k} - \mathbf{G}\lambda^{(l+1)} + \mathbf{D}\gamma^{(l+1)}, \tag{14a}$$
$$\mathbf{v}^{(l+1)} = \mathbf{M}^{-1}\left(\mathbf{k} - \mathbf{G}\lambda^{(l+1)} + \mathbf{D}\gamma^{(l+1)}\right). \tag{14b}$$

Define the full jacobian matrix $\mathbf{C}$ as follows:

$$\mathbf{C} = \begin{bmatrix}\mathbf{D} & \mathbf{G}\end{bmatrix} \tag{15}$$

specify the vector of stabilization terms $\mathbf{b}$ as:

$$\mathbf{b}_i = \left[\frac{1}{h}\Phi_i, 0, 0, ...\frac{1}{h}\Phi_{N_c}\right]^{\mathrm{T}} \tag{16}$$
$$\mathbf{b}_j = \left[\frac{1}{h}\mathbf{g}(\mathbf{q}^{(l)}, t)_j ... \frac{1}{h}\mathbf{g}(\mathbf{q}^{(l)}, t)_{N_b}\right]^{\mathrm{T}} \tag{17}$$
$$\text{for } i = 1, \ldots, N_c$$
$$\text{for } j = 1 + N_c, \ldots, N_b + N_c$$

then rewrite 10 and 11 to get

$$\mathbf{N} = \mathbf{C}^{\mathrm{T}}\mathbf{M}^{-1}\mathbf{C}, \tag{18}$$

$$\mathbf{r} = \mathbf{b} + \mathbf{C}^{\mathrm{T}}\mathbf{M}^{-1}\mathbf{k}.. \tag{19}$$

Next specify the new vector of unknowns ν:

$$\nu = \begin{bmatrix} \gamma \\ \lambda \end{bmatrix} \tag{20}$$

The CCP can then be re-written as shown in 21. The bilateral constrains are handled differently than the contact constraints as they do not nned to be projected onto a cone.

4 Solution Approaches

If we start at the CCP formulation of this problem, there are two possible quadratic problems that can be formed.

$$\begin{aligned}
&\text{Find } \nu_i^{(l+1)}, \text{ for } i = 1, \ldots, N_c \\
&\nu_j^{(l+1)}, \text{ for } j = 1 + N_c, \ldots, N_b + N_c \\
&\text{such that } \Upsilon_i \ni \nu_i^{(l+1)} \perp -\left(\mathbf{N}\nu^{(l+1)} + \mathbf{r}\right)_i \in \Upsilon_i^{\circ} && \text{(21a)} \\
&\text{such that } \mathbb{R}^m \ni \nu_j^{(l+1)} \perp -\left(\mathbf{N}\nu^{(l+1)} + \mathbf{r}\right)_j \in \{0\}^m && \text{(21b)} \\
&\text{where } \Upsilon_i = \{[x, y, z]^T \in \mathbb{R}^3 | \sqrt{y^2 + z^2} \leq \mu_i x\} && \text{(21c)} \\
&\text{and } \Upsilon_i^{\circ} = \{[x, y, z]^T \in \mathbb{R}^3 | x \leq -\mu_i \sqrt{y^2 + z^2}\}, && \text{(21d)}
\end{aligned}$$

4.1 Primal Model

The primal form of this model solves the problem using velocities as the unknowns. A change of variables needs to be performed for the complementarity condition stated in Eq. 13b. The set of equations Eq. 12a-12d can be used to convert the unknowns from ν, which are impulses, to $\mathbf{v}$ which are velocities.

Eq. 21a can then be re-written as shown in Eq. 22

$$\Upsilon_i \ni \nu_i^{(l+1)} \perp -\left(\mathbf{C}^{\mathrm{T}}\mathbf{v}^{(l+1)} + \mathbf{b}\right)_i \in \Upsilon_i^{\circ} \tag{22}$$

Equations 22, 13b and 13c form the first order optimality conditions which allow the problem to be posed as the following quadratic programming problem with conic constraints.

$$\begin{aligned} \min f(\gamma) &= \frac{1}{2}\mathbf{v}^{(l+1)^{\mathrm{T}}}\mathbf{M}\mathbf{v}^{(l+1)} + \mathbf{f}^{\mathrm{T}}\mathbf{v} \\ \text{subject to } \Upsilon_i^{\circ} &\ni \left(\mathbf{C}^{\mathrm{T}}\mathbf{v}^{(l+1)} + \mathbf{b}\right)_i \\ \{0\}^m &\ni \left(\mathbf{C}^{\mathrm{T}}\mathbf{v}^{(l+1)} + \mathbf{b}\right)_j \\ \text{for } i &= 1, 2, \ldots, N_c \\ j &= 1 + N_c, 2 + N_c, \ldots, N_b + N_c \end{aligned} \tag{23}$$

4.2 Dual Model

In the dual model the problem is posed such that the unknowns are impulses rather than velocities. The constraints are easier to handle as the projection of $\nu_i \in \Upsilon_i$ is more straightforward. However the computation of $\mathbf{N}$ is more costly. Most of the time needed to solve this optimization problem is used to perform matrix vector multiplications. In this case equations 13a, 13b and 13c form the first order optimality conditions which allow the problem to be posed as a quadratic programming problem with conic constraints.

$$\begin{aligned} \min f(\nu) &= \frac{1}{2}\nu^{\mathrm{T}}\mathbf{N}\nu + \mathbf{r}^{\mathrm{T}}\gamma \\ \text{subject to } \nu_i &\in \Upsilon_i \\ \nu_j &\in \mathbb{R}^m \\ \text{for } i &= 1, 2, \ldots, N_c \\ j &= 1 + N_c, 2 + N_c, \ldots, N_b + N_c \\ \nu &= \left[\gamma_1^{\mathrm{T}}, \cdots, \gamma_{N_c}^{\mathrm{T}}, \cdots \lambda_1^{\mathrm{T}}, \cdots, \lambda_{N_b}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^{3N_c+N_b} \end{aligned} \tag{24}$$

4.3 Fixed Point Iteration

An alternate way to write the equations of motion does not contain the relaxation term as in Eq. 2d, the discretized version is shown in Eq. 25

$$i \in \mathcal{A}(q^{(l)}, \delta) : 0 \leq \frac{1}{h}\Phi_i(\mathbf{q}^{(l)}) + \mathbf{D}_{i,n}^T\mathbf{v}^{(l+1)} \perp \gamma_{i,n} \geq 0 \tag{25}$$

For both of the models posed above an approximation was made to the relaxation term s where $s = \mu\sqrt{{\mathbf{D}_{i,v}}^{\mathrm{T}}\mathbf{v}^{(l+1)^2} + \mathbf{D}_{i,w}^{\mathrm{T}}\mathbf{v}^{(l+1)^2}}$ setting it to zero. By doing so the problem becomes easier to solve as the computation of s requires the unknown velocities. One solution is to use a fixed point iteration scheme where the same quadratic optimization problem is solved and s is fixed for each solve. At the end of the solve the value is updated and the problem solved again until s has converged. The two optimization problems in Eq. 23 and Eq. 24

need to be modified slightly as shown in Eq. 26 for the primal problem and Eq. 27 for the dual problem.

$$
\begin{aligned}
\min f(\gamma) &= \frac{1}{2}\mathbf{v}^{(l+1)^{\mathrm{T}}}\mathbf{M}\mathbf{v}^{(l+1)} + \mathbf{f}^{\mathrm{T}}\mathbf{v} \\
\text{subject to } \Upsilon_i^{\circ} &\ni \left(\mathbf{C}^{\mathrm{T}}\mathbf{v}^{(l+1)} + \mathbf{b}\right)_i - \mu\sqrt{\mathbf{C}_{i,v}^{\mathrm{T}}\mathbf{v}^{(l+1)^2} + \mathbf{C}_{i,w}^{\mathrm{T}}\mathbf{v}^{(l+1)^2}} \\
\{0\}^m &\ni \left(\mathbf{C}^{\mathrm{T}}\mathbf{v}^{(l+1)} + \mathbf{b}\right)_j \\
\text{for } i &= 1, 2, \ldots, N_c \\
j &= 1 + N_c, 2 + N_c, \ldots, N_b + N_c
\end{aligned}
\tag{26}
$$

$$
\begin{aligned}
\min f(\gamma) &= \frac{1}{2}\gamma^{\mathrm{T}}\mathbf{N}\gamma + \mathbf{r}^{\mathrm{T}}\gamma \\
\text{subject to } \nu_i &\in \Upsilon_i \\
\nu_j &\in \mathbb{R}^m \\
\mathbf{r}_i &= \mathbf{b}_i + \mathbf{D}^{\mathrm{T}}\mathbf{M}^{-1}\boldsymbol{k} - \mu\sqrt{\mathbf{D}_{i,v}^{\mathrm{T}}\mathbf{v}^{(l+1)^2} + \mathbf{D}_{i,w}^{\mathrm{T}}\mathbf{v}^{(l+1)^2}} \\
\mathbf{r}_j &= \mathbf{b}_j + \mathbf{G}^{\mathrm{T}}\mathbf{M}^{-1}\boldsymbol{k} \\
\text{for } i &= 1, 2, \ldots, N_c \\
j &= 1 + N_c, 2 + N_c, \ldots, N_b + N_c
\end{aligned}
\tag{27}
$$

4.4 Equations of Motion (EOM) Model Using Extended Mathematical Programming (EMP)

The final way to solve this problem is to solve the EOM in Eq. 2 directly using the GAMS Extended Mathematical Programming (EMP) framework [4]. Like before, the equations can be solved with or without the relaxation term.

5 Numerical Experiments

In this section the APGD solver is compared to other methods using several different numerical experiments. A static analysis in Sec. 5.2 is used to demonstrate how the APGD method scales and converges for the relaxed dual formulation. Dynamic tests in Sec. 5.3 are used to demonstrate how the relaxation term influences the results of several different test cases.

For all numerical experiments, double precision arithmetic was used so that a fair comparison with respect to timing could be made.

5.1 Convergence Behavior

The purpose of this test was to show how the APGD and Mosek [5, 6] solvers react when provided different initial conditions. Two sets of initial conditions were generated, the first is referred to as "thoroughly settled" and the second as "poorly settled". for each set of data, 320 spheres were settled in a container for 4 seconds of physical time with a time step of $h = .0005$. The thoroughly settled data used up to 10000 iterations to settle while the poorly settled data was limited to only 10 solver iterations per step. After 4 seconds of simulation the settled data sets were stored and then used as the initial conditions for both Mosek and APGD. The goal was to solve both problems using Mosek for a single time step and get an optimal objective value, then using APGD, determine how many iterations it takes to reach the same objective value. Table 5.1 shows the condition number of the shur matrix, the kinetic energy of the spheres and the maximum number of iterations used during the settling process.

Because the simulation parameters for the thoroughly settled data set allow the simulation to run to a larger number of iterations, and therefore better convergence, the kinetic energy of the settled data is lower than that of the poorly settled case. In practice this also means that the thoroughly settled data set has smaller inter-penetrations between spheres resulting in a more stable configuration that is easier to solve.

	$\text{cond}(\mathbf{D}^T\mathbf{M}^{-1}\mathbf{D})$	Kinetic Energy	Max Iterations
Thoroughly Settled	8.33E+22	0.0215	10000
Poorly Settled	4.67E+23	1.5637	10

Table 1: Condition number, kinetic energy and the maximum iterations used during settling for both data sets.

In order to plot the objective function on a log scale its absolute value was taken. Default settings were used for APGD and Mosek, PATH [7] is not shown for the objective function plots as it is solving the EOM directly and there is no specified objective function. PATH had difficulty solving larger problems as it has a default time limit of 3600 seconds when solving a model. This limit can be increased but the as the default was already several orders of magnitude slower than APGD or Mosek, the limit was not changed.

5.1.1 Thoroughly settled data

Letting the spheres settle thoroughly resulted in 1049 contacts, this problem was then solved using the Mosek solver, the objective function value returned by this solver was used as the target objective value for the APGD solver, the APGD solver was allowed to iterate without an upper bound until it reached this target value. In this case the target objective was -0.0232 reached after 23 Mosek iterations, after 561 iterations the APGD solver reached a value of -0.023199. Note that as the dual formulation is being used the objective value used here is the dual objective.

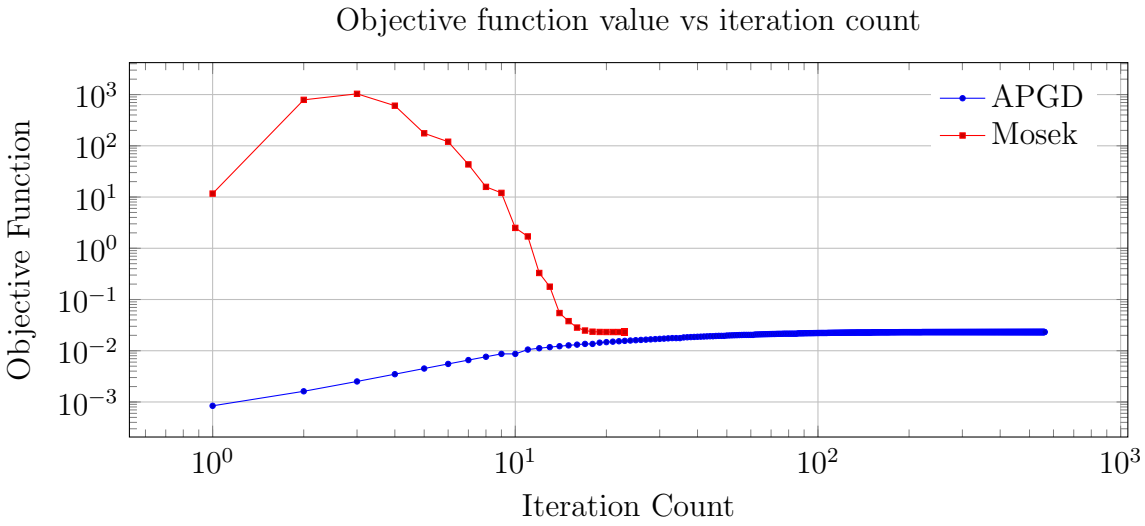

Figure 1: Absolute value of the objective function value for increasing number of iterations on a log-log plot. Mosek has strange convergence behavior in terms of it's objective function, ultimately it reaches the same solution.

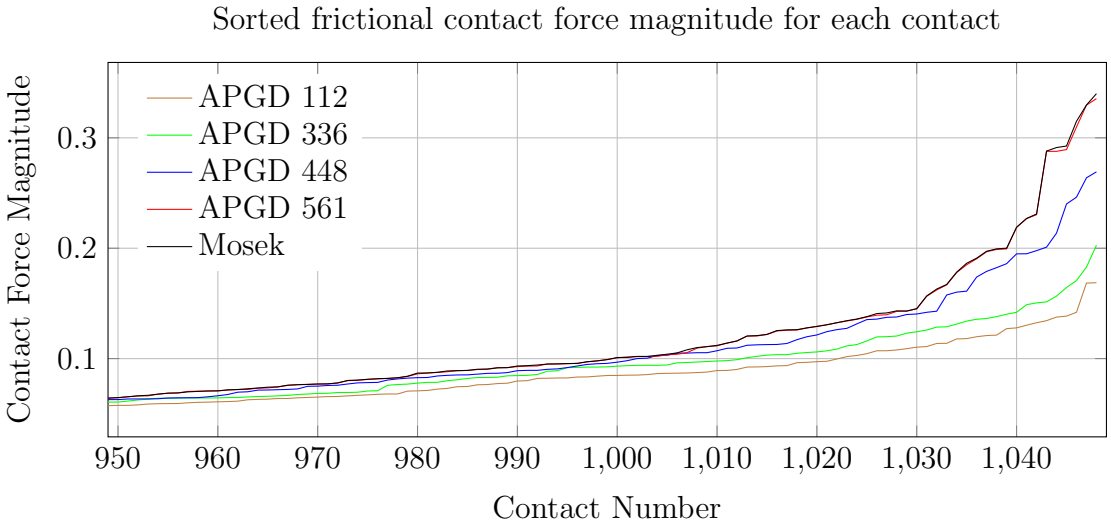

Figure 2: Increasing the number of iterations for the APGD solver brings the solution closer to that of Mosek. The APGD 561 case is exactly on top of the Mosek solution, making it difficult to see the line.

5.1.2 Poorly settled data

Letting the spheres settle poorly resulted in 1050 contacts, the Mosek solver performed 21 iterations to get to an objective function value of -0.5312, in comparison APGD took 825 to get to an objective function value of -0.53119.

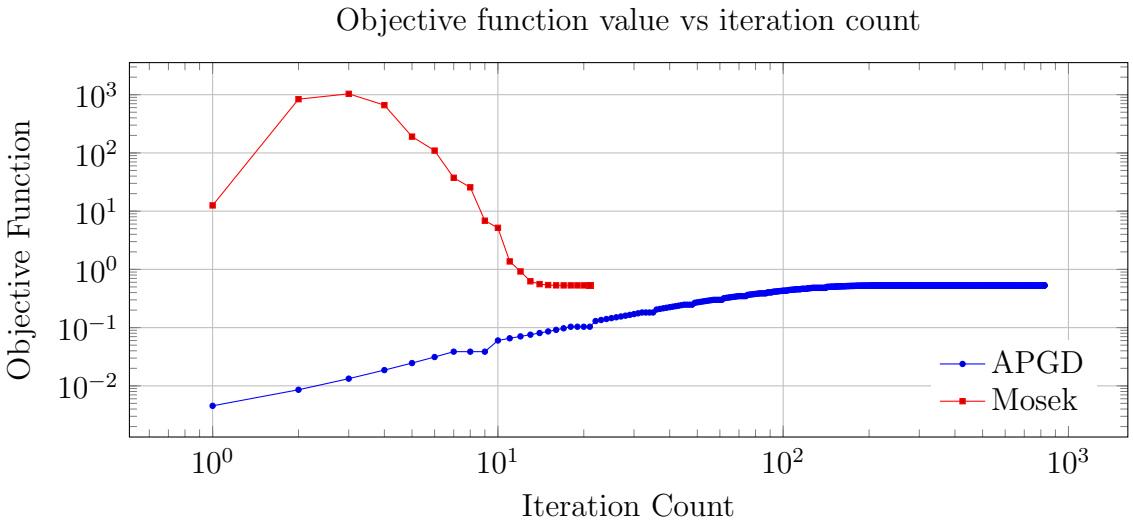

Figure 3: Absolute value of the objective function value for increasing number of iterations on a log-log plot.

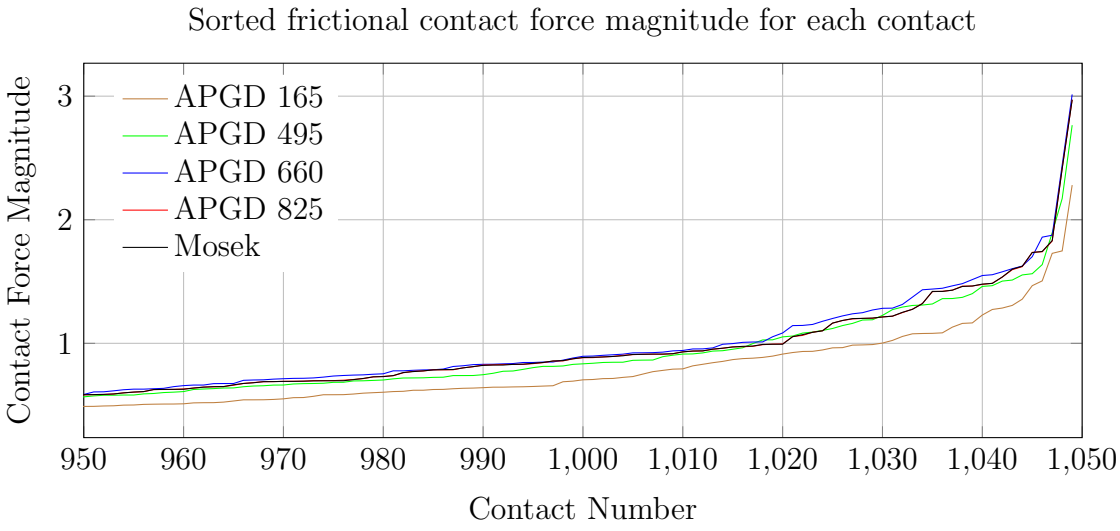

Figure 4: Increasing the number of iterations for the APGD solver brings the solution closer to that of Mosek.

5.2 Scaling analysis

The goal of this test was to understand how the solver scales with increasing problem size initial data with increasing number of settled spheres was used. The number of spheres ranged from 64 to 2560 at increments of 64 bodies. The number of contacts ranged from 77 to 9228. PATH results are shown for the first six problems after which it timed out after

3600 seconds. All tests were run on a 2.3 GHz Intel Core i7-3615QM with 4 threads, for mosek the `MSK_IPAR_NUM_THREADS` parameter was set to 4 in GAMS.

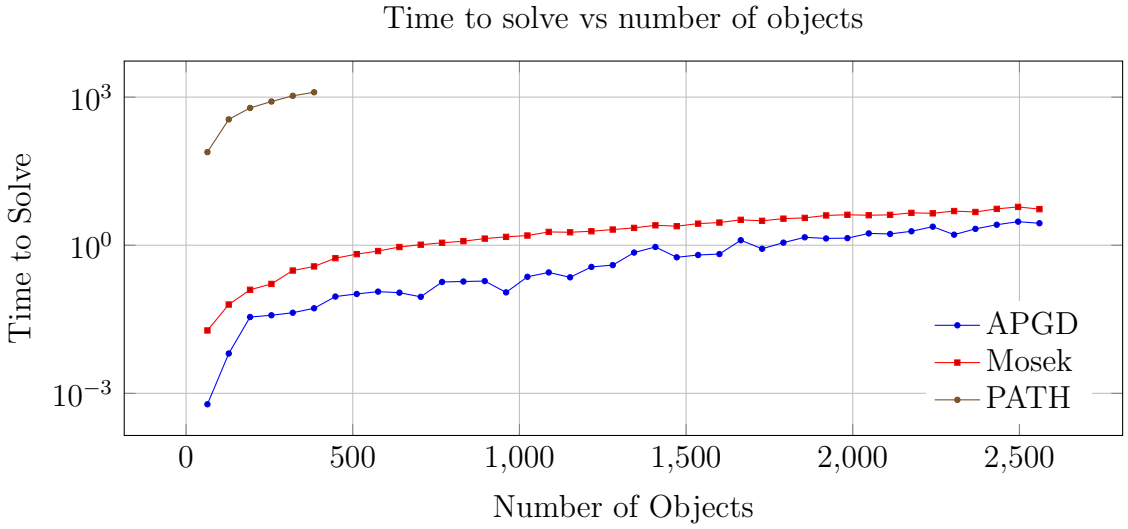

Figure 5: Wall time taken to converge for increasing problem sizes plotted on a log-log plot.

In Fig. 6 the primal objective function of the data input to the solver is comapred to that computed using the converged solution. Note that when settling the data there was an upperbound of iterations 10000 iterations that could be performed, this leads to an artificial "softening" of the problem. When enough iterations are used and the problem is properly converged, the simulation becaomes "stiffer" leading to a larger primal objective value after solving than the initial input. The reason that this occurs for larger problems is because during the settling process 10000 iterations are not enough to allow the problem to converge

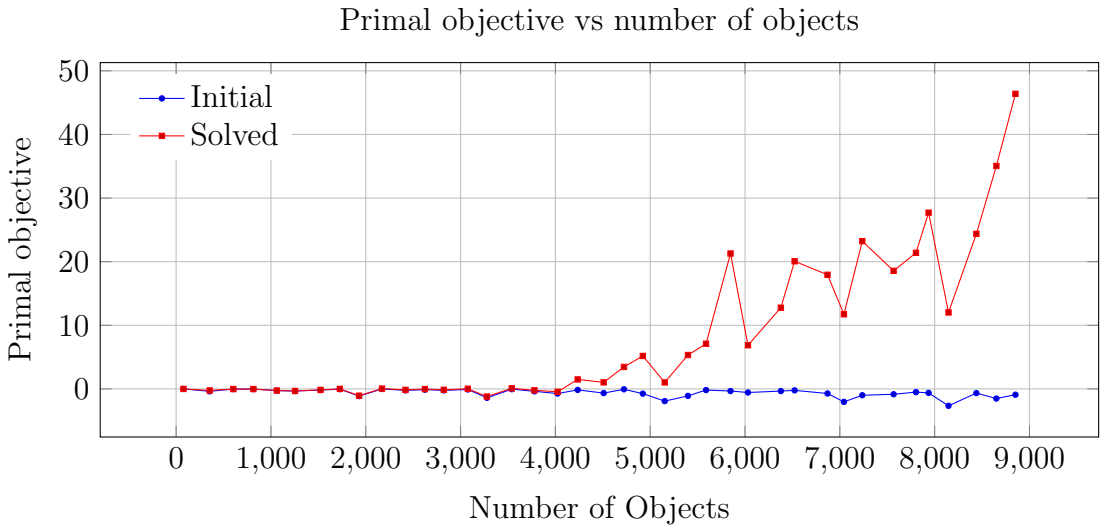

Figure 6: Comparison of the primal objective of the simulation input data and the primal objective of the converged solution.

5.3 Dynamic Results

In this section the differences between the dual, primal, EOM and their relaxed variations will be shown. Several simple problems will be shown to demonstrate how the choice in the formulation impacts the result. For brevity the following naming convention will be adopted in this section:

Primal Fixed Point Iteration Primal Formulation

Primal-Relaxed Relaxed Primal Formulation

Dual Fixed Point Iteration Dual Formulation

Dual-Relaxed Relaxed Dual Formulation

EOM Solving EOM without relaxation

EOM-Relaxed Solving EOM with relaxation

TO make it easier to understand the plots, the following convention will be used in this section: circle markers represent the fixed point iteration version of an algorithm, square markers represent the relaxed version.

5.3.1 Ball Sliding on Plane

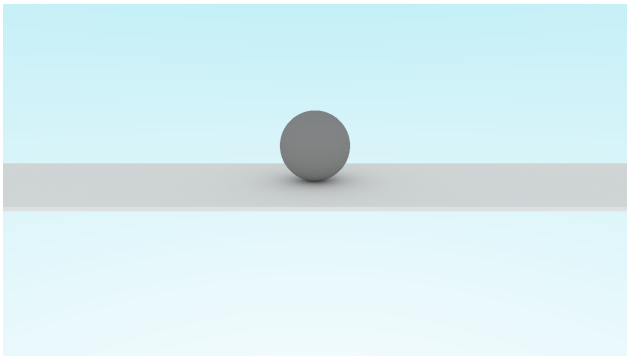

Figure 7: Ball rolling on plane

In this model a ball is in contact with a plane with an initial velocity of $-2\,\mathrm{m/s}$ in the x direction. The ball has a radius of $1\,\mathrm{m}$ and the contact has a friction value of $\mu = .2$. The ball is initially sliding but slowly begins to roll due to friction, eventually it goes into a steady state rolling motion. The time it takes to get to this state can be computed from the equation $t_{rolling} = \frac{2*v_0}{7*\mu*g}$ [8].For an initial velocity of $2\,\mathrm{m/s}$ and $g = 9.81 m/s^2$, the ball will be fully rolling at $t_{rolling} = .291s$. A numerical integration step size of $h = 0.0025$ was used for these simulations.

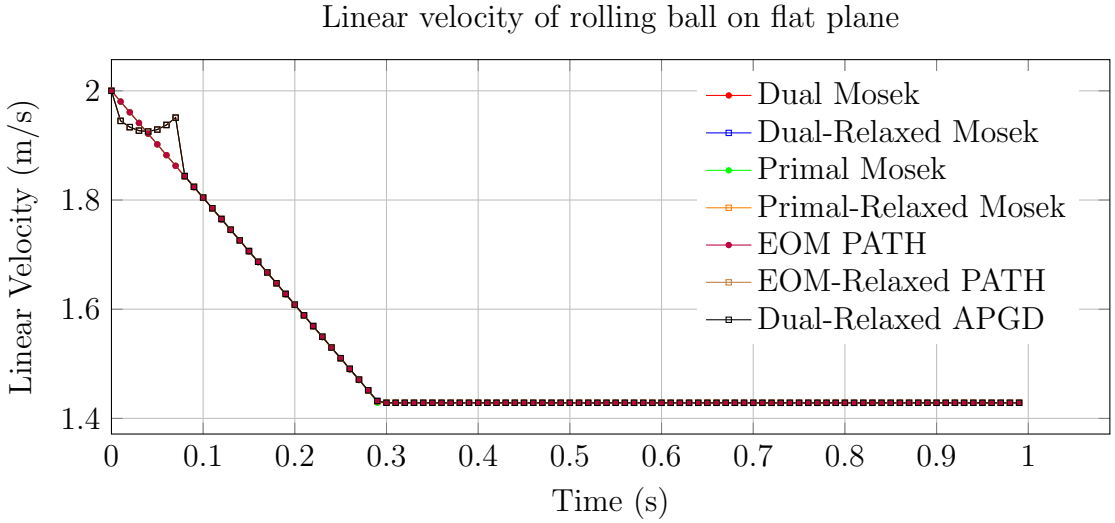

Figure 8: Comparison of Mosek, PATH and APGD for the relaxed and fixed point iteration versions. Note that because the problem is very simple the solutions returned by the relaxed and fixed point solutions are on top of each other.

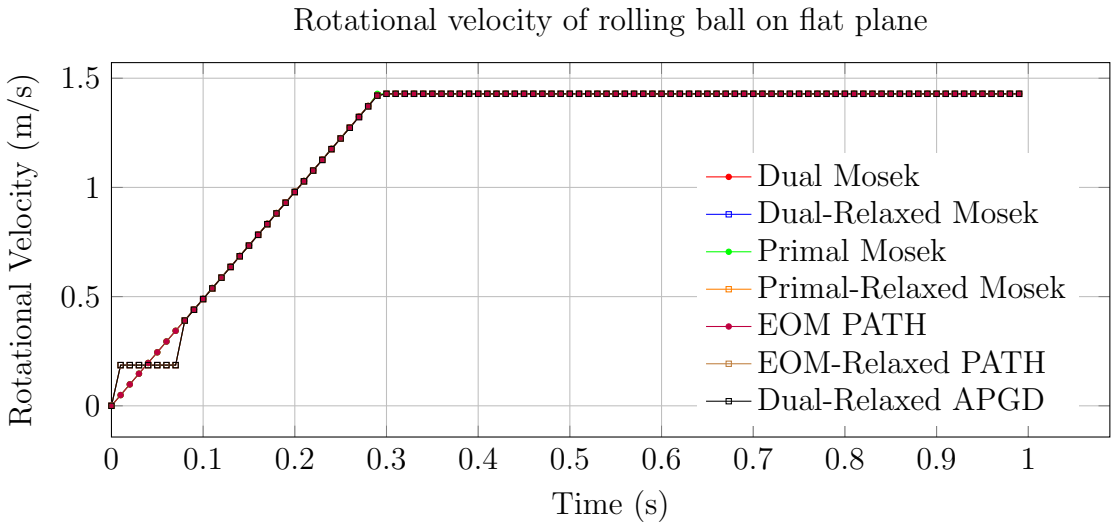

Figure 9: Comparison of Mosek, PATH and APGD for the relaxed and fixed point iteration versions. Note that because the problem is very simple the solutions returned by the relaxed and fixed point solutions are on top of each other.

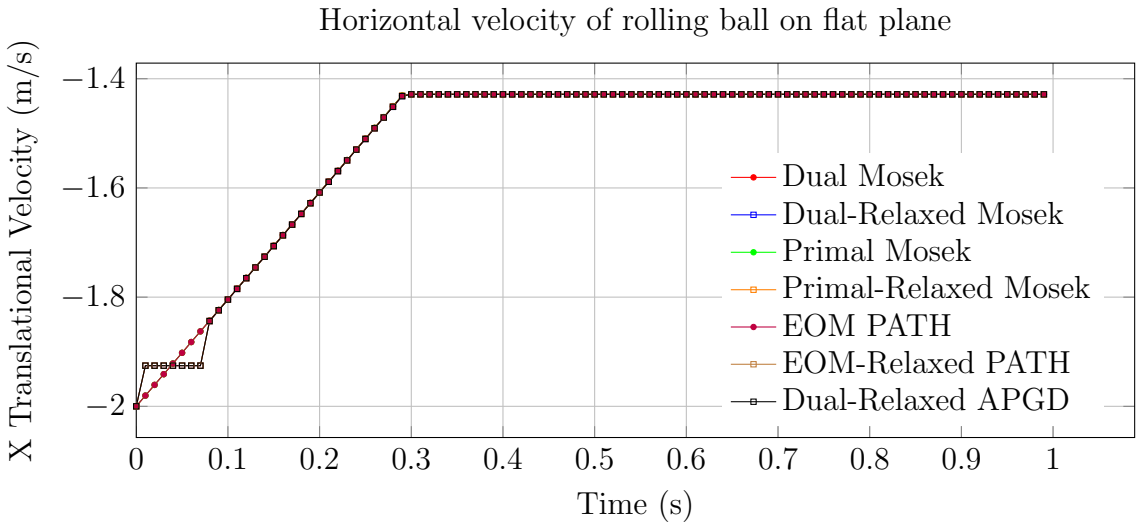

Figure 10: Comparison of Mosek, PATH and APGD for the relaxed and fixed point iteration versions. Note that because the problem is very simple the solutions returned by the relaxed and fixed point solutions are on top of each other.

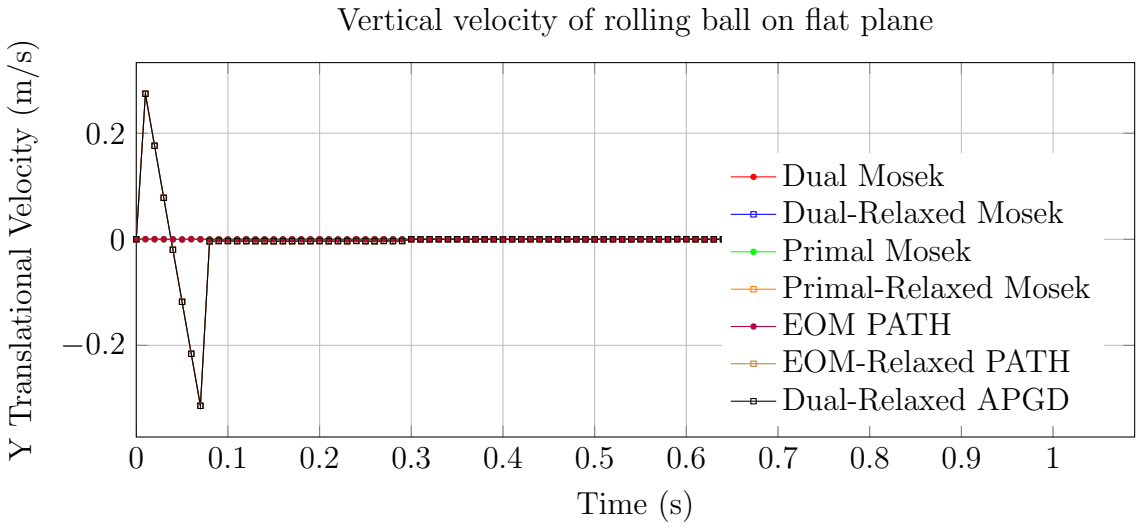

Figure 11: Comparison of Mosek, PATH and APGD for the relaxed and fixed point iteration versions. The relaxation causes an initial vertical velocity once sliding begins, this is what causes the spike in velocity magnitude.

5.3.2 Box Sliding on Plane

In this model a box is in contact with a plane with an initial velocity of $-2\,\mathrm{m/s}$ in the x direction. The box has a length/width/height of $1\,\mathrm{m}$ and the contact has a friction value of $\mu = .3$ with 4 contact points between the pair of objects.

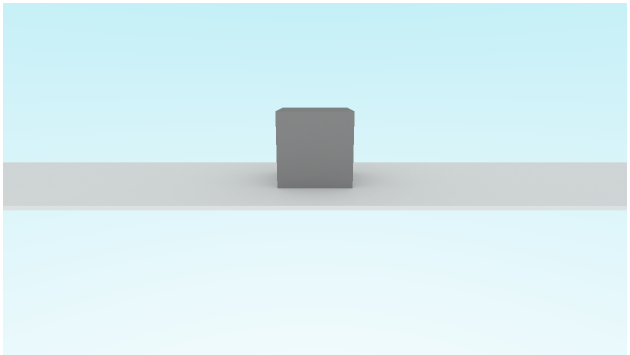

Figure 12: Box sliding on plane

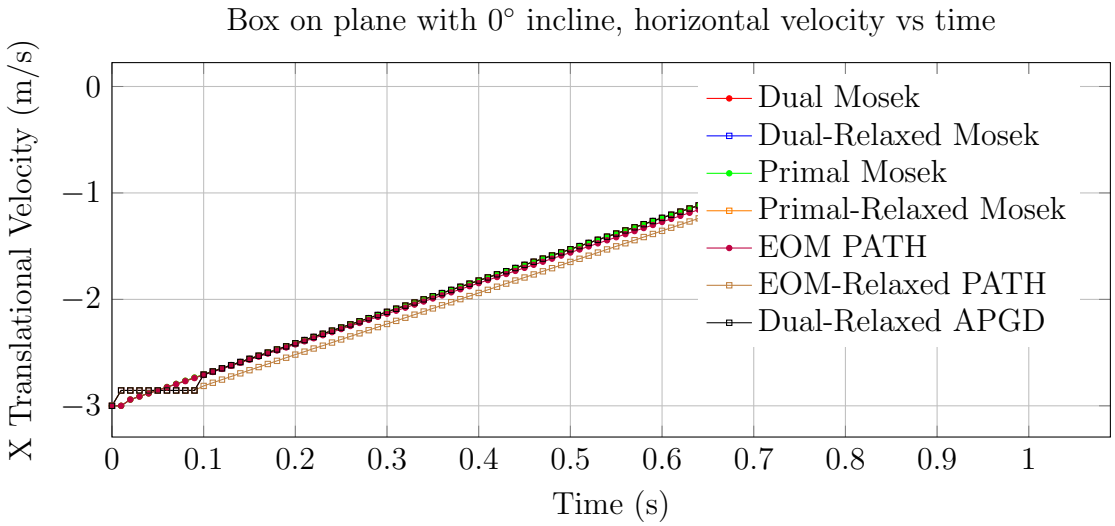

Figure 13: Comparison of Mosek, PATH and APGD for the relaxed and fixed point iteration versions. For this simulation the solution returned by the PATH solver for the relaxed EOM model diverges from the others.

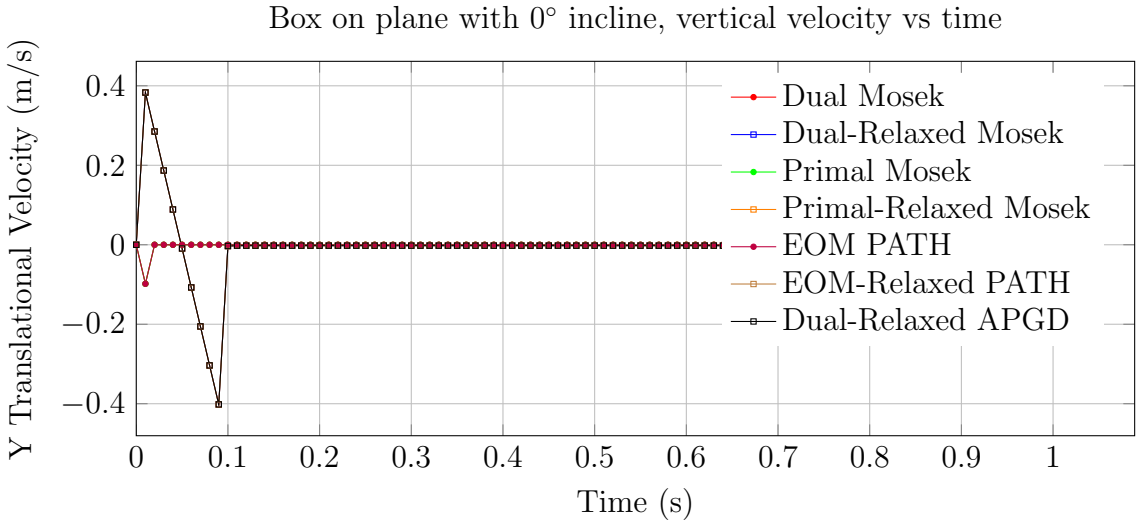

Figure 14: Comparison of Mosek, PATH and APGD for the relaxed and fixed point iteration versions. Notice that because of the relaxation term the box initially lifts off of the surface before falling and coming back into contact with the ground plane again.

5.4 Box Sliding on Incline From Rest

In this model a box is in perfect contact with a plane at an incline of 65° and initially at rest. The box has a length/width/height of 1 m and the friction is $\mu = .2$.

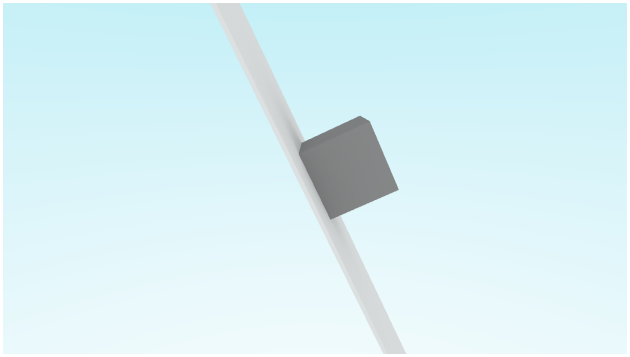

Figure 15: Box sliding on plane

[H]

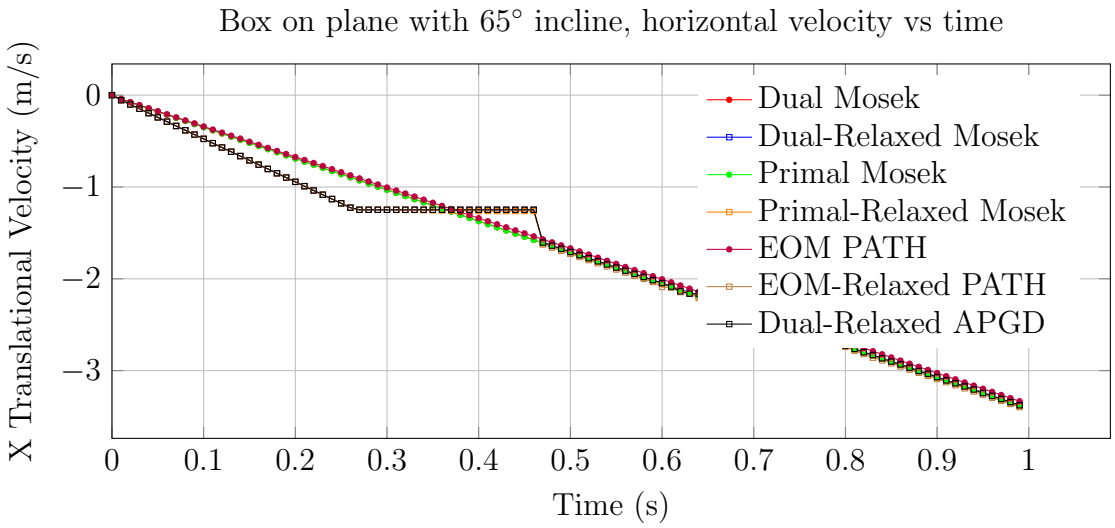

Figure 16: Comparison of Mosek, PATH and APGD for the relaxed and fixed point iteration versions. For this test all of the solvers come up with slightly different solutions, this could be due to a number of factors including small numerical errors that compound together.

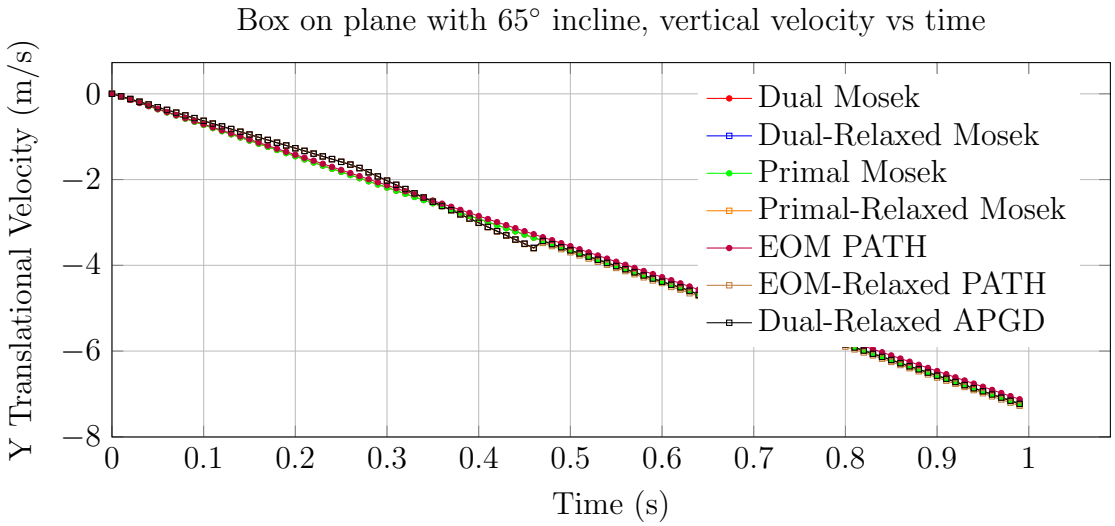

Figure 17: Comparison of Mosek, PATH and APGD for the relaxed and fixed point iteration versions. For this test all of the solvers come up with slightly different solutions, this could be due to a number of factors including small numerical errors that compound together.

5.5 Box Sliding on Incline To Rest

In this model a box is in perfect contact with a plane at an incline of 25° with an initial velocity of $-2\,\mathrm{m/s}$ in the x direction. The box has a length/width/height of $1\,\mathrm{m}$ and the friction is $\mu = .6$.

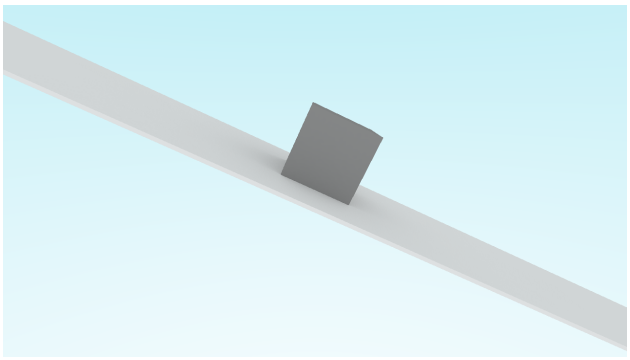

Figure 18: Box sliding on plane

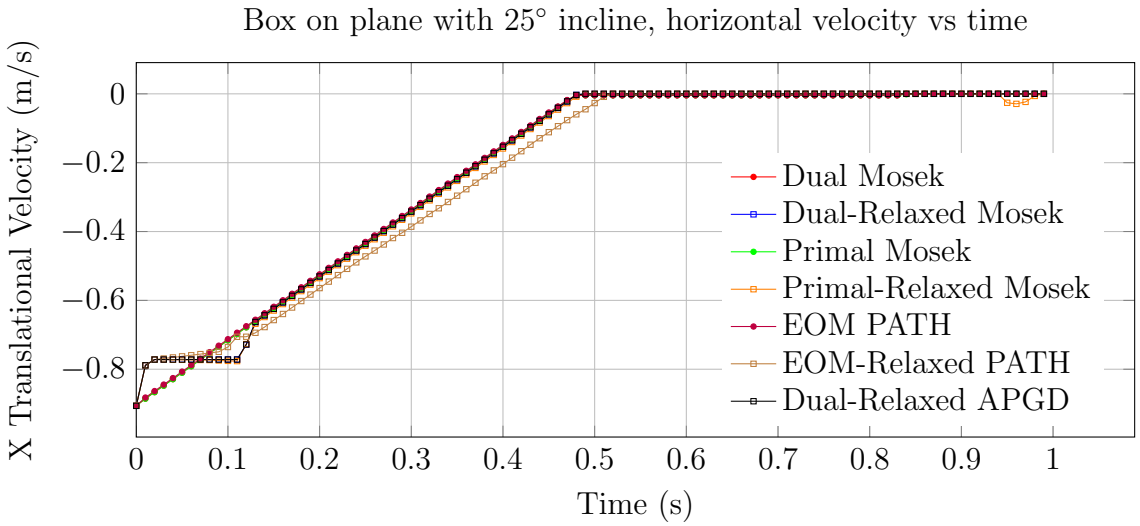

Figure 19: Comparison of Mosek, PATH and APGD for the relaxed and fixed point iteration versions. For this simulation the solution returned by the PATH solver for the relaxed EOM model diverges from the others.

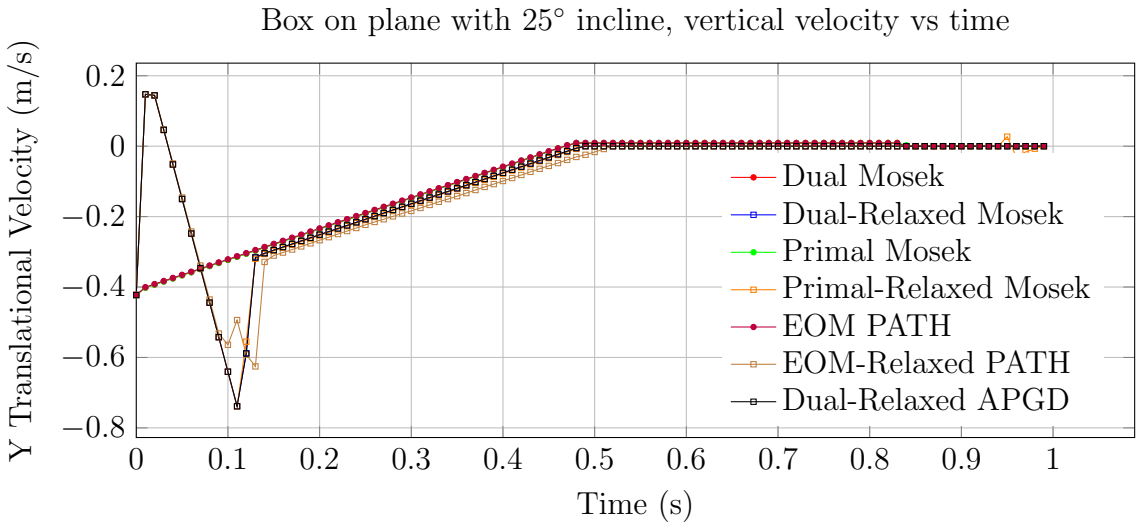

Figure 20: Comparison of Mosek, PATH and APGD for the relaxed and fixed point iteration versions. For this simulation the solution returned by the PATH solver for the relaxed EOM model diverges from the others. Also, notice that because of the relaxation term the box initially lifts off of the surface before falling and coming back into contact with the ground plane again.

6 Conclusions

6.1 Static Results

Results showed that both APGD and Mosek converge to similar objective values at similar speeds for the settled simulation results if the data set was settled properly. When the data was not settled properly, small changes in the number of objects resulted in large changes in the objective value. In such cases the number of iterations required to converge for APGD varied greatly while Mosek consistently converges within the same number of iterations and wall clock time. One downside to using Mosek, as shown in Fig. 1 is that it does not guarantee a better solution with increasing iterations. APGD can be terminated at any time and the method guarantees that the solution is better or equal to that computed using the previous iteration. Mosek [5] is a homogeneous primal-dual interior point method that, due to it's homogenization constant, does not guarantee that the solution lies on the interior of the feasible set. The result of this is that the solver might initially reach a very small objective function value that is not feasible, the solver will then correct for this, which will cause the objective function value to get larger, until the solution is within the feasible set and is optimal.

In terms of memory use and scalability, both APGD and Mosek are sparse solvers that do not store the full matrix which is why their performance and scaling is similar.

6.2 Dynamic Results

The dynamic results showed how the choice of method (primal or dual) along with relaxation effect the solution. Both the primal and dual methods, without relaxation along with the EOM without relaxation will result in the same solution. This means that a fixed point iteration needs to be carried out to update the tilting term in either the constraints (for the primal) or the objective function (for the dual). This requires the solution of several optimization problems which is prohibitive in terms of the time taken to solve. By adding in a relaxation term the solution can be acquired by solving a single optimization problem but as Fig. 8 shows for a simple test problem of a ball rolling on a plane, the solution diverges from the expected solution. Overall it does not affect the steady state solution but it will cause differences initially.

For problems with more than one contact point, such as the box sliding on an inclined plane, the same phenomena can be seen with the initial results differing when using a relaxed model.

References

[1] GAMS Development Corporation. General Algebraic Modeling System (GAMS) Release 24.2.1. Washington, DC, USA, 2013.

[2] Project Chrono. Chrono: An Open Source Framework for the Physics-Based Simulation of Dynamic Systems, 2014.

[3] GAMS Development Coproration. *GAMS GDX facilities and tools*. Washington, DC, USA, jun 2014.

[4] GAMS Development Corporation, Washington, DC, USA. *GAMS - The Solver Manuals, GAMS Release 24.2.1*, 2013.

[5] ErlingD. Andersen and KnudD. Andersen. The mosek interior point optimizer for linear programming: An implementation of the homogeneous algorithm. In Hans Frenk, Kees Roos, Tams Terlaky, and Shuzhong Zhang, editors, *High Performance Optimization*, volume 33 of *Applied Optimization*, pages 197–232. Springer US, 2000.

[6] E.D. Andersen, C. Roos, and T. Terlaky. On implementing a primal-dual interior-point method for conic quadratic optimization. *Mathematical Programming*, 95(2):249–277, 2003.

[7] Steven P Dirkse and Michael C Ferris. The path solver: A non-monotone stabilization scheme for mixed complementarity problems. 1993.

[8] J. C. Trinkle. Formulation of multibody dynamics as complementarity problems. In *Volume 5: 19th Biennial Conference on Mechanical Vibration and Noise, Parts A, B, and C*. ASME.