An Evaluation of Methods for Modeling Contact in Multibody Simulation

Evan Drumwright and Dylan A. Shell

***Abstract*— Modeling contact in multibody simulation is a difficult problem frequently characterized by numerically brittle algorithms, long running times, and inaccurate (with respect to theory) models. We present a comprehensive evaluation of four methods for contact modeling on seven benchmark scenarios in order to quantify the performance of these methods with respect to robustness and speed. We also assess the accuracy of these methods where possible. We conclude the paper with a prescriptive description in order to guide the user of multibody simulation.**

I. INTRODUCTION

There is a strong argument to be made for the suggestion that most multi-body simulation techniques are inadequate for robotics research needs. Clearly, an ideal simulator would be fast, accurate, and robust while being capable of modeling non-trivial interactions between bodies (e.g., friction and joint constraints). But as this paper will demonstrate, selecting the underlying algorithms and codes remains a task that is as challenging as it is important. Selecting an inaccurate or infeasibly slow algorithm will undermine the benefits offered by simulation in the first place. Although the best algorithm is domain dependant, we know of few evaluations across benchmarks (or even test scenarios) that might enable the roboticist to make an informed choice.

The impact of the contact model on performance is comprehensively evaluated in this paper across several simulation tasks of interest to the practitioner. The results show that making an informed choice is a balancing act; accuracy, reliability, and run-time vary across contact algorithms, underlying numerical solvers and input parameters. We believe these latter two dimensions in particular have been given inadequate attention; the data show that they can have a significant effect on simulation performance. In the comparisons made, no single contact method faired best under all circumstances. Nevertheless, the results do suggest that some methods are ill-suited for robotics and may be discarded in favor of others in nearly all circumstances.

This paper primarily serves to assess the speed and robustness of both the contact models and the methods for solving them. Assessing accuracy is a thorny issue, as explained by Chatterjee and Ruina [1]:

> Unfortunately no known collision law of any kind is both predictive and accurate, in the sense, say that Newton's laws, or even linear elasticity or the Navier-Stokes equations can be accurate. For a given pair of bodies, the collisional outcome may well depend on not just the initial velocities, masses, and mass moments of inertia, but also on some combination of contact shape, contact mechanics, surface chemistry, fracture, and vibration phenomena that are not well understood, especially *a priori*. Further, in the case of multiple superficially-simultaneous collisions, the prediction of the collisional outcome often depends so sensitively on initial conditions that sufficiently accurate initial conditions cannot be expected to be known by a simulator.

Thus, the best ways that accuracy can be assessed using a set of benchmarks are to compare the relative performance between the various contact models (*e.g.*, if ten contact models predict one result, and an eleventh predicts another, the eleventh is likely wrong) and to look for qualifying aspects of behavior that occurs in the real world (*i.e.*, does the robot drive or grasp as it should?) We utilize both assessments in this suite of benchmarks.

II. BACKGROUND

Among contact methods for computer-based multibody dynamics simulation are penalty methods [2], [3], [4], [5], force/acceleration LCP-based methods [6], [7], impulse-based event driven methods [8], [9], [10], time-stepping methods [11], [12], and computer graphics methods [13], [14], [15], [16]. A comprehensive survey of these models has been conducted by Brogliato *et al.* [17].

This paper studies only impulse-based event driven methods. Penalty methods are not considered because they permit interpenetration. Force/acceleration LCP-based methods are susceptible to inconsistent configurations and are thus not considered. Although we do not study time-stepping approaches, such methods are often fundamentally similar to event-driven schemes (*e.g.*, the time-stepping method of Stewart and Trinkle [12] is similar to the event-driven approach of Anitescu and Potra [10]) and often employ identical algorithms in their solution (*e.g.*, linear complementarity problem solvers); therefore, we expect our experimental results that indicate the efficacy of solvers on certain problems to yield insights into the performance of time-stepping methods on those same problems.

III. TESTED METHODS

Four contact models are tested: a Newton model augmented with friction; Mirtich's energetically consistent,

E. Drumwright is with the Department of Computer Science, The George Washington University, Washington, DC 20052, USA drum@gwu.edu

D. Shell is with the Department of Computer Science, Texas A&M University, College Station, TX 77840, USA dshell@cs.tamu.edu

Stronge-based collision model; the model of Anitescu and Potra; and the convex optimization-based model of Drumwright and Shell [18]. These contact models were chosen because they work with multi-rigid bodies, guarantee non-interpenetration, and operate as event-driven schemes. The models, as described in the literature, often employ optimizations (*e.g.*, the shock propagations in [9]). We attempted to tease these optimizations apart from the models in order to perform a fair evaluation across the cores of the various methods.

Additionally, we note that there exist numerous open source software libraries for computing rigid dynamics, including *ODE*, *Bullet*, *OpenTissue*, *Siconos*, and *daVinci*. Incorporating the contact models of those simulators into our comparison would be ideal but is impossible: the contact models cannot, in general, be separated from their libraries, and using the libraries themselves would not permit us to evaluate the contact models in isolation (it is not possible to use a common set of simulation parameters across all libraries).

A. Newton model

The Newton-based model implementation used in the experiments employs Newton's model for collision restitution augmented with a mechanism for simulating Coulomb friction. Our implementation is consistent with that described by Hahn [8].

B. Mirtich method

We implemented the method described in Mirtich's thesis [19] that adheres to Stronge's collision model [20].

C. Anitescu-Potra method

We implemented the method of Anitescu and Potra [10] using impulses to handle all contacts. Efficient solution of the linear complementarity problem generated by the Anitescu and Potra model is key to a fair comparison between competing methods. We use three LCP solvers. The first is a Lemke solver (adapted to C++) from the LEMKE library [21]. The second solver is the PATH solver [22], a commercial grade library for solving mixed complementarity problems. The third solver is the robust LCP solver of Yamane and Nakamura [23], augmented with Lloyd's speedups [24] that utilize problem structure. We implemented this solver, hereafter denoted Yamane-Nakamura-Lloyd (YNL), ourselves.

We do not employ any "splitting methods" to solve the linear complementarity problems, though these methods are popularly employed in simulators like ODE. As reported by Lacoursiére [25], the rates of convergence of such methods are unknown; indeed, it is not guaranteed that such methods will converge (leading to, in the case of the contact modeling problem, interpenetration). We ignore these methods, as widespread in practice as they may be, to avoid yet one more parameter (the number of iterations) for tuning.

D. Convex optimization method

We have implemented the convex-optimization-based method of Drumwright and Shell [18] that treats both resting contact and collisions in an impulse-based formulation with a Poisson-type collision model. This model necessitates the solution to a convex quadratically constrained, quadratic programming problem. This particular problem always has a solution.

In the experiments below, we have used our own implementation of a primal-dual interior point method described by [26]. The optimizer uses a single parameter, the ϵ-suboptimality constant, reported in the experiments.

IV. EXPERIMENTS

The benchmark scenarios used in the experiments have been implemented on the *Moby* [27] multi-rigid body simulation library. Every benchmark is encoded in human-readable XML format, and the set of benchmarks is downloadable from `http://www.seas.gwu.edu/~drum/bench`. By implementing the benchmarks in the *Moby* simulator, numerical issues (*i.e.*, integration and collision detection algorithms, floating point representation, *etc.*) can be held fixed and thereby ignored. *Moby* is released under the GNU Lesser Public License (LGPL), so researchers can improve upon and add to the implementations of contact models within *Moby*.

We selected these benchmarks for comparison because each was thought to emphasize a strength or weakness of a particular method and to gauge the effectiveness of contact models on standard robotic simulation scenarios. These comparisons are far from exhaustive, and we have even observed the same contact models applied to similar scenarios with vastly different performance. Nevertheless, these benchmarks are a starting point for a persistent means to measure the performance of the evolving space of contact models. We hope that the set of benchmarks will increase in size and scope as researchers discover phenomena that are poorly modeled by some or all of existing contact models.

The geometries used in the experiments are intentionally kept simple– boxes are used predominantly– in order to focus on contact methods, rather than issues with collision detection. Similarly, we set the coefficient of restitution to zero in order to evaluate the underlying contact model instead of restitution model (e.g., Newtonian, Poisson, etc.) employed: in the case of the convex optimization and Anitescu-Potra methods, it is possible to use alternative restitution models without difficulty.

Explicit Euler integration was used for every example. The acceleration due to gravity vector was set to $9.8m/s^2$. The continuous collision detection system of Shell and Drumwright [28], provided contact data (times of contact, contact points, and contact normals).

Table I summarizes the timing results for all experiments.

A. Impacting sphere

The impacting sphere scenario begins with a sphere of unit density, radius 1m, and particular velocity impacting a

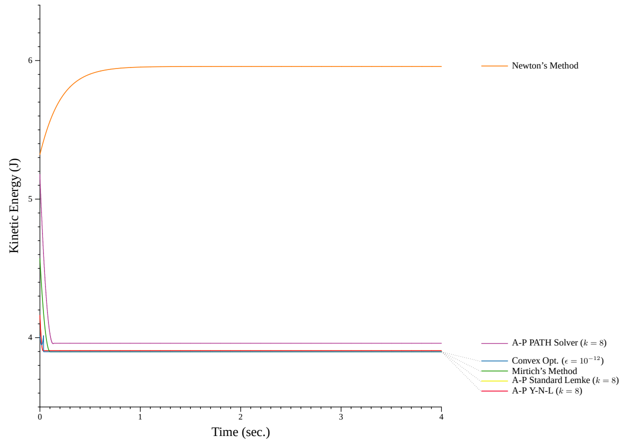

Fig. 1. Kinetic energy on impacting sphere scenario. Unit density sphere with radius 1m^3. Contact parameters $\epsilon = 0$ and $\mu = 0.60581$.

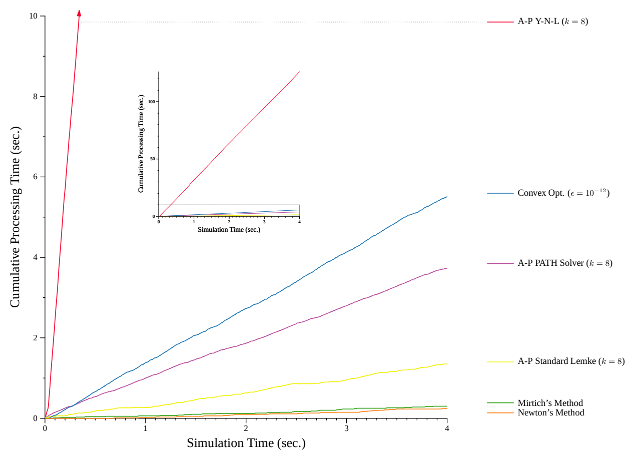

Fig. 2. Method timings for impacting sphere scenario.

ground plane. We chose to include this simple scenario in the evaluation because it incorrectly results in an increase in kinetic energy on impact with the Newton contact model; Figure 1 indicates that the remaining contact models do not exhibit energy gain. Figure 2 depicts the timings.

B. Sliding box

We included the scenario of a box being pushed along a planar surface as a relatively simple example that includes friction. This scenario consists of a 1m^3 box, initially at rest on the planar surface; from $t = 1\text{s}$ to $t = 5\text{s}$, a $1N$ force, parallel to the plane, is applied to the box. The coefficient of friction is 0.1.

As seen in Figure 3, most of the models yield identical performance. The exceptions are the Newton model, which permits the box to move farther, and the Anitescu-Potra method with Yamane-Nakamura-Lloyd LCP solver. Terminating the convex optimization solver early ($\epsilon = 10^{-2}$) also allows the box to move too far; this result is consistent with

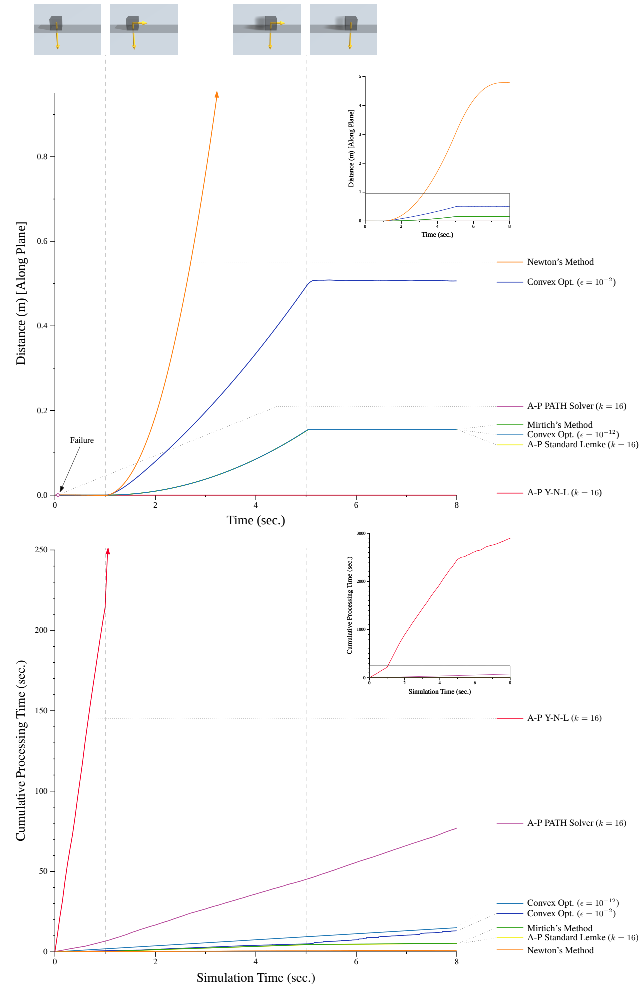

Fig. 3. Sliding box scenario. A unit mass, 1m^3 box starts at rest. From time $t = 1\text{s}$ to $t = 5\text{s}$ a $1N$ force is applied to the box. Contact parameters $\epsilon = 0$ and $\mu = 0.1$.

the convex optimization model.[1]

C. Sticking box

The scenario of a box resting on a ramp tests the extent to which a contact model admits small numerical errors: ideally, the box would not move. The Newton model failed to maintain the sticking condition for much more than a second of simulation time, as seen in Figure 4. The remaining methods exhibited little, though measurable, creep; it should be noted that there is approximately seven orders of magnitude difference between the best performing method (the Anitescu-Potra method using standard Lemke solver) and the poorest performing method (Anitescu-Potra with PATH solver). We also note that, as indicated in Figure 4, both Mirtich's method and Anitescu-Potra with PATH solver (sixteen friction cone edges), failed to complete the scenario.

[1]The initial feasible point is equivalent to frictionless contact, so increasing the number of iterations will generally– though not always– decrease the tangential velocity.

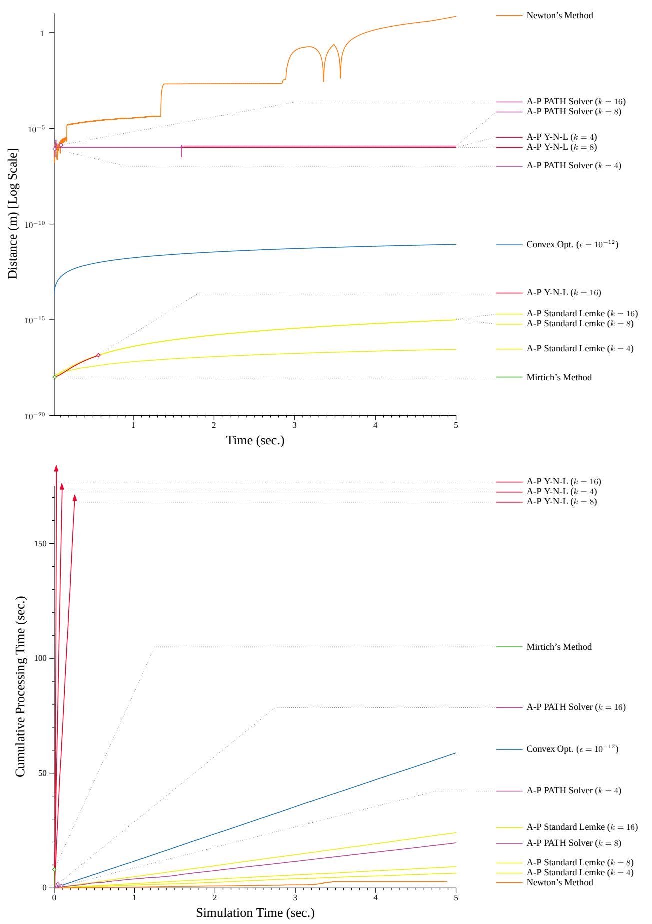

Fig. 4. Creep of the box in the sticking box scenario (top) and computation timings for the scenario (bottom).

D. Box sliding in circular motion

We use the scenario of a 1m^3, 1kg box sliding along a planar surface as if the box were at the end of a string (and hence under the influence of centrifugal forces). This scenario was modeled by artificially changing the gravity vector to $\begin{bmatrix} 0.4\cos(t) & -9.8 & 0.4\sin(t) \end{bmatrix}^\mathsf{T}$.

Figure 5 seems to indicate that the convex optimization method produces the solution that the Anitescu-Potra method converges to (given a sufficiently faithful approximation to the friction cone). Mirtich's method generates seemingly nice circular trajectories, but does not replicate the theoretically predicted result; Newton's model causes the box to slide in almost a single direction, and produces quite different results from the other methods. Figure 6 presents the timings for this experiment.

E. Stack of boxes

The stack of boxes scenario illustrates the performance of the various methods as the number of contact points increase. This scenario was run with one, two, four, eight, and sixteen blocks; four contact points were used at each contact surface, for a total of four, eight, sixteen, thirty-two, and sixty-four contact points.

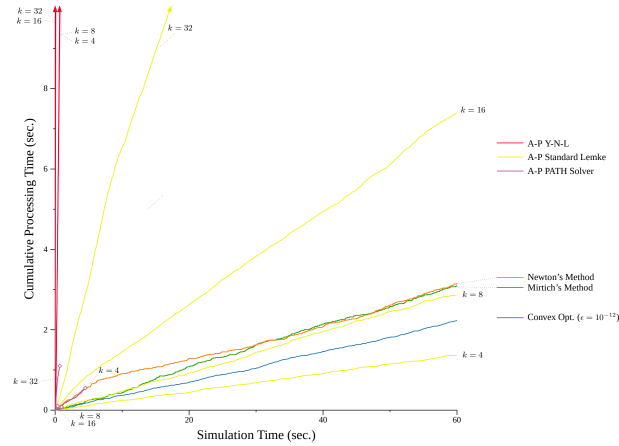

Fig. 6. Computation timings for the box sliding circularly.

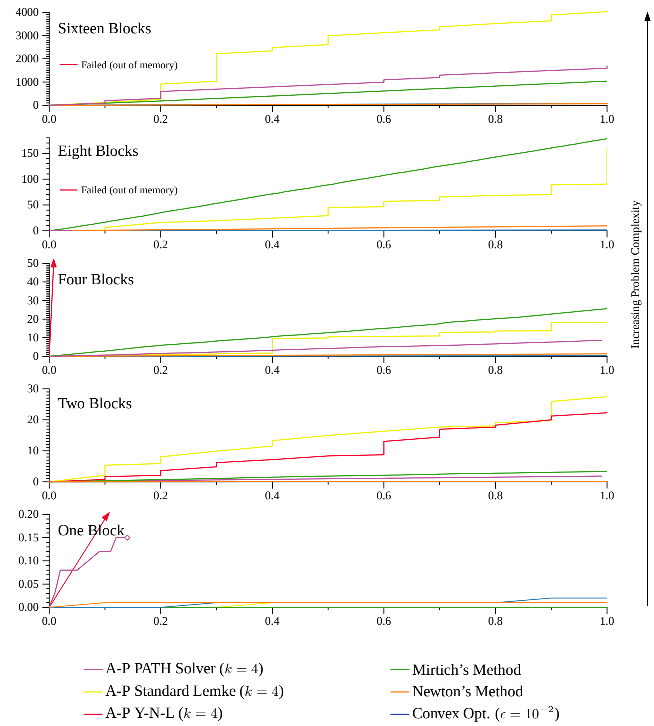

Fig. 7. Computation times for the stack scenario.

The convex optimization method and the Newton model were the fastest method on these examples; they are consistently plotted as the bottom lines in Figure 7. It is unsurprising that the Anitescu-Potra methods were among the slowest in this experiment; the problem sizes for the LCP solver were quite large. Mirtich's method- which requires integrating differential equations modeling the contact, for each contact point at a time- is also slow.

F. Mobile robot locomotion

We use a simulated mobile robot attempting to drive in a figure-eight path to test the contact methods' performance

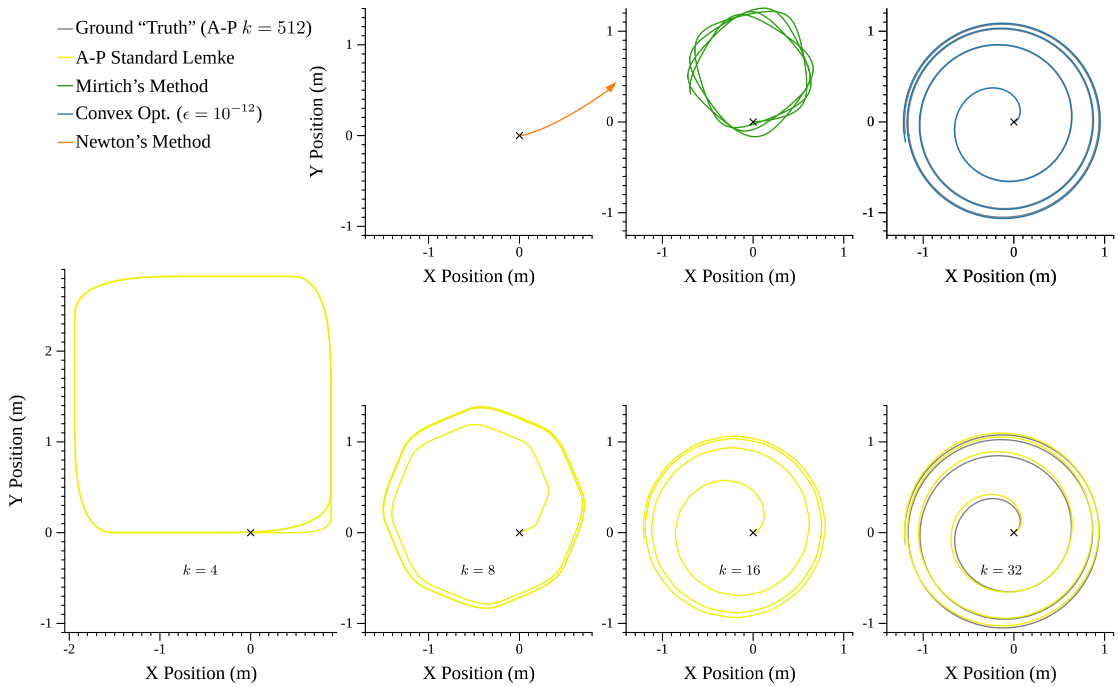

Fig. 5. Trajectories of the box sliding in a circular motion.

on a simple scenario in the robotics domain. The robot is simulated using Featherstone’s articulated body method [29] with a Coulomb friction coefficient between the wheels and the ground of 1.7. A virtual counterweight is placed below the robot’s center of mass in order to ignore issues of balance. Feedback control was used to drive the robot around the desired path.

As seen in Figure 8, only the convex optimization method permits the robot to drive in a figure-eight trajectory; the discrepancy from the true trajectory is due to driving the robot using only feedback control. An analysis of the failure of the Anitescu and Potra method indicates that a normal force is applied to only one wheel at a time (recall that the virtual counterweight keeps the robot balanced), meaning that one of the robot’s wheels is arbitrarily selected to receive no traction. We note that the virtual counterweight should not take the blame; replacing the counterweight with a caster does not distribute the contact forces equally among the wheels (as one expects) when one of the Lemke-based solvers is employed.

G. Manipulator grasping

Manipulator grasping has only recently been simulated (*cf.*, [30], [31]). This scenario tests both the robustness of methods for solving the contact model– kinematic loops are induced by the grasping– and the propensity of the contact model to permit creep (*i.e.*, the grasped object slipping).

This scenario consists of a six degree of freedom (DOF) manipulator arm and two DOF gripper setup with a small

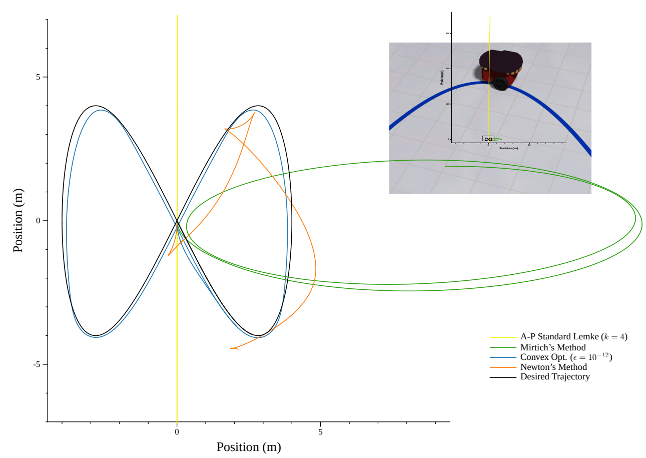

Fig. 8. Path taken by the mobile robot for the various methods in the mobile robot locomotion scenario.

box positioned snugly within the robot’s gripper; the goal of this example is to get the robot arm to follow a sinusoidal trajectory while grasping the box. The robot is simulated using Featherstone’s articulated body method [29]; the robot is controlled using a composite feedforward (Recursive Newton-Euler inverse dynamics [29]) plus proportional-derivative (PD) method. The robot’s two shoulder joints are driven to follow the path $\frac{\sin(t)}{4}$ and $\frac{\sin(2t)}{4}$ over 25 seconds of simulation time, allowing for four cycles of the movement. A desired acceleration of 100 m/s^2 for the gripper joints is used

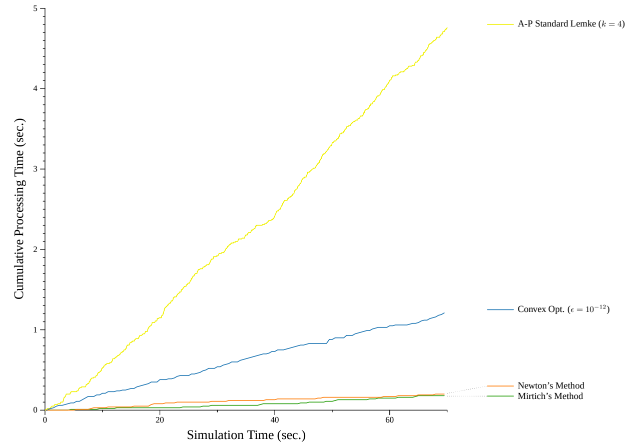

Fig. 9. Computation times for the mobile robot locomotion scenario.

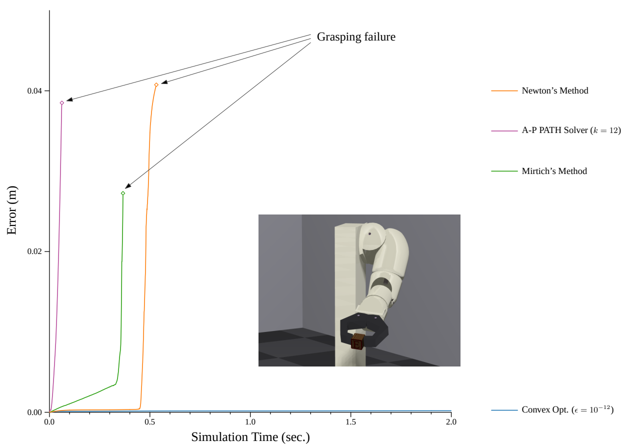

Fig. 10. Manipulator grasping scenario creep.

in combination with inverse dynamics to generate torques for grasping the box.

As seen in Figure 10, only the convex optimization method was able to model this problem properly[2]; the remaining methods would either drop the box (Newton and Mirtich) or fail to solve the linear complementarity problem (Anitescu-Potra). Using smaller simulation step sizes did not correct the problem. Figure 11 provides timings for all of the methods on this scenario.

V. CONCLUSION

The experiments presented in this paper elucidate the advantages and disadvantages of the various methods clearly. For scenarios with simple rigid bodies and few contact points and where simple (e.g., pyramidal) frictional approximations are adequate, the method of Anitescu-Potra method with standard Lemke LCP solver is superior with respect to speed; incorporating Lloyd's structural enhancements would speed

[2]We note that GraspIt! [32] uses the model of Anitescu and Potra in its simulator; Miller uses Tikhonov regularization in an attempt to better condition the LCP.

METHOD	IMPACTING SPHERE	SLIDING BOX	STICKING BOX			CIRCULAR SLIDING				STACK OF OF BOXES 1	2	4	8	16	MOBILE ROBOT	MANIP. GRASPING
Simulation Time	4.0	8.0	5.0			60.0				1.0	1.0	1.0	1.0	1.0	70.0	25.0
multicolumn1l**Mirtich's Method**	0.3	5.32	7.30			3.10				0.00	3.35	25.57	178.20	1036.87	0.18	DNF
Newtons's Method	0.24	0.97	2.84			3.15				0.01	0.13	1.32	9.55	66.02	0.20	DNF
Convex Optimization																
$\epsilon = 10^{-2}$		13.05								0.02	0.07	0.29	1.60	79.57	1.21	1108.29
$\epsilon = 10^{-12}$	5.51	15.01	58.83			2.23										
Anitescu & Potra																
Standard Lemke	1.35	5.18	6.47	9.28	24.08	1.36	2.86	7.40	32.86	0.01	27.24	18.69	159.72	4025.48	4.76	DNF
PATH	5.69	77.0	33.9	136.68	115.73	DNF	DNF	DNF	DNF	0.19	DNF	DNF	DNF	1694.45		
Yamane-Nakamura-Lloyd	126.19	2896.42	8867.59	3167.40	44361.36	0.06	0.09	5.19	DNF	1.53	22.28	DNF	DNF	DNF		
	$k = 8$	$k = 16$	$k = 4$	$k = 8$	$k = 16$	$k = 4$	$k = 8$	$k = 16$	$k = 32$	$k = 4$	$k = 4$	$k = 4$	$k = 4$	$k = 4$	$k = 4$	$k = 12$

DNF means "Did Not Finish"

TABLE I

SUMMARY OF THE CUMULATIVE PROCESSING TIME (IN SECONDS) REQUIRED TO SIMULATE EACH OF THE SCENARIOS DESCRIBED IN THE PAPER.

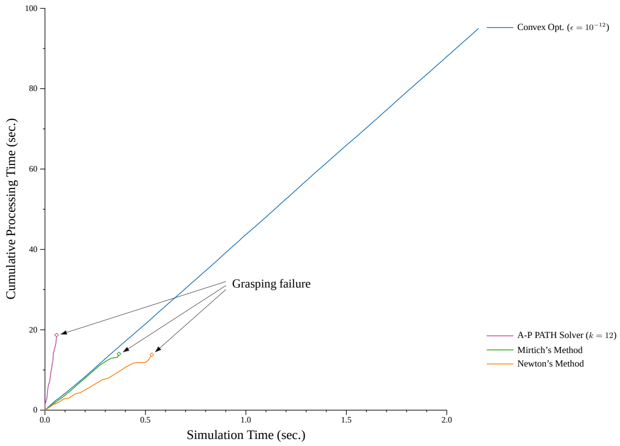

Fig. 11. Computation times for the manipulator grasping scenario. The standard Lemke and Yamane-Nakamura-Lloyd LCP solver for the Anitescu-Potra method failed to model a single iteration of this scenario.

this method further. Additionally, we found the standard Lemke LCP (*i.e.*, LEMKE [21]) solver to be far more robust than the PATH solver; this result defies common wisdom. For scenarios with kinematically looped contact constraints or greater numbers of contact points or where a more accurate conical frictional model is warranted, the convex optimization based method is superior; it is robust, runs quickly, and exhibits good worst case complexity.

Though we are confident that some parameter tweaking could alter our results, we cannot recommend the method of Yamane and Nakamura; its excessive memory consumption makes it unable to model scenarios of moderate complexity. Neither can we recommend the simple Newton contact model; it is robust, but its accuracy is questionable, even on simple scenarios. Mirtich's method is relatively fast, but does not seem to model scenarios with multiple contact points well; this method also exhibits robustness issues that are difficult to correct.

VI. ACKNOWLEDGMENTS

The authors gratefully acknowledge implementation advice from Katsu Yamane and John Lloyd.

REFERENCES

[1] A. Chatterjee and A. Ruina, "A new algebraic rigid body collision law based on impulse space considerations," *ASME J. Appl. Mech.*, vol. 65, no. 4, pp. 939–951, Dec 1998.

[2] M. Moore and J. Wilhelms, "Collision detection and response for computer animation," in *Proc. of Intl. Conf. on Computer Graphics and Interactive Techniques*, 1988, pp. 289–298.

[3] S. Hasegawa and M. Sato, "Real-time rigid body simulation for haptic interactions based on contact volume of polygonal objects," in *Proc. of Eurographics*, 2004.

[4] K. Yamane and Y. Nakamura, "Stable penalty-based model of frictional contacts," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Orlando, FL, USA, May 2006.

[5] E. Drumwright, "A fast and stable penalty method for rigid body simulation," *IEEE Trans. on Visualization and Computer Graphics*, vol. 14, no. 1, pp. 231–240, Jan/Feb 2008.

[6] P. Löstedt, "Numerical simulation of time-dependent contact friction problems in rigid body mechanics," *SIAM J. of Scientific Statistical Computing*, vol. 5, no. 2, pp. 370–393, 1984.

[7] D. Baraff, "Fast contact force computation for nonpenetrating rigid bodies," in *Proc. of SIGGRAPH*, Orlando, FL, July 1994.

[8] J. K. Hahn, "Realistic animation of rigid bodies," *Computer Graphics*, vol. 22, no. 4, 1988.

[9] E. Guendelman, R. Bridson, and R. Fedkiw, "Nonconvex rigid bodies with stacking," *ACM Trans. on Graphics*, vol. 22, no. 3, pp. 871–878, 2003.

[10] M. Anitescu and F. A. Potra, "Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems," *Nonlinear Dynamics*, vol. 14, pp. 231–247, 1997.

[11] 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.

[12] D. Stewart and J. Trinkle, "An implicit time-stepping scheme for rigid body dynamics with coulomb friction," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, San Francisco, CA, April 2000.

[13] H. Schmidl and V. J. Milenkovic, "A fast impulsive contact suite for rigid body simulation," *IEEE Trans. on Visualization and Computer Graphics*, vol. 10, no. 2, pp. 189–197, March/April 2004.

[14] D. M. Kaufman, T. Edmunds, and D. K. Pai, "Fast frictional dynamics for rigid bodies," *ACM Trans. on Graphics*, vol. 24, no. 3, August 2005.

[15] K. Erleben, "Velocity-based shock propagation for multibody dynamics animation," *ACM Trans. on Graphics*, vol. 26, no. 12, June 2007.

[16] D. M. Kaufman, S. Sueda, D. L. James, and D. K. Pai, "Staggered projections for frictional contact in multibody systems," *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia)*, vol. 27, no. (to appear), 2008.

[17] B. Brogliato, A. A. ten Dam, L. Paoli, F. Génot, and S. Abadie, "Numerical simulation of finite dimensional multibody nonsmooth mechanical systems," *ASME Appl. Mech. Reviews*, vol. 55, no. 2, pp. 107–150, March 2002.

[18] E. Drumwright and D. A. Shell, "Modeling contact friction and joint friction in dynamic robotic simulation using the principle of maximum dissipation," in *Proc. of Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2010.

[19] B. Mirtich, "Impulse-based dynamic simulation of rigid body systems," Ph.D. dissertation, University of California, Berkeley, 1996.

[20] W. J. Stronge, "Rigid body collisions with friction," *Proc. of the Royal Society of London A*, vol. 431, no. 169–181, 1990.

[21] P. L. Fackler and M. J. Miranda, "LEMKE," http://people.sc.fsu.edu/ burkardt/m_src/lemke/lemke.m.

[22] M. C. Ferris and T. S. Munson, "Complementarity problems in GAMS and the PATH solver," *J. of Economic Dynamics and Control*, vol. 24, no. 2, pp. 165–188, Feb 2000.

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

[24] J. E. Lloyd, "Fast implementation of lemke's algorithm for rigid body contact simulation," in *Proc. of the IEEE Conf. on Robotics and Automation (ICRA)*, 2005, pp. 4538–4543.

[25] C. Lacoursière, "Splitting methods for dry frictional contact problems in rigid multibody systems: Preliminary performance results," in *Proc. of SIGRAD*, M. Ollila, Ed., Nov 2003, pp. 11–16.

[26] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[27] E. Drumwright, "Moby," http:://physsim.sourceforge.net.

[28] D. Shell and E. Drumwright, "Precise generalized contact point and normal determination for rigid body simulation," in *Proc. of the ACM Symp. on Applied Computing*, Honolulu, HI, March 2009, pp. 2107–2108.

[29] R. Featherstone, *Robot Dynamics Algorithms*. Kluwer, 1987.

[30] A. T. Miller and H. I. Christensen, "Implementation of multi-rigid-body dynamics within a robotic grasping simulator," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Sept 2003, pp. 2262–2268.

[31] E. Drumwright and D. A. Shell, "A robust and tractable contact model for dynamic robotic simulation," in *Proc. of ACM Symp. on Applied Computing (SAC)*, 2009, pp. 1176–1180.

[32] A. T. Miller, "Graspit: A versatile simulator for robotic grasping," Ph.D. dissertation, Columbia University, 2001.