Accelerating (Not!) the Iterative Convergence Method

Richard W. DeVaul
Masters Student, Aesthetics and Computation Group
The Media Laboratory
Massachusetts Institute of Technology

December 17, 1997

1 Introduction

Central to my prior masters research is an iterative technique for solving Lagrange multiplier constraint problems, the Iterative Convergence method. This method allows constraint forces to be calculated in linear time for constraint systems which would otherwise be $O(n^3)$.

2 Algorithm Overview

Conventional Lagrange multiplier techniques require solving the linear system

$$A\lambda = \mathbf{b}, \tag{1}$$

where A is a square positive definite matrix, and λ and $\mathbf{b}$ are vectors of dimension n, where n is the number of constraints. Since A is not guaranteed to be invertible or sparse, solving for λ requires dense matrix methods (such as calculating the pseudo-inverse of A) which are generally $O(n^3)$

Baraff[2] has developed a method for solving *acyclic* systems of constraints in linear time. The Iterative Convergence method[1] exploits this property to allow cyclic problems to be iteratively solved in linear time. The algorithm has four phases; The first two phases are analytic, and involve the subdivision of the cyclic constraint system into acyclic sub-problems, and

[1]Please see my thesis for an in-depth discussion of this process, as well as a more general discussion of the problem.

the grouping together of those sub-problems which do not share constrained elements. The result is groups of acyclic constraints which have the property that each group member interacts only with members of other groups. As a result, every group can be solved in $O(n)$ time as though it were an independent problem, which is done in phase three. In the fourth phase, the initial guess produced in phase three is iteratively refined until a desired accuracy is reached.

3 Algorithm Order

In any iterative technique, the overall order of the algorithm is determined by the order of the loop body (how much effort goes into each iteration) and the order of the iteration process itself. In the case of the Iterative Convergence method, it is clear that each pass through the body of the loop is an $O(n)$ process. The only question remains as to the order of the iteration process.

As I stated in my problem proposal, the primary goal of this project was to determine the order of the Iterative Convergence method. The Iterative Convergence method operates on an m by 1 lattice, where m is the number of constraint groups. Each group interacts with every other group. Written in terms of a matrix problem (there are no source terms)

$$A \cdot \mathbf{u} = 0, \tag{2}$$

where the $m \times m$ matrix A (which itself is composed of matrices) is dense. Relaxation techniques are generally $O(n^2)$ when not accelerated[1], with the actual convergence rate being determined by the eigenvalues of the iteration matrix. This suggests that the order of the iteration process is $O(m^2)$, though the complexity of the process of linear constraint-solving which takes place at each step makes this far from certain.

4 Improving the Algorithm

The other stated goal of the research proposal is to produce an improved algorithm. Since the Iterative Convergence method appears to be a relaxation algorithm, I thought it might be possible to improve the performance by using an over-relaxation formulation. In this form, for each constraint group i,

$$\mathbf{F}_i^0 = \mathcal{C}(\mathbf{x}, \dot{\mathbf{x}}, J_i, \dot{J}_i, M, \mathbf{F}_{applied}), \tag{3}$$

and the first approximation solution to the overall problem is

$$\mathcal{F}^{\mathbf{0}} = \sum_{i=1}^{m} \mathbf{F}_i^0, \tag{4}$$

where $\mathbf{x}$ is the position vector, $\dot{\mathbf{x}}$ is velocity, J_i is the Jacobian of group i, $\dot{J}_i$ is the first time derivative of J_i, M is the mass matrix, and $\mathbf{F}_{applied}$ is the vector of applied forces. For successive iterations $n > 1$,

$$\mathbf{F}_i^n = (1-\alpha)\mathbf{F}_i^{n-1} + \alpha\mathcal{C}\left(\mathbf{x}, 0, J_i, 0, M, \sum_{j \neq i} \mathbf{F}_j^{n-1}\right), \tag{5}$$

and

$$\mathcal{F}^{\mathbf{n}} = \mathcal{F}^{\mathbf{n-1}} + \sum_{i=1}^{m} \mathbf{F}_i^n, \tag{6}$$

where α is the relaxation coefficient.This formulation (which I will refer to as the Relaxation Constraint method) is a straightforward extension of the Iterative Convergence method.

For $\alpha = 1$, the Relaxation Constraint formulation is equivalent to the Iterative Convergence method. Values $0 \leq \alpha < 1$ result in under-relaxation, and $1 < \alpha \leq 2$ produce over-relaxation. An empirical search suggests that over-relaxation results in better numerical behavior (and hence larger time steps) but at the expense of solver accuracy. Under-relaxation produces slower convergence and less accurate solutions. Thus, the the optimal value for solver accuracy and speed of convergence is $\alpha = 1$!

5 Results and Conclusions

Although the convergence behavior of this algorithm is difficult to analyze, it seems to be well behaved and robust for constraint meshes. Increasing the total number of constraints appears to cause a low-order increase in iteration (order linear or less), but in the test cases I have run this could just be a sampling issue related to changes in geometry and the stiffness of springs.

In a simulation run of 0.57 seconds, a simulated 1 meter square sheet of cotton draping over a cube sampled using a 9x9 mesh resulted in a total of 41,532 calls to the solver, averaging 6.45 iterations per call. An identical simulation run with a 17x17 mesh (roughly quadruple the total number of constraints) resulted in 52,724 calls, averaging 8.87 iterations. Tests with larger meshes suggest that quadrupling the size of the problem results in a proportional (roughly %30) increase in the number of iterations.

Empirical evidence suggests that the iteration order dependence on m is not simple — dividing one logical grouping of constraints into two (and hence increasing the number of groups) does not appear to cause an increase in the number of iterations, and may even improve performance. I suspect that the iteration order is determined more by the geometry and topology of the decomposed constraint system than the actual number of groups, but I have not yet been able to test this hypothesis in a meaningful way.

6 Future Research

More can be done to investigate the properties of the Iterative Convergence method, particularly in terms of its applicability to a wider range of problems, and performance scaling for significantly larger problems.

References

[1] William F. Ames. *Numerical Methods for Partial Differential Equations.* Barnes & Nobel, New York, 1st edition, 1969.

[2] David Baraff. Linear-time dynamics using lagrange multipliers. In *Computer Graphics Proceedings, Annual Conference Series, 1996*, pages 137–146, New York, August 1996. ACM SIGGRAPH.