On the Use of Iterative LCP Solvers for Dry Frictional Contacts in Grasping

Gustavo Arechavaleta, Efraín López-Damian and José Luis Morales.

***Abstract*— In this paper we propose the use of new iterative methods to solve symmetric linear complementarity problems (SLCP) that arise in the computation of dry frictional contacts in Multi-Rigid-Body Dynamics. Specifically, we explore the two-stage iterative algorithm developed by Morales, Nocedal and Smelyanskiy [1]. The underlying idea of that method is to combine projected Gauss-Seidel iterations with subspace minimization steps. Gauss-Seidel iterations are aimed to obtain a high quality estimation of the active set. Subspace minimization steps focus on the accurate computation of the inactive components of the solution. Overall the new method is able to compute fast and accurate solutions of severely ill-conditioned LCPs.**

We compare the performance of a modification of the iterative method of Morales *et al* with Lemke's algorithm on robotic object grasping problems.

I. INTRODUCTION

Contact dynamics is a well-known and important topic in physics-based animation and robotics among other disciplines. Video games, for example, are a good source of medium to large-scale contact problems. The common issue in these fields is the fast computation of contacts. Constraints, however, strongly characterize the application. In computer animation the constraints are mainly intended to produce visually plausible motions. Thus, simple friction models suffice to preserve realism in the animation. In robotics, say for problems regarding stable biped locomotion and grasping tasks, contact constraints must guarantee feasible robot motions in the presence of friction [2].

It turns out that solving contact problems is a time-consuming process mainly because at each time step of the simulation, performed by numerical integration, a linear complementarity problem has to be solved.

In this paper we address the problem of computing multi-rigid-body dynamics with frictional contacts. Specifically, we explore the use of the algorithm described in [1] to determine the contact forces in the presence of friction between colliding rigid bodies.

In addition to the frictionless contact problem, in a wide variety of applications it is desirable to consider a friction model (e.g. the Coulomb's friction model). This fact increases the difficulty of the problem since there is coupling between the friction (tangential) and normal force components. In practice, the most common and widely used model of friction is the linearized version of the Coulomb friction law by a polyhedral approximation of the friction cone at each contact point [3], [4]. Nevertheless, there exist other ways for approximating the Coulomb friction model [5].

A consistent and solvable mathematical model of dry frictional contacts has been proposed in [3]; for variants based on LCP formulations see for example [4], [6], [7]. Alternative strategies include spring and damper systems [8], although they tend to be unstable.

Among the wide range of LCP contact formulations either acceleration based models [9], [10], [11], [12], or velocity stepping models [4], [6], [3], the most common method to solve the LCP is still that of Lemke [13], [14]. Pivotal methods are efficient for small problems but they do not exploit sparsity effectively and may suffer from numerical instability. Therefore they are not suitable for large-scale applications, or when the LCPs are ill-conditioned; see for example [15]. In the robotics literature, some improvements on Lemke's algorithm have been proposed in [16], [17] to overcome some of its inherent deficiencies.

Iterative LCP solvers, including the projected Gauss-Seidel (PGS) algorithm, have been utilized to compute frictional contacts problems (e.g. [18], [19], [20]) since they are numerically robust. However, they can be very slow for medium to large-scale and ill-conditioned problems. In grasping tasks, the contact LCPs are ill-conditioned in most of the cases. These evidences motivated us to use a modification of the algorithm described in [1]. In the forthcoming discussion we will refer to this method as PGS-SM.

This paper is organized as follows. In Section II we recall the general form of a SLCP. Section III gives a short description of the two stages of PGS-SM. In Section IV we review the standard mixed LCP formulation for multi-rigid-body dynamics with dry frictional contacts. We also describe the approximation of the friction model used to formulate the SLCP contact problem. Section V presents a series of numerical experiments and simulations that illustrate the effectiveness of PGS-SM. Finally, we provide in Section VI some concluding remarks.

II. STRUCTURE OF THE SLCP

The form of a SLCP that will be used in the paper is:

$$0 \leq u \perp Au + a \geq 0 \tag{1}$$

where the unknown of the problem is u, the matrix A and the vectors a are given. We assume that A is $n \times n$ symmetric positive definite.

G. Arechavaleta. Robotics and Advanced Manufacturing Division, CINVESTAV, Mexico. garechav@cinvestav.edu.mx

E. López-Damian. Department of Computer Science, INAOE, Enrique Erro No. 1, Tonantzintla, Puebla, Mexico. eldamian@inaoep.mx

J. L. Morales. Department of Mathematics, ITAM, Río Hondo No. 1, Tizapán, Mexico. jmorales@itam.mx

We recall that (1) corresponds to the first-order optimality conditions of the bound constrained quadratic program

$$\begin{aligned} \min_u \ & \phi(u) = \frac{1}{2} u^T A u + a^T u \\ \text{s.t. } & u \geq 0. \end{aligned} \tag{2}$$

The optimization literature on methods for solving bound constrained problems is wide. Gradient projection methods and interior-point methods are among the best techniques to solve this kind of problems. However, the computational experience reported in [1] indicates that if speed is a hard constraint then PGS-SM is a strong competitor.

III. The Iterative Algorithm

The underlying idea of the iterative algorithm is to alternate two strategies. First, a few iterations of the projected Gauss-Seidel (PGS) method are applied to identify an estimate of the optimal active set. Then, the algorithm performs consecutive subspace minimization (SM) steps to improve this estimate and to compute accurate values of the variables in the inactive set. The algorithm returns to the PGS phase to check the convergence tests, if satisfied the algorithm exits with the solution; otherwise it continues to the SM phase.

A. Projected Gauss-Seidel Method

We briefly outline one iteration of the PGS iterative method. The $k+1$-th approximation can be defined in scalar form as follows

Algorithm PGS. Given the k-th approximation $u^k \geq 0$.

for $i = 1, 2, ..., n$

$$\begin{aligned} \Delta u_i &= A_{ii}^{-1}(a_i + \textstyle\sum_{j<i} A_{ij} u_j^{k+1}) \\ u_i^{k+1} &= \max(0, u_i^k - \Delta u_i,). \end{aligned} \tag{3}$$

end

Observe that, since A is symmetric positive definite, the iteration is well defined.

B. The PGS-SM Method

The algorithm can be formally described as follows.

Algorithm PGS-SM. Choose k_{PGS}, k_{SM} the maximum number of PGS and subspace minimization iterations respectively; choose $tol > 0$; choose an initial approximation $u^0 \geq 0$.

Begin

- **PGS-stage**.
  Perform k_{PGS} iterations of PGS.
  if $||u^{k+1} - u^k||_\infty \leq tol$ **then** exit
  otherwise
  - Set $u \leftarrow u^{k+1}$
  - Form the approximation to the active set

$$\mathcal{A} = \{\, i \mid u_i = 0 \,\}$$

  - Form the corresponding reduced problem

$$\hat{A}\hat{u} + \hat{a} = 0, \quad \hat{u} \geq 0 \tag{4}$$

  by eliminating from the original problem (1) rows and columns whose indices belong to $\mathcal{A}$.

- **SM-stage**.
  Set $k \leftarrow 0$.
  while $k < k_{\mathrm{SM}}$
  1) Solve the linear system $\hat{A}\hat{u} + \hat{a} = 0$ with any method suitable for symmetric positive matrices.
  2) **if** $\hat{u} \geq 0$ **then** return to **PGS-stage**.
     otherwise
     - Compute the projection $\hat{u} \leftarrow \max(0, \hat{u})$.
     - Form a new reduced system $\hat{A}\hat{u} + \hat{a} = 0$.
     - Set $\leftarrow k + 1$.

  end
  Return to **PGS-stage**

End

Observe that, due to the equivalence between the SLCP and the bound constrained optimization problem, all reduced problems are well defined.

IV. The Dry Frictional Contact SLCP

We apply the multi-rigid-body contact formulation described in [3], [2], whose modeling strategy allows both unilateral (contacts) and bilateral (joints) constraints between rigid bodies. This is a time-stepping scheme, based on velocity, that makes use of an approximation of the Coulomb friction law.

A. Dynamic contact Model

We start from the Newton equation of motion

$$M\dot{v} = F_c + F_e, \tag{5}$$

where M is a block-diagonal symmetric positive definite matrix that collects the mass and rotational inertia of each body; $\dot{v}$ represents the acceleration; $F_c = f_j + f_l + f_n + f_t$ is the sum of joint, joint limit, normal and friction constraint forces. F_e represents both Coriolis and external forces.

To solve the multi-body dynamics it is common to use an explicit Euler integration method over a time step from t to $t+1$ for the equation of motion

$$M(v^{t+1} - v^t) = h(F_c + F_e), \tag{6}$$

where h is the integration step length. We consider pairwise constraints between rigid bodies. The non-penetration constraints and frictional forces are expressed as inequalities that can be used to formulate the problem with complementarity conditions. These conditions are defined as

$$0 \leq \hat{N}^T v^{t+1} \quad \perp \quad \hat{c_n} \geq 0 \tag{7}$$

$$0 \leq \hat{D}^T v^{t+1} + E\hat{\lambda} \quad \perp \quad \hat{\beta} \geq 0 \tag{8}$$

$$0 \leq \hat{\mu}\hat{c_n} - E^T\hat{\beta} \quad \perp \quad \hat{\lambda} \geq 0 \tag{9}$$

where $\hat{c_n} = [c_n^{(1)} \ ... \ c_n^{(n)}]^T$ and $\hat{\beta} = [\beta^{(1)} \ ... \ \beta^{(nd)}]^T$ are the normal and friction force components; n is the number of contacts; d is the number of friction directions at contact j. $\hat{N} = [N^{(1)} \ ... \ N^{(n)}]$ and $\hat{D} = [D^{(1)} \ ... \ D^{(nd)}]$ are the constraint matrices for normal and friction forces. Each

column of $\hat{N}$ denotes the contact normal; d columns of $\hat{D}$ span the tangent space of friction forces at contact j; $\hat{\lambda} = [\lambda^{(1)} \ ... \ \lambda^{(n)}]^T$ is a vector of Lagrange multipliers; $\hat{\mu} = \text{diag}[\mu^{(1)} \ ... \ \mu^{(n)}]$ is a diagonal matrix of friction coefficients; E is a matrix defined as

$$E = \text{diag}[\mathbf{1}], \quad \mathbf{1} = [1 \ ... \ 1]^T \in \mathbb{R}^d.$$

The condition (7) serves to avoid interpenetration. The equations (8) and (9) are the complementary conditions of the linearized cone of Coulomb friction law $||f_t|| \leq \mu ||f_n||$.

B. Joint Constraints

As we will show in Section V, we have integrated the PGS-SM algorithm into the open source package, *Graspit* [21]. The current representation of joints between two connected rigid bodies has the form of equality constraints on their relative velocities. Although, given that we do not constrain the relative position of the bodies, the joint connections occasionally break over time due to small drifts in the system. This is corrected by applying

$$\begin{array}{l} \hat{J}^T v^{t+1} = \epsilon_j \\ \epsilon_j = -\zeta \frac{\Delta P}{h}, \end{array} \tag{10}$$

where $\hat{J} = [J^{(1)} \ ... \ J^{(m)}]$; m is the number of joints; ΔP represents the positional errors between connected bodies; ζ takes values in $[0, 1]$ for controlling the error corrections. We then define the constraint impulses as follows

$$hf_j = \hat{J}\hat{c_j}, \tag{11}$$

where $\hat{c_j} = [c_j^{(1)} \ ... \ c_j^{(m)}]^T$ is the vector of magnitudes of joint constraint impulses. In addition, we should consider that joints have mechanical limits, thus we can formulate another inequality constraint similar to the contact normal constraint as:

$$\hat{L}^T v^{t+1} \geq 0 \ \perp \ \hat{c_l} \ \geq 0, \tag{12}$$

where $\hat{L} = [L^{(1)} \ ... \ L^{(q)}]$, $\hat{c_l} = [c_l^{(1)} \ ... \ c_l^{(q)}]^T$ are the required impulses to keep the joints within their bounds and q is the number of reached joint limits.

C. Constructing the SLCP

Considering the constraints (7)-(12) mentioned above and adding the equation of motion (6) to the complementarity conditions, we obtain the following mixed LCP [3]:

$$\begin{bmatrix} M & -\hat{J} & -H \\ \hat{J}^T & 0 & 0 \\ H^T & 0 & N \end{bmatrix} \begin{bmatrix} v^{t+1} \\ \hat{c_j} \\ z \end{bmatrix} + \begin{bmatrix} -a \\ -\epsilon_j \\ b \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ w \end{bmatrix} \tag{13}$$

$$0 \ \leq z \ \perp \ w \ \geq 0,$$

where $H = [\hat{L} \ \hat{N} \ \hat{D} \ 0]$, $z = [\hat{c_l} \ \hat{c_n} \ \hat{\beta} \ \hat{\lambda}]^T$; $a = Mv^t + hk$; $b = [0 \ 0 \ 0 \ 0]^T$; $w = [\varepsilon \ \rho \ \sigma \ \gamma]^T$; N has the form

$$N = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & E \\ \hat{\mu} & -E^T & 0 \end{bmatrix}. \tag{14}$$

Solving for v^{t+1} and $\hat{c_j}$ in (13) we obtain a LCP of the form

$$(G + N)z + g = w,$$

$$\begin{array}{rcl} G & = & H^T M^{-1} H - \\ & & H^T M^{-1} \hat{J} (\hat{J}^T M^{-1} \hat{J})^{-1} \hat{J}^T M^{-1} H \\ g & = & b + H^T M^{-1} a - \\ & & H^T M^{-1} \hat{J} (\hat{J}^T M^{-1} \hat{J})^{-1} \hat{J}^T M^{-1} a. \end{array} \tag{15}$$

The LCP can be solved using Lemke pivot-based algorithm [13]-[17], [22]. However, using this formulation, it is not possible to directly apply the PGS-SM algorithm since the matrix $G+N$ is not positive definite. Only when $N = 0$, G is positive semi-definite and symmetric. To force the positive definiteness of G, and without losing the physical meaning of the formulation, we add to it a square diagonal matrix CFM (called constraint force mixing). The positive diagonal elements of CFM allow *soft constraints*.

To avoid $N \neq 0$, we replace the friction constraints (8) and (9) by the following complementarity conditions [5]:

$$\begin{array}{c} \hat{D}v - \tilde{v}_+ + \tilde{v}_- = 0 \\ 0 \leq \hat{\beta} - \hat{\beta_-} \perp \tilde{v}_+ \geq 0, \ 0 \leq \hat{\beta_+} - \hat{\beta} \perp \tilde{v}_- \geq 0 \end{array} \tag{16}$$

where $\tilde{v}_+$ and $\tilde{v}_-$ are the positive and negative components of the tangential velocity; $\hat{\beta_+}, \hat{\beta_-}$ are the lower and upper bounds of $\hat{\beta}$ respectively. In this case, the matrix $\hat{D}$ is composed by two perpendicular directions, D^{j1}, D^{j2}. Thus, $\hat{D}$ is on the tangent plane at contact j and perpendicular to f_n. With these parameters, we can define the size of a *friction box* by imposing a fixed bound on the tangential forces $-\mu\tilde{c_n} \leq \beta^{ji} \leq \mu\tilde{c_n}$ for $i = 1, 2$, $\tilde{c_n}$ is an estimate of c_n (e.g. the last LCP solution before the current integration step).

According to the above considerations, the system still has the form (13) but now $H = [\hat{L} \ \hat{N} \ \hat{D}]$, $z = [\hat{c_l} \ \hat{c_n} \ \hat{\beta}]^T$, $a = Mv^t + hk$, $b = [0 \ 0 \ 0]^T$, $w = [\varepsilon \ \rho \ \sigma]^T$ and N is the null matrix. Finally, we solve for v^{t+1} and $\hat{c_j}$ to get (15) with the complementarity conditions in (16) such that:

$$\begin{array}{c} 0 \leq \hat{c_l} \perp \ \varepsilon \geq 0, \ 0 \leq \hat{c_n} \perp \rho \geq 0, \\ 0 \leq \beta - \beta_- \perp \sigma \geq 0, \ 0 \leq \beta_+ - \beta \perp \sigma \geq 0. \end{array} \tag{17}$$

Finding the LCP solution vector z, we then obtain

$$\begin{array}{rcl} v^{t+1} & = & M^{-1}(\hat{J}\hat{c_j} + Hz + a) \\ \hat{c_j} & = & -(\hat{J}^T M^{-1} \hat{J})^{-1} \hat{J} M^{-1} (Hz + a) + \\ & & (\hat{J}^T M^{-1} \hat{J})^{-1} \epsilon_j. \end{array}$$

V. Simulation Experiments

In our experiments we compare three LCP methods to solve robotic object grasping with frictional contacts: the standard PGS method (III-A), the adapted PGS-SM method (III-B), and the standard Lemke method provided by *Graspit* [21].

The experiments have been performed on a AMD Turion 64 X2 dual-core 3.0 GHz. *Graspit* was compiled using GNU C++. It includes a collision detection library called PQP [23] and performs basic linear algebra operations with BLAS [24]. The PGS and PGS-SM algorithms have been

implemented in Fortran 77 package using double precision and sparse linear algebra. The subspace minimization is carried out with preconditioned GMRES for nonsymmetric linear systems [25]. Both algorithms were compiled using GNU f77.

The PGS and PGS-SM methods have been adapted for frictional contact problems by adding lower and upper bounds on the variables. We have integrated both methods in *Graspit* and calling them via a standard Fortran to C library.

A. Grasping Scenarios

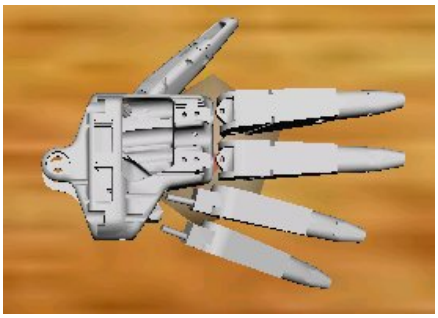
The implementations of PGS and PGS-SM were tested using some scenarios provided by *Graspit*.

In the 3D grasping simulator, we defined an initial configuration of a robotic hand that can be attached to a Puma arm or not, and with the fingers opened. We also defined the initial configuration of a movable object on a static table and close to the robotic hand (see Fig. 1.(a), Fig. 2.(a)). We can see that objects are already in contact with the table surface, generating friction cones at these contacts.

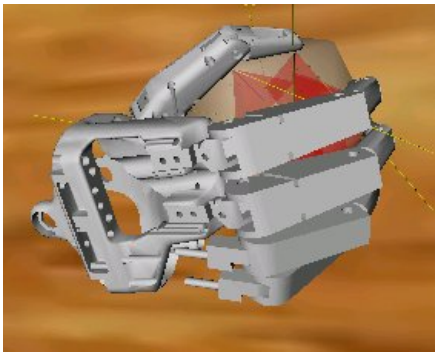
We then activated the multi-rigid-body dynamics process considering both the robot and the object. At this stage, the hand started to close their fingers. After some iterations of the numerical integration process the fingers collide with the object producing some contacts between any part of the hand (fingers' phalanges and/or the palm), the object and the table. Depending on the pre-shape of the hand, the fingers do not always touch the object at the same time, causing the object to slip unless the grasp can balance the forces to finally carrying the object to a stable pose, see Figs. 1.(b)-(d)). The simulation is stopped by the user.

In object manipulation and more precisely, in grasp analysis and planning, a fundamental problem is the computation of stable grasps. This allows the robot to perform a series of tasks (e.g. pick and place) in human environments. Most of the planners only solve the problem from a geometric and kinematic point of view giving good results. However, it is necessary to incorporate dynamics in the analysis of grasping for practical applications in robotics. One of the most important elements of this analysis is the frictional contact model between the object and hand.

A grasp could seem to be stable but when dynamic effects are considered, we notice that it is not. We show in Fig. 2 four instances of a grasping task. We have a plastic hand while the object is made of glass, thus, the coefficient of friction between these materials is $\mu = 0.2$. We observe that contact points appear with their respective frictional cones when the fingers touch the object (see Fig. 2.(b)). These contact points change in time due to the applied forces of the hand (see Fig. 2.(c)). In this example, the solutions of the multi-rigid-body dynamics in contact have been solved by the PGS-SM algorithm at every iteration. At the end of the simulation, we notice that even if the initial grasp seems to be good and the finger velocities are not so large, the hand cannot grasp the object (see Fig. 2(d)). This behavior frequently occurs for small coefficients of friction allowing slippage, which is a logic physical consequence.

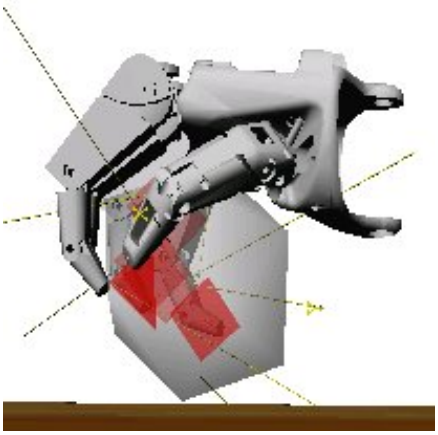


(a) Inital grasp



(b) Final grasp

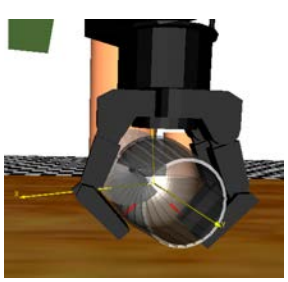
(c) After few iterations



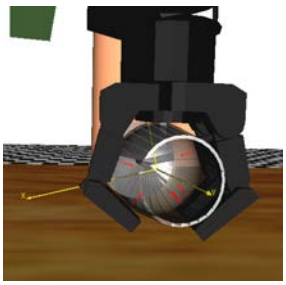
(d) Final grasp side view

Fig. 1. The figures show the results for one of the experiments. We used a five-fingered metal hand and a rubber ball object, after some iterations the robot picked the object up stably.

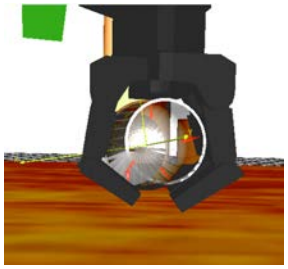
We perform the same experiment but changing only the hand material to rubber. In this case, the coefficient of friction is $\mu = 1.0$. We see in Fig. 3.(a) the frictional cones with $\mu = 0.2$ corresponding to wood table and glass cup while Fig. 3.(b) illustrates when the frictional cones between the hand (rubber) and the object. After some iterations the robot is capable to grasp the object stably.



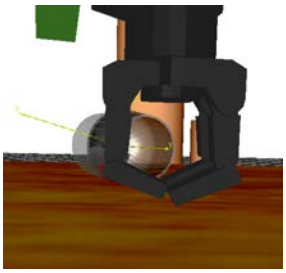
(a) First contacts with object



(b) Contacts with fingertips and distal phalanges

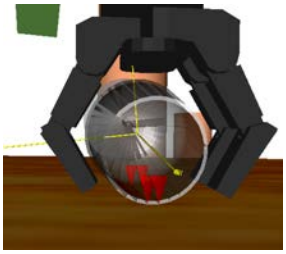


(c) Contact with palm

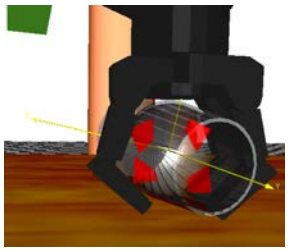


(d) Final result

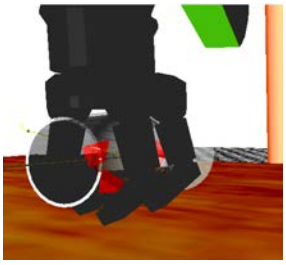
Fig. 2. The figure shows a three-fingered Barrett hand (plastic) and a glass object with $\mu = 0.2$. The fingers make contact with the object applying forces at the contact points. Due to the small friction between materials, the grasp cannot resist and the object slips.



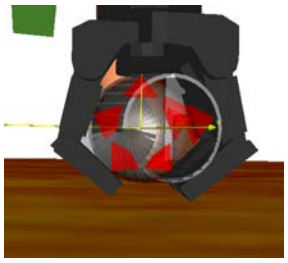
(a) Initial grasp



(b) Intermediate grasp (front view)



(c) Intermediate grasp (side view)



(d) Final grasp

Fig. 3. The figure shows the experiment with the same three-fingered hand but now it is made of rubber and the object is made of glass, with $\mu = 1.0$. We see that the final grasp is stable unlike the previous case.

B. Comparing Performances

To illustrate the behavior of the PGS-SM algorithm, we have conducted several comparisons between Lemke, PGS and PGS-SM in terms of computation time and number of contacts (size of the problem). We tested different examples of grasping tasks and we executed 3 times the same task invoking different LCP solver each trial.

At each time step throughout the simulation, we measured the CPU time required to find a solution of the frictional contact LCP.

In these experiments, before calling the PGS-SM method, we should specify the input parameters k_{PGS} and k_{SM}. After some tunning phase, we identified that $k_{\mathrm{PGS}} \in [3, 6]$ and $k_{\mathrm{SM}} \in [1, 3]$ are the adequate ranges. We set the stopping tolerance to $1.0e-8$ and the matrix $CFM = \mathrm{diag}[1.0e-11]$. The reason for these values are only for the numerical precision problem in computing operations when exact zero value is considered, this does not modify the algorithms or affect their performance.

We observed that the performance of the Lemke method was inefficient with respect to PGS and PGS-SM methods when the number of contacts increase. Note that the size of the problems depends on the number of contacts and joint limits reached. From 1 to 3 contacts, there are not significant difference between the 3 methods. From 4 to 12 contacts, the CPU time varied from 79 to 11820 microseconds. Table I shows the minimum, the maximum and the mean CPU time for Lemke, PGS and PGS-SM respectively. These quantities show the advantage of the adapted PGS-SM.

TABLE I

CPU TIME COMPLEXITY (IN MICROSECONDS).

	Min	Max	Mean
3-fingered hand			
- Lemke	146	7050	672
- PGS	97	1332	207
- PGS-SM	84	355	145
5-fingered hand			
- Lemke	220	11820	1885
- PGS	95	3348	238
- PGS-SM	79	452	155

We noted that the PGS method was able to identify the active set in few iterations as it is pointed out in [1]. The adapted PGS-SM method is quite efficient due to the right balance between identifying the active set (PGS iterations) and moving toward optimality (subspace minimization cycles). In our experiments the PGS-SM algorithm only did from 3 to 6 PGS iterations and used one subspace minimization cycle to get the solution. The PGS-SM is also significantly faster than Lemke in most cases.

The frictional contact problems in our experiments can be classified as small but ill-conditioned problems. Because the size of the problems was small, the performance of Lemke algorithm was acceptable. The PGS-SM method, on

the other hand, continue to be efficient for solving large-scale and ill-conditioned LCPs. It has been tested with large-scale frictionless contact problems extracted from the open source package *Open Dynamic Engine* (see [1] for details).

These preliminary results indicate that robotic frictional contact problems would be gained computation time by using PGS-SM method without regarding the size of ill-conditioned LCPs.

Simulation sequences of the examples presented in this work can be found at:

http://www.cinvestav.edu.mx/saltillo/robotica/garechav/grasping/

VI. Conclusions

In this paper we used an adaptation of the two-stage iterative algorithm proposed in [1] to deal with ill-conditioned dry frictional contact problems in grasping tasks. We formulate contact dynamics in a similar manner as [3].

The PGS-SM method allows us to solve constrained motions for robotic hands that have complex dynamics by applying appropriate iterative methods. Bounds in frictional variables are utilized to guarantee convergence to the grasping problems. We compute the frictional bounds by using the information of the previous iteration. As a result, the grasp motion has been successfully verified by simulations. We tested our approach with different friction coefficients and we compared the resulting behaviors with the solutions computed using Lemke algorithm. We are still working to find different ways to estimate the bounds. Nevertheless, our preliminarily results show that the PGS-SM method can be successfully applied in practice.

Other challenging problems arising in Humanoid robotics and motion planning for several and highly articulated mechanisms can be viewed as complex contact problems. For instance, the problem of computing whole-body stable object manipulation [26], [27]. Our current work is focusing on validating the use of the PGS-SM method in these classes of robotic problems.

VII. ACKNOWLEDGMENTS

The first two authors acknowledge the financial support of the National Council of Science and Technology (CONACyT) within the project No. 84855. J.L. Morales was supported by Asociación Mexicana de Cultura A.C.

References

[1] J. L. Morales, J. Nocedal, and M. Smelyanskiy, "An algorithm for the fast solution of symmetric linear complementarity problems," *Numerische Mathematik*, vol. 111, no. 2, pp. 251–266, 2008.

[2] A. Miller and H. Christensen, "Implementation of multi-rigid-body dynamics within a robotic grasping simulator," in *IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, September 2003, pp. 2262–2268.

[3] M. Anitescu and F. Potra, *Formulating Dynamic Multi-Rigid-Body Contact Problems with Friction as Solvable Linear Complementary Problems*, ser. Nonlinear Dynamics. Netherlands: Kluwer Academic Publishers, 1997, vol. 14, pp. 321–247.

[4] D. Stewart and J. Trinkle, "An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction," *International Journal for Numerical Methods in Engineering*, vol. 39, pp. 2673–2691, 1996.

[5] C. Lacoursiere, "Splitting methods for dry frictional contact problems in rigid multibody systems: preliminary performance results," in *SIGRAD*, 2003.

[6] M. Anitescu, F. Potra, and D. Stewart, "Time-stepping for three-dimensional rigid body dynamics," *Computer Methods in Applied Mechanics and Engineering*, vol. 177, pp. 183–197, 1999.

[7] A. Al-Fahed, G. Stravroulakis, and P. Panagiotopoulos, "Hard and soft fingered robot grippers, the linear complementary approach," *Z. angew. Math. u. Mech.*, vol. 71, no. 7-8, pp. 257–265, 1991.

[8] B. Mirtich and J. Canny, "Impulse based simulation of rigid bodies," in *Symposium on Interactive 3D Graphics*, New York, 1995, pp. 181–188.

[9] J. Trinkle *et al.*, "On dynamic multi-rigid-body contact problems with coulomb friction," *Z. angew. Math. u. Mech.*, vol. 77, no. 4, pp. 267–279, 1991.

[10] J. Pang and J. Trinkle, "Complementary formulations and existence of solutions of dynamics multi-rigid-body contact problems with coulomb friction," *Journal of Mathematical Computing*, vol. 73, no. 2, 1996.

[11] F. Pfeiffer and C. Glocker, *Multibody Dynamics with Unilateral Contacts*, ser. Wiley Series in Nonlinear Science. New York: John Wiley and Sons, 1996.

[12] D. Baraff, "Issues in computing contact forces for non-penetrating rigid bodies," *Algorithmica*, vol. 10, pp. 292–352, 1993.

[13] C. Lemke, "Bimatrix equilibrium points and mathematical programming," *Managment Science*, vol. 11, pp. 681–689, 1965.

[14] R. W. H. Sargent, "An efficient implementation of the lemke algorithm and its extension to deal with upper and lower bounds," *Mathematical Programming Study*, vol. 7, pp. 36–54, 1978.

[15] J. L. Morales and R. W. H. Sargent, "Computational experience with several methods for large-scale convex quadratic programming," *Aportaciones Matemáticas. Comunicaciones*, vol. 14, pp. 141–158, 1994.

[16] J. Lloyd, "Fast implementation of lemke's algorithm for rigid body contact simulation," in *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, April 2005, pp. 4538–4543.

[17] K. Yamane and Y. Nakamura, "A numerically robust lcp solver for simulating articulated rigid bodies in contact," in *Robotics:Science and Systems*, Zurich, June 2008.

[18] E. Catto, "Iterative dynamics with temporal coherence," in *Technical report, Crystal Dynamics*, Menlo Park, CA, 2005.

[19] M. J. F. Jourdan, P. Alart, "A gauss-seidel like algorithm to solve frictional contact problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 155, pp. 31–47, 1998.

[20] T. Liu and M. Wang, "Computation of three-dimensional rigid-body dynamics with multiple unilateral contacts using time-stepping and gauss-seidel methods," *IEEE Transactions on Automation Science and Engineering*, vol. 2, no. 1, pp. 19–31, January 2005.

[21] A. Miller and P. Allen, "Graspit!: A versatile simulator for grasp analysis," in *ASME Dynamic Systems and Control Division*, Orlando, FL, November 2000, pp. 1251–1258.

[22] R. Cottle and G. Dantzig, "Complementary pivot theory of mathematical programming," *Journal of Linear Algebra Applications*, vol. 1, pp. 103–125, 1968.

[23] U. G. Group, "Pqp - a proximity query package," in *http://www.cs.unc.edu/ geom/SSV/*.

[24] L. Blackford, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, M. Heroux, L. Kaufman, A. Lumsdaine, A. Petitet, R. Pozo, K. Remington, and R. C. Whaley, "An updated set of basic linear algebra subprograms (blas)," *ACM Transactions on Mathematical Software*, vol. 28, pp. 135–151, 2002.

[25] Y. Saad and M. Schultz, "Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 7, pp. 856–869, July 1986.

[26] K. Hsiao and T. Lozano-Perez, "Imitation learning of whole-body grasps," in *IEEE International Conference on Intelligent Robots and Systems*, Beijing, China, October 2006, pp. 5657–5662.

[27] E. Yoshida, M. Poirier, J.-P. Laumond, O. Kanoun, F. Lamiraux, R. Alami, and K. Yokoi, "Whole-body motion planning for pivoting based manipulation by humanoids," in *IEEE International Conference on Robotics and Automation*, Pasadena, CA, May 2008, pp. 3181–3186.