A Parallel Block Iterative Method for Interactive Contacting Rigid Multibody Simulations on Multicore PCs

Claude Lacoursière[1 2]

[1] HPC2N/VRlab and Department of Computing Science
Umeå University, SE-901 87 Umeå, Sweden,
claude@cs.umu.se, http://www.cs.umu.se/~claude
[2] CMLabs Simulations Inc.
420 Notre-Dame Ouest, #505, Montréal, Qc, H2Y 1V3, Canada
claude@cm-labs.com, http://www.cm-labs.com

Abstract. Fast solution methods for complementarity problems (CPs) with several hundred variables are essential for interactive realtime simulation of systems of constrained, contacting rigid bodies. These simulations are commercially relevant for several applications such as virtual environment (VE) heavy machinery operator training systems. We describe a hybrid, asynchronous, block parallel method to approximately solve these CPs in realtime on multicore CPUs. A graph analysis phase identifies components which are weakly coupled using a physically inspired weight function. Each component is then solved in parallel using either a block principal pivot method or a projected block Gauss Seidel method running in separate threads. Couplings which generate forces between the subsystems are handled iteratively using a Gauss-Seidel process which communicates updates between the interacting subsystems asynchronously. Preliminary results show that this approach delivers good performance while keeping overhead small.

1 Introduction

Interactive realtime simulation of systems of constrained rigid bodies and particles is an essential component of several commercially relevant applications, such as virtual environment heavy machinery operator training systems. Numerical integration of multibody systems needed for these simulations involves solution of systems of equations with several hundred variables. When contact constraints and friction are considered, one must solve mixed linear complementarity problems (MLCP) as described below. In the realtime interactive simulation context, solutions of these MLCPs must be delivered in less than 10ms to keep screen refresh rates above 60 frames per second. Direct solution methods for MLCP are difficult to implement and are slow for large systems because they rely on pivoting methods [2] and sparsity exploitation is rather cumbersome. For this reason, direct solution methods have been neglected especially in the games and

3D graphics literature where iterative methods–which are fast but inaccurate–have been preferred [4] [3]. On the other hand, parallelization in this field has been generally neglected, except for task parallelism.

The present note proposes a parallel hybrid method which uses either a direct or iterative solver on large blocks and Gauss-Seidel iterations to compute the coupling forces between each block. This allows to easily exploit the extra computing power recently made available on commodity single CPU multicore PCs. This scheme allows to reach good performances for relevant applications such as log forwarder simulation which includes a complicated vehicle as well as a pile of logs carried in a trailer.

2 Problem Formulation

Consider a general mechanical system with n-dimensional generalized velocity vector $v \in \mathbb{R}^n$. The system has a square, real, positive definite and block diagonal $n \times n$ mass matrix M, with easily computed inverse $U = M^{-1}$. The system is subject to a variety of constraints which have Jacobian matrix G of size $m \times n$. The essential computation of a large family of stepping schemes involves the solution of the *mixed complementarity problem* (MLCP):

$$\begin{gathered} Sy + q = w^{(+)} - w^{(-)} \\ 0 \le y - l \perp w^{(+)} \ge 0, \qquad 0 \le u - \lambda \perp w^{(-)} \ge 0, \end{gathered} \tag{1}$$

at each time step. The real, square $m \times m$ matrix S has the form $S = GUG^T$. Vector $q \in \mathbb{R}^m$ is a real vector, $w^{(+)}, w^{(-)}$ are the real positive and negative components of the residual vector $w \in \mathbb{R}^m$, respectively. The lower and upper bound vectors $l, u \in \mathbb{R}^m + \{\pm\infty\}$ are extended real vectors. The inequality and orthogonality signs in (1) are understood componentwise so that for $a, b \in \mathbb{R}^m$, $a \ge b \Rightarrow a_i \ge b_i, i \in \{1, 2, \ldots, m\}$, $a \le b \Rightarrow a_i \le b_i, i \in \{1, 2, \ldots, m\}$, and $a \perp b \Rightarrow a_i b_i = 0$ for all $i \in \{1, 2, \ldots, m\}$, whenever $a, b \ge 0$. This problem of solving (1) is abbreviated as MLCP(S, q, l, u). The solution of this MLCP is the vector $y \in \mathbb{R}^m$, which is unique as long as S is symmetric and positive definite. Moreover, it produces the constraint force vector $G^T y$, from which the updated velocities and coordinates can be computed directly.

For most mechanical systems, the constraint Jacobians have a simple block structure:

$$G^T = \left[{G^{(1)}}^T \; {G^{(2)}}^T \; \ldots \; {G^{(m_c)}}^T \; \right], \tag{2}$$

where each row block $G^{(i)}$ has sizes $n_i \times n$, with $\sum_i n_i = m$. Each row block $G^{(i)}$ contains only a few non-zero column blocks, $j_1, j_2, \ldots, j_{m_i}$ where m_i is usually 1 or 2. This is the case for most of the common mechanical constraints such as hinge, prismatic joints, and contact constraints. In this format, each column block usually corresponds to a single physical body. This produces the first level of partitioning of the system where the velocity vector v is decomposed into blocks $v^{(i)}, i = 1, 2, \ldots n_b$, which may or may not have the same dimensionality. Each block here corresponds to a given physical body.

The connectivity graph at this level of partitioning consists of a bipartite graph where nodes are either physical bodies or constraints. It is easy to extract the connected components from this graph to produce decoupled problems and avoid large matrices, allowing for trivial parallelism. To parallelize further, we consider one of these connected components and assume a partitioning of the bodies into two groups for example, labeled as $1, 2$. Any given constraint Jacobian can then be split into blocks: $G^{(i)} = [G_1^{(i)}, G_2^{(i)}]$, where block $G_j^{(i)}$ acts only on the block coordinates $v_j, j = 1, 2$. The natural separation of constraints produces three groups, namely, those acting only on the first group, those acting only on the second group, and those acting on both groups of bodies:

$$G = \begin{bmatrix} G_{11} & 0 \\ 0 & G_{22} \\ G_{31} & G_{32} \end{bmatrix} . \tag{3}$$

After splitting the mass matrix correspondingly, matrix S in (1) has the following form:

$$S = \begin{bmatrix} S_{11} & 0 & S_{31}^T \\ 0 & S_{22} & S_{32}^T \\ S_{31} & S_{32} & S_{33} \end{bmatrix} , \tag{4}$$

where $S_{3i} = G_{3i} U G_{ii}^T, i = 1, 2$, and $S_{33} = \sum_{i=1}^{2} G_{3i} M_{ii}^{-1} G_{3i}^T$. Obviously, if the constraints in group 3 are chosen to be mass orthogonal to those in groups 1 and 2, there are no off-diagonal terms. Minimizing these terms is key to fast convergence of the iterations.

To formulate an iterative method for solving this problem, we need a definition for the residual error. Given a candidate solution vector $x \in \mathbb{R}^m$, we first define the infeasibility vector $f \in \mathbb{R}^m$ as: $r = l - u - \min(l, x) + \max(u, x)$, where the min and max operators are taken componentwise. It is clear that $r_i = 0$ whenever $l_i \leq x_i \leq u_i, \forall i$. The total error vector is then defined as $e_i = r_i + \theta(r_i) w_i$ where $\theta(z)$ is the indicator function with $\theta(z) = 0$ whenever $z \leq 0$ and $\theta(z) = 1$ otherwise. The root mean square residual error is then defined as: $\xi = (1/m)\|e\|$.

A projected block Gauss-Seidel solution of MLCP(S, c, l, u) is formulated in Algorithm 1.

This Gauss-Seidel process reduces the infeasibility monotonically for a positive definite matrix, even in the case of MLCP [2] but the convergence is linear at best and can be stationary in case the matrix S is degenerate or nearly so. This algorithm can be parallelized with or without syncrhonization (see [1] for descriptions of chaotic asynchronous schemes) by using threads so that solutions to each of the subsystems are computed in parallel. Overhead can be kept small by only having mutex and condition variables to control block reading from and block writing to the shared data, namely, the kinematic variables in subsystems $1, 2$.

Algorithm 1 Serial block projected Gauss-Seidel for solving partitioned MLCP(S, c, l, u).

```
Set $\nu \leftarrow 1$, choose $tol > 0$ and $\nu_m \in \mathbb{N}, \nu_m > 0$
Initialize $\lambda_1^{(1)}, \lambda_2^{(1)}, \lambda_3^{(1)}$.
**repeat**
    Solve: $\lambda_1^{(\nu+1)} \leftarrow \mathrm{SOL}(\mathrm{MLCP}(S_{11}, c_1 + S_{31}^T \lambda_3^{(\nu)}, l_1, u_1))$
    Solve: $\lambda_2^{(\nu+1)} \leftarrow \mathrm{SOL}(\mathrm{MLCP}(S_{22}, c_2 + S_{32}^T \lambda_3^{(\nu)}, l_2, u_2)))$
    Solve: $\lambda_3^{(\nu+1)} \leftarrow \mathrm{SOL}(\mathrm{MLCP}(S_{33}, c_3 + S_{31} \lambda_1^{(\nu+1)}, +S_{32} \lambda_2^{(\nu+1)}, l_3, u_3))$
    $\nu \leftarrow \nu + 1$
**until** $\|e_\nu\| < tol$ OR $\nu > \nu_m$
```

3 Results and Conclusion

An implementation of Algorithm (1) was realized using Vortex, a commercial rigid multibody dynamics simulation library not originally designed to handle couplings between paritition (cf. `http://www.cm-labs.com`). Numerical results demonstrate that significant speedup from parallelism can be achieved using simple partitioning schemes. The current commercially relevant computers for this type of application being single CPU dual core systems, the present strategy meets the goal of extracting good performance at a reasonable hardware cost.

Future work will investigate more sophisticated partitioning schemes based on spectral analysis as well as conjugate gradient-type methods for computing inter-partition couplings.

4 Acknowledgments

This research was conducted using the resources of High Performance Computing Center North (HPC2N), and supported in part by the "Objective 1 Norra Norrlands" EU grant awarded to HPC2N/VRlab at Umeå University, and by the *Swedish Foundation for Strategic Research* under the frame program grant SSF-A3 02:128.

References

1. Zhonghi Bai and Yuguang Huang. A class of asynchronous parallel multisplitting relaxation methods for the large sparse linear complementarity problems. *Journal of Computational Mathematics*, 21(6):773–790, 2003.
2. Joaquim J. Júdice. Algorithms for linear complementarity problems. In E. Spedicato, editor, *Algorithms for Continuous Optimization*, volume 434 of *NATO ASI Series C, Mathematical and Physical Sciences, Advanced Study Institute*, pages 435–475. NATO, Kluwer Academic Publishers, 1994.
3. Danny M. Kaufman, Timothy Edmunds, and Dinesh K. Pai. Fast frictional dynamics for rigid bodies. *ACM Trans. Graph.*, 24(3):946–956, 2005.
4. R. Weinstein, J. Teran, and R. Fedkiw. Dynamic simulation of articulated rigid bodies with contact and collision. *IEEE TVCG*, 12(3):365–374, 2006.