On the Convergence of Fixed-point Iteration in Solving Complementarity Problems Arising in Robot Locomotion and Manipulation

Ying Lu[1] and Jeff Trinkle[2]

***Abstract*— Model-based approaches to the planning or control of robot locomotion or manipulation requires the solution of complementarity problems that model intermittent contact. Fixed-point iteration is a method of computing fixed points of functions and there are several fixed-point theorems to guarantee the existence of fixed points. With the help of proximal point functions, the complementarity problems that arise in multibody dynamics can be rewritten in a form suitable for solution by a fixed-point iteration method. This fixed-point "prox method" has been popular over the last decades. However, the tuning of the iteration parameter r is difficult, because r affects the convergence of the fixed-point iteration method in ways not understood by current theoretical results. In this paper, we first investigate some factors that affect the choice of r, which further determines the convergence rate. Also we study the loss of accuracy caused by a commonly used relaxation parameter, which is known as "constraint force mixing".**

I. INTRODUCTION

Simulation of multibody dynamics is important in a number of fields in science and engineering, including the area of robotics. The analysis of robots in simulation not only provides meaningful guideline for design real physical robots, but also makes the manufacture process cheaper and more efficient, since modification of design in virtual simulation environment is convenient than on a physical robot. Multibody dynamics plays an important role in simulation because of the involved constraints and contacts problems. When we plan complex robot behaviors such as grasping and manipulation [1], [2], we have to solve the contact dynamics between the robot hands and objects. When it comes to the contact dynamics between a robot and the ground with hundreds of rocks or sands, such as the Cheetah robot from Boston Dynamics and other humanoid robots, we have to solve the dynamics with hundreds or thousands of contacts and constraints. One of our long-term goals is to enhance planning and control of robot locomotion and manipulation through efficient and accurate simulation of dynamic systems with intermittent contact.

A natural way to formulate the dynamic model of a multibody system in intermittent contact is as a system of differential equations and complementarity conditions [3]. Such models are in the class known as differential complementarity problems (dCPs). They are nonsmooth and nonlinear, and thus very difficult to solve. Many numerical algorithms have been proposed to integrate dCPs forward in time to predict motion with contact [4][5][6]. All of them convert the continuous-time dCP into a sequence of nonlinear or linear complementarity problems (NCPs or LCPs), so that simulation becomes equivalent to solving a sequence of CPs. Because the naturally arising CPs do not have desirable numerical properties, many methods regularize the CPs to improve the convergence rate of available CP solvers. Regularization is done for both classes of methods: pivoting and iterative.

In real-time applications with embedded simulators, such as model-predictive control of robots and interactive computer games, long solution times and high variance of solution times is highly undesirable. Similarly, in off-line design and planning problems, solutions should be as fast as possible. These considerations have motivated researchers to provide theoretical guidelines for setting parameters of numerical solvers to give optimal performance. In this paper, we focus on a particular iterative scheme that has gained popularity in the past few years. The method is based on converting the CP formulation into an equivalent system of equations involving proximal point functions, a.k.a, prox functions. This system is then solved by fixed-point iteration.

The prox function iterative method is extremely simple, containing only one parameter, r, that determines the speed of convergence. Theoretical results by Föerg [7] provide a means for choosing the value of r that minimizes the number of iterations to converge. It is based on the eigenvalues of the Delassus matrix, which is constructed from the system inertia and constraint Jacobians. However, the choice of r is not a function of the system initial conditions or external impulse. The goal of this paper is to convince the reader of the important impact that these other factors can have on the convergence of fixed-point solvers for prox function formulations of the simulation subproblems. This is demonstrated through the problem of a particle undergoing inelastic collision with a plane, whose equations of motion, when mapped into the contact frame, are nearly identical to that of a robot colliding with its environment a single point. We show that the number of iterations required for convergence for "bad" choices of external factor can be hundreds of time larger than for "good" choices.

[1] Ying Lu is research assistant with the Department of Computer Science, Rensselaer Polytechnic Institute, 108 eighth street, Troy, NY 12180, USA rosebudflyaway@gmail.com

[2] Jeff Trinkle is with Faculty of Computer Science, Rensselaer Polytechnic Institute, 108 eighth street, Troy, NY 12180, USA trinkle@gmail.com

II. Background

A. Particle Problem on a 3-D Plane

The work in this paper uses the Benchmark Problems for Multibody Dynamics (BPMD) framework [8] as platform, which has been used to collect benchmark multibody problems from several physics engines [9] The problem we choose in this paper is collected from RPISim [10].

The set up of the simulation is displayed in figure 1, where we have a particle (the blue circle) falling from a initial position onto a flat plane (green area). Once the particle collides with the plane, there will be a contact force applied on the particle by the plane. In figure 1, ψ_n is the gap distance between particle and the plane to decide whether there is contact between them. xyz coordinate represent the world frame, while nto coordinate is for the local contact frame.

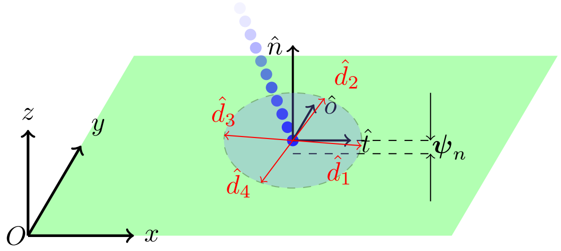

Fig. 1: Simulation Set up of the Particle on a 3-D Plane Problem

The particle with mass m collides with a plane under the action of an external force $\boldsymbol{\lambda}_{\text{ext}}$. The position of the particle is $\mathbf{q} = [q_x, q_y, q_z]^T$ and $\boldsymbol{\nu}$ denotes its velocity. We use the superscript $^\ell$ to represent the value at a certain time, for example $\mathbf{q}^\ell$ is the particle position at the ℓ-th simulation step. When there exists contact between the particle and the plane, the contact force is $\boldsymbol{\lambda} = [\lambda_n, \lambda_t, \lambda_o]^T$. λ_n is the normal contact force and λ_t, λ_o are components of frictional force along the $\hat{t}, \hat{o}$ direction, respectively. So the resultant frictional force is $\boldsymbol{\lambda}_f = \begin{bmatrix}\lambda_t & \lambda_o\end{bmatrix}^T$.

III. Formulation of the "Prox Method"

The equation of motion for particle problem is:

$$\mathbf{M}\dot{\boldsymbol{\nu}} = \mathbf{G}_n\boldsymbol{\lambda}_n + \mathbf{G}_f\boldsymbol{\lambda}_f + \boldsymbol{\lambda}_{\text{ext}} \tag{1}$$

where $\mathbf{M} = m\mathbf{I}$ and $\mathbf{I}$ is the 3×3 identity matrix and $\mathbf{G}_n$, $\mathbf{G}_f$ are the Jacobian matrices that transforms the normal and frictional force from the local nto frame to the global xyz frame, respectively. Let h be the time step, then $\dot{\boldsymbol{\nu}} = \dfrac{\boldsymbol{\nu}^{\ell+1} - \boldsymbol{\nu}^\ell}{h}$, we substitute this into equation (1) and arrive at:

$$\mathbf{M}\boldsymbol{\nu}^{\ell+1} = \mathbf{M}\boldsymbol{\nu}^\ell + \mathbf{G}_n\mathbf{p}_n^{\ell+1} + \mathbf{G}_f\mathbf{p}_f^{\ell+1} + \mathbf{p}_{\text{ext}} \tag{2}$$

where $\mathbf{p}_n = \boldsymbol{\lambda}_n h$ is the normal contact impulse and $\mathbf{p}_f, \mathbf{p}_{\text{ext}}$ are defined as impulses in exactly the same way. In this particle problem, $\mathbf{G}_n$ is:

$$\mathbf{G}_n = \begin{bmatrix}0\\0\\1\end{bmatrix} \tag{3}$$

Before the Jacobian matrix $\mathbf{G}_f$ are defined, we introduce the Coulomb friction law that we adopt in our model.

1) Conic model: We will solve λ_t, λ_o along the $\hat{t}, \hat{o}$ directions, shown in figure 1. Therefore, λ_t and λ_o can be any value either positive or negative. The Jacobian for this model is:

$$\mathbf{G}_f = \begin{bmatrix}1 & 0\\0 & 1\\0 & 0\end{bmatrix} \tag{4}$$

2) Linearized model: The quadratic cone is approximated using a four-sided pyramid. The vectors $\hat{d}_1, \hat{d}_2, \hat{d}_3$ and $\hat{d}_4$ in figure 1 are unit vectors denoting discretized friction directions. Therefore, we have to solve $\lambda_{f1}, \lambda_{f2}, \lambda_{f3}$ and λ_{f4} along the $\hat{d}_1, \hat{d}_2, \hat{d}_3$ and $\hat{d}_4$ directions in figure 1, respectively, and they can only be non-negative values. The Jacobian for this model is:

$$\mathbf{G}_f = \begin{bmatrix}1 & 0 & -1 & 0\\0 & 1 & 0 & -1\\0 & 0 & 0 & 0\end{bmatrix} \tag{5}$$

Once there is contact between the particle and the plane, the "rigid" property of the bodies will prevent them from penetration, the normal constraint is :

$$\mathbf{0} \le \mathbf{p}_n^{\ell+1} \perp \boldsymbol{\psi}_n^{\ell+1} \ge \mathbf{0} \tag{6}$$

where $\perp$ denotes orthogonality between two vectors: $\mathbf{a}, \mathbf{b} \in \mathcal{R}^n$, and $\mathbf{a} \perp \mathbf{b}$ means $\mathbf{a}^T\mathbf{b} = 0$.

The frictional constraints are (the detailed derivation is in Berard's thesis [11]):

$$\mathbf{0} \le \mathbf{p}_f^{\ell+1} \perp \boldsymbol{\rho}_f^{\ell+1} \ge \mathbf{0} \tag{7}$$

$$\mathbf{0} \le \mathbf{s}^{\ell+1} \perp \boldsymbol{\sigma}^{\ell+1} \ge \mathbf{0} \tag{8}$$

where

$$\boldsymbol{\rho}_f^{\ell+1} = \mathbf{G}_f^T\boldsymbol{\nu}^{\ell+1} + \mathbf{E}\mathbf{s}^{\ell+1} \tag{9}$$

$$\boldsymbol{\sigma}^{\ell+1} = \mathbf{U}\mathbf{p}_n^{\ell+1} - \mathbf{E}^T\boldsymbol{\alpha}^{\ell+1} \tag{10}$$

where $\boldsymbol{\alpha}$ is the nonnegative friction magnitude along each discretized direction, $\mathbf{E}$ is a vector of 1s with its dimension equal to the number of friction directions and $\mathbf{U}$ is the frictional coefficient between the particle and the plane, s is a vector of the sliding speed along each of the friction directions. $\boldsymbol{\rho}_f^{\ell+1}$ is the vector of the predicted sliding speed at each of the frictional direction at the $(\ell+1)$-th step and $\boldsymbol{\sigma}^{\ell+1}$ is a non-negative slack variable denoting whether it is sticking or sliding case at the contact at the $(\ell+1)$-th step.

Let n_d be the number of frictional directions, which is the number of facets in the polyhedral to approximate the frictional cone in LCP. Then in this particle problem, if we use Nonlinear Complementarity Problem (NCP) formulation, we have to solve two friction components along two perpendicular directions $\hat{t}$ and $\hat{o}$, so $n_d = 2$, if we use Linear Complementarity Problem (LCP) formulation, $n_d = 4$. $\mathbf{p}_f^{\ell+1}$ is the frictional component along the n_d friction directions at the $(\ell + 1)$-th time step.

Once we apply Taylor expansion to $\boldsymbol{\psi}_n^{\ell+1}$ in equation (6), we will arrive at the following equations:

$$\begin{bmatrix} 0 \\ \boldsymbol{\rho}_n^{\ell+1} \\ \boldsymbol{\rho}_f^{\ell+1} \\ \mathbf{s}^{\ell+1} \end{bmatrix} = \begin{bmatrix} -\mathbf{M} & \mathbf{G}_n & \mathbf{G}_f & \mathbf{0} \\ \mathbf{G}_n^T & \zeta\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{G}_f^T & \mathbf{0} & \zeta\mathbf{I} & \mathbf{E} \\ \mathbf{0} & \mathbf{U} & -\mathbf{E}^T & \zeta\mathbf{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\nu}^{\ell+1} \\ \mathbf{p}_n^{\ell+1} \\ \mathbf{p}_f^{\ell+1} \\ \mathbf{s}^{\ell+1} \end{bmatrix} + \begin{bmatrix} -\mathbf{M}\boldsymbol{\nu}^{\ell} - \mathbf{p}_{\text{ext}} \\ \dfrac{\boldsymbol{\psi}_n^{\ell}}{h} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{11}$$

where $\boldsymbol{\rho}_n^{\ell+1}$ is the predicted gap distance at the $(\ell + 1)$-th step, ζ is the relaxation parameter, which is a small positive number and makes the problem easier to solve. When we choose $\zeta = 0$, then equation (11) represents the original problem without using relaxation metric. $\mathbf{I}$ is the identity matrix with compatible dimension.

If we substitute the expression of $\boldsymbol{\nu}^{\ell+1}$ in equation (2) into (6)-(8). We will arrive at the following standard LCP form:

$$\mathbf{0} \leq (\mathbf{A}\mathbf{z} + \mathbf{b}) \perp \mathbf{z} \geq \mathbf{0} \tag{12}$$

where $\mathbf{A}$, $\mathbf{b}$ and $\mathbf{z}$ are defined as:

$$\mathbf{A} = \begin{bmatrix} \mathbf{G}_n^T\mathbf{M}^{-1}\mathbf{G}_n & \mathbf{G}_n^T\mathbf{M}^{-1}\mathbf{G}_f & \mathbf{0} \\ \mathbf{G}_f^T\mathbf{M}^{-1}\mathbf{G}_n & \mathbf{G}_f^T\mathbf{M}^{-1}\mathbf{G}_f & \mathbf{E} \\ \mathbf{U} & -\mathbf{E}^T & \mathbf{0} \end{bmatrix} \tag{13}$$

$$\mathbf{b} = \begin{bmatrix} \mathbf{G}_n^T(\boldsymbol{\nu} + \mathbf{M}^{-1}\mathbf{p}_{\text{ext}}) + \dfrac{\boldsymbol{\psi}_n}{h} \\ \mathbf{G}_f^T(\boldsymbol{\nu} + \mathbf{M}^{-1}\mathbf{p}_{\text{ext}}) \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} \mathbf{p}_n \\ \mathbf{p}_f \\ \mathbf{s} \end{bmatrix}. \tag{14}$$

The NCP formulation takes the form:

$$\boldsymbol{\nu}^{\ell+1} = \boldsymbol{\nu}^{\ell} + \mathbf{M}^{-1}\left(\mathbf{G}_n\mathbf{p}_n^{\ell+1} + \mathbf{G}_f\mathbf{p}_f^{\ell+1} + \mathbf{p}_{\text{ext}}\right) \tag{15}$$

$$0 \leq \rho_n^{\ell+1}(p_n^{\ell+1}) \perp p_n^{\ell+1} \geq 0 \tag{16}$$

$$\mu p_n \nu_t + p_t s = 0 \tag{17}$$

$$\mu p_n \nu_t + p_o s = 0 \tag{18}$$

$$0 \leq s \perp \sigma \geq 0 \tag{19}$$

where ν_t, ν_o are the tangential velocity components along $\hat{t}, \hat{o}$, p_t, p_o are friction impulse components along $\hat{t}, \hat{o}$, and ρ_n and σ in this NCP formulation are defined:

$$\rho_n = \mathbf{G}_n^T\boldsymbol{\nu}^{\ell+1} + \zeta\mathbf{p}_n^{\ell+1} \tag{20}$$

$$\sigma = (\mu p_n)^2 - p_t^2 - p_o^2. \tag{21}$$

The complementarity conditions can be written as equivalent nonsmooth functions [9]; one of them is the prox function [12]. If we have the complementarity problem as $\boldsymbol{\rho}(\mathbf{p}) \perp \mathbf{p}$, the equivalent proximal function is:

$$\phi_{\text{prox}}(\mathbf{p}, \boldsymbol{\rho}, r) = \mathbf{p} - \text{prox}_{\mathcal{C}}(\mathbf{p} - r\boldsymbol{\rho}(\boldsymbol{p})) = \mathbf{0} \tag{22}$$

The choice of parameter r plays an important role in the convergence of fixed-point iteration method with proximal function. Reformulating equation (22) and adding the time step index as superscript, we arrive at:

$$\mathbf{p}^{\ell+1} = \text{prox}_{\mathcal{C}}(\mathbf{p}^{\ell} - r\boldsymbol{\rho}(\mathbf{p}^{\ell})) \tag{23}$$

where $\mathcal{C}$ is the feasible set of $\mathbf{p}$. So here for the normal impulse at contact i, the feasible set is:

$$\mathcal{C}_n = \{\mathbf{p}_n \mid \mathbf{p}_n \geq 0\} \tag{24}$$

And the feasible set for friction impulse at contact i is:

$$\mathcal{C}_f = \{\mathbf{p}_f \mid \ \parallel \mathbf{p}_f \parallel \leq p_n\} \tag{25}$$

where $\mathbf{p}$ stands for the contact impulse. Based on the definition of feasible set, when the term $(\mathbf{p} - r\boldsymbol{\rho}(\mathbf{p}))$ is inside the feasible set, then $\text{prox}_{\mathcal{C}}(\mathbf{p} - r\boldsymbol{\rho}(\mathbf{p})) = \mathbf{p} - r\boldsymbol{\rho}(\mathbf{p})$; Otherwise, it will be projected onto the nearest point on the boundary of the feasible set. We summarize the projection rule into a formula:

$$\text{prox}_{\mathcal{C}}(\mathbf{x}) = \begin{cases} \mathbf{x} & \text{if } \mathbf{x} \in \mathcal{C} \\ \mathbf{d}^*, \mathbf{d}^* \text{ solves } \ \min_{\mathbf{d} \in \partial \mathcal{C}} ||x - \mathbf{d}|| & \text{if } \mathbf{x} \notin \mathcal{C} \end{cases}$$

The equivalence and difference between the proximal function and the complementarity condition has been proved by Schindler and Trinkle [13].

When we rewrite the complementarity condition in the form of equation (23), then we run one step of fixed-point iteration method (we use iterator k to denote the iterations for fixed-point iteration method, to separate from ℓ for time step in simulation), then we have:

$$\boldsymbol{\nu}^{k+1} = \boldsymbol{\nu}^{k} + \mathbf{M}^{-1}\left(\mathbf{G}_n\mathbf{p}_n^{k} + \mathbf{G}_f\mathbf{p}_f^{k} + \mathbf{p}_{\text{ext}}\right) \tag{26}$$

$$\mathbf{p}_n^{k+1} = \text{prox}_{\mathcal{C}_n}(\mathbf{p}_n^{k} - r\boldsymbol{\rho}_n^{k}(\mathbf{p}_n^{k})) \tag{27}$$

$$\mathbf{p}_f^{k+1} = \text{prox}_{\mathcal{C}_f}(\mathbf{p}_f^{k} - r\boldsymbol{\rho}_f^{k}(\mathbf{p}_f^{k})) \tag{28}$$

All of these equations have the form of:

$$\mathbf{x}^{\ell+1} = \mathbf{f}(\mathbf{x}^{\ell}) \tag{29}$$

where $\mathbf{x}^{\ell+1}$ are unknowns to solve (it contains $\boldsymbol{\nu}, \mathbf{p}_n, \mathbf{p}_f$). This form is friendly to fixed-point iteration, which is a method of computing fixed-point of iteration functions. Furthermore, to evaluate the performance of fixed-point iteration method with different parameters in a standard way, we define the error based on the Chen-Chen-Kanzow function [14]:

$$\phi_{\text{CCK}i}(p_i, \xi) = \xi\left(p_i + \rho_i - \sqrt{p_i^2 + \rho_i^2}\right) + (1 - \xi)\, p_i^+ \rho_i^+ \tag{30}$$

$$totalError = \frac{1}{2}\boldsymbol{\phi}_{\text{CCK}}^T(\mathbf{p}, \boldsymbol{\rho}(\mathbf{p}), \xi)\, \boldsymbol{\phi}_{\text{CCK}}(\mathbf{p}, \boldsymbol{\rho}(\mathbf{p}), \xi) \tag{31}$$

where $\xi = 0.7$. We set the convergence tolerance to 10^{-6}. Once the error is less than the tolerance, the iteration is terminated and the flow returns from the fixed-point iteration method to continue to the next simulation time step. The pseudo code of one time step with fixed-point iteration

```
Data: $\boldsymbol{\nu}^\ell$ : generalized velocity at $\ell$-th step
      $p_n^\ell$ : normal contact impulse at $\ell$-th step
      $\mathbf{p}_f^\ell$ : frictional impulse at $\ell$-th step
      $toler$: tolerence
      $maxIter$: maximum number of iterations
Result: $\boldsymbol{\nu}^{\ell+1}$ : generalized velocity at $(\ell+1)$-th step
        $p_n^{\ell+1}$ : normal contact impulse at $(\ell+1)$-th step
        $\mathbf{p}_f^{\ell+1}$ : frictional impulse at $(\ell+1)$-th step
        $iter$ : number of iterations used to converge
Initialization :
$k \leftarrow 0$
$\boldsymbol{\nu}^k \leftarrow \boldsymbol{\nu}^\ell$ :
$p_n^k \leftarrow p_n^\ell$
$\mathbf{p}_f^k \leftarrow \mathbf{p}_f^\ell$
$totalErr = 10^{16}$
/*The main loop*/
while $totalErr \geq toler\ and\ k \leq maxIter$ do
    $\boldsymbol{\nu}^{k+1} = \boldsymbol{\nu}^k + \mathbf{M}^{-1}\mathbf{G}_n\mathbf{p}_n^k + \mathbf{M}^{-1}\mathbf{G}_f\mathbf{p}_f^k + \mathbf{M}^{-1}\mathbf{p}_{\text{ext}}$
    $p_n^{k+1} = p_n^k - r\rho_n^k(p_n^k)$
    if $p_n^{k+1} < 0$ then
        $p_n^{k+1} = 0$
    end
    $\mathbf{p}_f^{k+1} = \text{prox}_{\mathcal{C}_f}(\mathbf{p}_f^k - r\boldsymbol{\rho}_f^k(\mathbf{p}_f^k))$
    if $\parallel \mathbf{p}_f^{k+1} \parallel > \mu p_n^{k+1}$ then
        $p_f^{k+1} = \frac{\mathbf{p}_f^{k+1}}{\parallel \mathbf{p}_f^{k+1} \parallel}\mu p_n^{k+1}$
    end
    /* Update $totalErr$ */
    $totalErr = \frac{1}{2}\phi_{\text{CCK}}^T(\mathbf{p}, \boldsymbol{\rho}(\mathbf{p}), \xi)\, \phi_{\text{CCK}}(\mathbf{p}, \boldsymbol{\rho}(\mathbf{p}), \xi)$
    $k \leftarrow k+1$
end
$\boldsymbol{\nu}^{\ell+1} \leftarrow \boldsymbol{\nu}^{k+1}$
$p_n^{\ell+1} \leftarrow p_n^{k+1}$
$\mathbf{p}_f^{\ell+1} \leftarrow \mathbf{p}_f^{k+1}$
$iter \leftarrow k$
```

Algorithm 1: Fixed-point iteration method with prox function formulation

method with only one contact point is shown in Algorithm 1.

Since the parameter r in the proximal function plays an important role on the convergence of the fixed-point iteration method, we will provide analysis on parameter choice for fixed-point iteration method. How the change in vector $\mathbf{b}$ will influence the convergence of the solver as well as how to choose the parameter by analyzing the property of matrix $\mathbf{A}$. We present the problem and manually formulate the matrix and set the parameter, following Föerg's work [7] as our analytical criterion for selecting parameter r.

IV. RESULTS

Föerg et al [7] has provided some theoretical guidance on how the choice of r will affect the convergence of the solvers. In Föerg's work, the choice of r totally depends on the property of matrix $\mathbf{A}$ in equation (13), and he doesn't mention any relevance with the vector $\mathbf{b}$ [7]. However, we find cases where the convergence of the solver depends strongly on the vector $\mathbf{b}$ in equation (14) when the recommend r value is used. While the property of a matrix can be positive semi-definite (PSD), the property of a vector is more general and has drawn little attention, especially in range of complementarity problems. This simple particle example shows that the choice of r should depend on $\mathbf{b}$.

A. Analytical Choice for Parameter r

Föerg [7] claims the optimal choice of r to be:

$$r^* = \frac{2}{\eta_{\max} + \eta_{\min}} \tag{32}$$

where $\eta_{\max}$ and $\eta_{\min}$ are the maximum and minimum eigenvalues of the Delassus matrix $\mathbf{A}$. In our simulation, the mass of the particle is $m = 1$, the initial position is $\mathbf{q} = [1, 1, 0.15]^T$, so the initial gap distance $\psi_n = 0.15$. The initial velocity is $\boldsymbol{\nu} = [1, 0, -1]^T$, to make the matrix simple to compute manually, we choose the acceleration of gravity $g = -10$ (i.e. $\boldsymbol{\lambda}_{\text{ext}} = \begin{bmatrix} 0 & 0 & g \end{bmatrix}^T$) and the time step $h = 0.1$. Here we consider the frame where the particle has no contact with the plane at the current time step but will at the next one. The frictional coefficient between particle and plane is $\mu = 0.5$. To substitute these known variables into the matrix $\mathbf{A}$ in equation (13) and we get:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1+\zeta & 0 & -1 & 0 & 1 \\ 0 & 0 & 1+\zeta & 0 & -1 & 1 \\ 0 & -1 & 0 & 1+\zeta & 0 & 1 \\ 0 & 0 & -1 & 0 & 1+\zeta & 1 \\ 0.5 & -1 & -1 & -1 & -1 & \zeta \end{bmatrix} \tag{33}$$

where ζ is the relaxation parameter. If there is no relaxation, i.e. $\zeta = 0$, the maximum and minimum eigenvalue of matrix $\mathbf{A}$ defined in equation (33) is $\eta_{\max} = 2$ and $\eta_{\min} = 0$, so the optimal $r^* = 1$. This will be verified in the following results. However, when we add a positive value to matrix $\mathbf{A}$, the maximum eigenvalue is not changed: $\eta_{\max} = 2$, the minimum eigenvalue becomes $\eta_{\min} = \zeta$ due to the relaxation. The optimal r^* will change correspondingly.

B. Numerical Choices for Parameter r

Based on the theoretical guidance for parameter choices in the previous section, we run the "same" particle problem in simulation. To investigate the convergence trend of the solver, we only test one time step at the iterative level, which is the frame that the particle is going to collide the plane. To demonstrate the dependence of solver convergence on the vector $\mathbf{b}$, the external force is regarded as parameter by changing the acceleration of gravity, which is exactly the same as changing externally applied forces.

In figure 2, the horizontal axis represents the value of parameter r, and the vertical axis stands for the number of iterations it takes to solve the problem to the desired tolerance. The lines with different colors represent the various external forces, corresponding to different vector $\mathbf{b}$. In this

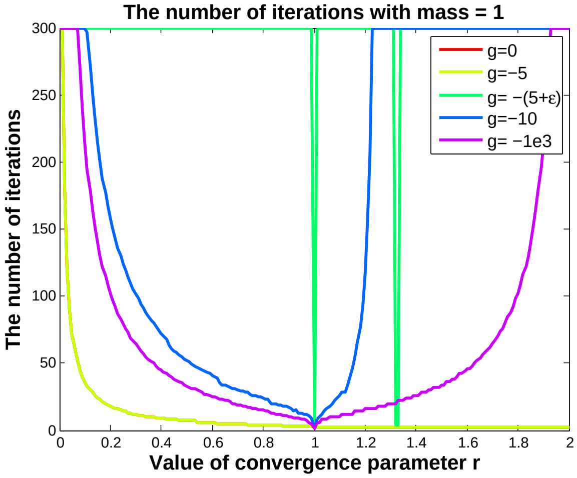

Fig. 2: Number of iterations versus r parameter for different external forces with mass $m = 1$

plot, when $r = 1$, which is the optimal value, the solver converges to the solution using only 1 iteration. However, as r is larger or smaller, the number of iterations will increase. For example, when $g = -10$, the optimal parameter $r = 1$, as r grows to 1.25, the number of iterations reaches 300, which is the maximum number we allowed. So we still don't know whether we converged at the tolerance accuracy since the simulation might terminate because the maximum iteration number is reached.

Another interesting phenomenon in figure 2 is: when we set $g = -5$, the particle touches the plane at the end of this time step, with contact impulse equal to zero. For $g > -5$, the contact impulse is zero and the number of iterations is small except as r approaches zero. When $g \ll -5$, a large contact impulse occurs and values of r near the optimal value perform well. However, when $g = -(5 + \epsilon)$, where $\epsilon \ll 1$, the convergence is very poor except for two values $r = 1$ and $r = 1.3$.

When it comes back to the analytical solution, we find that when $g = -5$, the particle touches the plane right at end of this time step and there is no contact impulse between the particle and the plane. This matches well with the results from numerical experiment. If we increase the external force to $g = -(5 + \epsilon)$, the small change will cause the normal contact force to be positive to prevent penetration between the particle and the plane. The initial velocity of the particle has a tangential component as $\nu_x = 1$, which will result in the sliding case when the particle collides with the plane. The sliding case is harder to solve and makes the choices of parameter r in a very narrow range.

C. Limitation of relaxation

If the relaxation ζ is set to zero, then the constraint will be exactly the same as the original and it is hard to solve. However, if we choose ζ to be a positive value (usually a small positive number), then the property of matrix $\mathbf{A}$ will be changed and the problem is easier to solve. But it will be possible to violate the constraint, which means the constraints will become compliant. This relaxation approach has been popular in a number of physics engines such as Bullet and ODE.

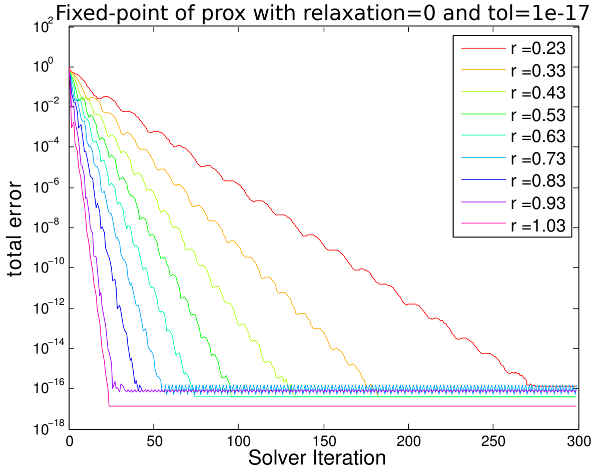

Fig. 3: The total error for different r values with $\zeta = 0$ at tolerance of 10^{-16}, with mass $m = 1$, acceleration of gravity is $g = -10$.

In this particle problem, we tested this metric by comparing different ζ values and we found that the solver can never converge to an accuracy less than the order of the relaxation value. This is important for the relaxation parameter choice in simulations since if we want to achieve high accuracy, we cannot set a relaxation parameter larger than the tolerance, otherwise, we can never terminate the solver until the maximum allowed iterations is achieved.

Figure 3 shows the results for fixed-point iteration method, based on the proximal function, without any relaxation: $\zeta = 0$, the tolerance is set to 10^{-17}. The total error is computed using formula in equation (31). The acceleration of gravity is firm: $g = -10$ and the maximum number of iterations is set to 300. Once our solver reaches the maximum iteration, or the error of the solution drops below the tolerance, the iterative process will terminate. For this case, we find that as r goes close to 1, which is the optimal parameter, the solver will converges faster. However, as r goes farther away from $r^* = 1$, the total error is at an order of 10^{-16}, which is actually the machine precision of MATLAB computing environment. This is caused by round off errors with more iterations to converge to the solution.

Figure 4 shows the results for the same problem as that in figure 3, except that the relaxation parameter is set to $\zeta = 10^{-12}$. Then we will have $\eta_{\max} = 2$ and $\eta_{\min} = \zeta = 10^{-12}$, the optimal $r^* = 0.9999999999995$, which is very close to 1. It shows in the plot that the experiment with r more close to 1 converges faster (using less iteration numbers). However, the total error is at an order of 10^{-12}, even though the tolerance is still 10^{-17}, this means the solver terminates when the maximum number of iterations is used up. The total error is right at the same order of the relaxation parameter. Moreover, since 10^{-12} is larger than the MATLAB machine epsilon here, so all the cases with different r converges to the same

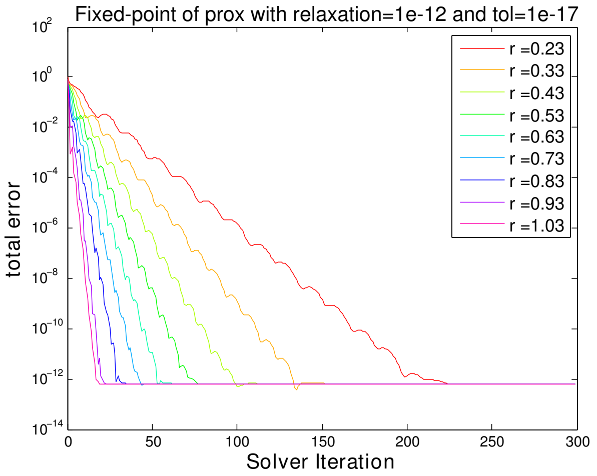

Fig. 4: The total error for different r values with $\zeta = 10^{-12}$ at tolerance of 10^{-16}, with mass $m = 1$, acceleration of gravity is $g = -10$.

total error, no matter how many iterations it takes.

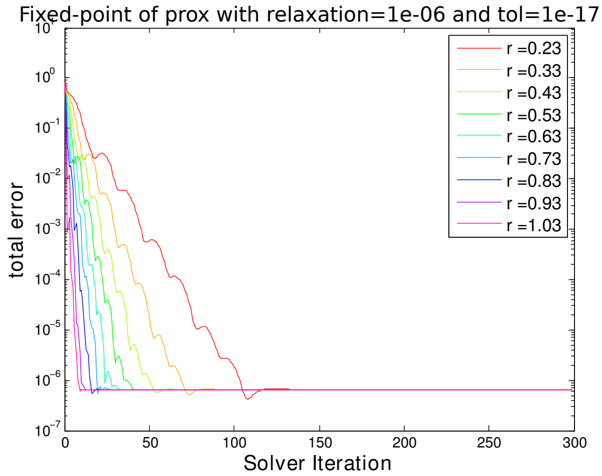

Fig. 5: The total error for different r values with $\zeta = 10^{-6}$ at tolerance of 10^{-16}, with mass $m = 1$, acceleration of gravity is $g = -10$.

Figure 5 shows the results for the same problem as that in figure 3, except that the relaxation parameter is set as $\zeta = 10^{-6}$. Then we will have $\eta_{\max} = 2$ and $\eta_{\min} = \zeta = 10^{-6}$, the optimal $r^* = 0.99999950000025$, which is very close to 1. The tolerance is constant as 10^{-17} and the total error is never less than about 10^{-6}.

In conclusion, the solver can not converge at an accuracy that is smaller than ζ. So if we want an accurate solution for complementarity problems, then we have to set the relaxation value to a number smaller than the accuracy we want. But one reason we still consider adding relaxation is that the metric really makes the problem easy to solve and when we emphasize on the computational efficiency more than accuracy, we can still adopt this relaxation metric.

V. CONCLUSIONS

In this paper, we demonstrated the effects parameter choices on solving complementarity problems by fixed-point iteration using proximal point functions. We verified that the available theoretical guidelines for setting the convergence parameter, but our main finding is that the theory does not cover an important case that can affect the convergence rate by several orders of magnitude. We also presented in a quantitative way how the relaxation parameter affects the final accuracy at convergence.

REFERENCES

[1] E. Todorov, "A convex, smooth and invertible contact model for trajectory optimization." in *ICRA*. IEEE, 2011, pp. 1071–1076.

[2] M. Posa, M. Tobenkin, and R. Tedrake, "Lyapunov analysis of rigid body systems with impacts and friction via sums-of-squares," in *Proceedings of the 16th International Conference on Hybrid Systems: Computation and Control*, ser. HSCC '13. New York, NY, USA: ACM, 2013, pp. 63–72.

[3] J. Trinkle, J. shi Pang, S. Sudarsky, and G. Lo, "On dynamic multi-rigid-body contact problems with coulomb friction."

[4] A. Fischer, "A newton-type method for positive-semidefinite linear complementarity problem," *Journal of Optimization Theory and Applications*, vol. 86, pp. 585–608, 1995.

[5] N. Chakraborty, S. Berard, S. Akella, and J. Trinkle, "An implicit time-stepping method for multibody systems with intermittent contact," in *In Robotics Science and Systems*, 2007.

[6] E. Todorov, "Implicit nonlinear complementarity: A new approach to contact dynamics." in *ICRA*. IEEE, 2010, pp. 2322–2329.

[7] M. Forg, T. Geier, L. Neumann, and H. Ulbrich, "R-factor strategies for the augmented lagrangian approach in multi-body contact mechanics," III European Conference on COmputational Mechanics, Lisbon, Portugal, June 2006.

[8] Y. Lu, J. Williams, C. Lacoursiere, and J. Trinkle, "A framework for problem standardization and algorithm comparison in multibody system," in *ASME International Design and Engineering Technical Conferences and Computers and Information in Engineering*, August 2014.

[9] C. Lacoursiere, Y. Lu, J. Williams, and J. Trinkle, "Standard interface for data analysis of solvers in multibody dynamics," in *Canadian Conference on Nonlinear Solid Mechanics (CanCNSM)*, July 2013.

[10] J. Williams, Y. Lu, S. Niebe, M. Andersen, K. Erleben, and J. Trinkle, "Rpi-matlab-simulator: A tool for efficient research and practical teaching in multibody dynamics," in *Workshop on Virtual Reality Interaction and Physical Simulation (VRIPHYS)*, November 2013.

[11] S. G. Berard, "Using simulation for planning and design of robotics systems with intermittent contact," Ph.D. dissertation, Rensselaer Polytechnic Institute, 2009.

[12] R. I. Leine and H. Nijmeijer, *Dynamics and Bifurcations of Non-Smooth Mechanical Systems*. Academic Press, 1992.

[13] T. Schindler, B. Nguyen, and J. Trinkle, "Understanding the difference between prox and complementarity formulations for simulation of systems with contact," *IROS, IEEE*, pp. 1433–1438, 2011.

[14] B. Chen, X. Chen, and C. Kanzow, "A penalized fischer-burmeister ncp-function: Theoretical investigation and numerical results," 1997.