Technical Report 2007-04

An Investigation on New Numerical Methods for Molecular Dynamics Simulation

Nick Schafer, Radu Serban and Dan Negrut

August 13, 2007

Abstract

Explicit integrators have found common use in Molecular Dynamics (MD) simulations because they are easy to implement and work well under many conditions. However, in other classical mechanics applications that require the numerical solution of the equations of motion for complex systems, explicit methods have encountered major difficulties. In these cases, the state of the art relies on implicit methods, which are stable under large time steps and therefore can be used to decrease the number of integration steps necessary for a simulation. This in turn results in an overall reduction of CPU time that opens the door to an increase in the dimension of the problem that can be considered. The premise of this work is that numerical methods that are suitable for efficient simulation of mechanical systems will lead to significant gains when used in MD. The goal of the proposed work is to investigate this assumption. This paper (a) proposes a set of two benchmark problems used to validate new numerical solution techniques, and (b) presents simulation results obtained with a new class of implicit integrators (generalized alpha) and compares its performance against current explicit MD integrators when used in conjunction with the proposed benchmark problems.

Keywords: Molecular Dynamics, Implicit Integrators

Contents

1 Introduction

Molecular Dynamics is an extremely powerful tool that is based on a rather simple idea. Newtons second law says that the force on an object is equal to its mass times its acceleration. If the force on the particle is known as a function of space and time then the trajectory of the particle can be computed by integrating Newtons equations of motion. Classical Molecular Dynamics assumes that molecules can be treated as if they were under the influence of classical forces. This assumption works remarkably well for many situations and computer simulations based on this idea have already found many applications. In addition to yielding microscale dynamical information, the methods of statistical mechanics can be applied to derive macroscale properties of interest. Like most means of investigation, however, there are some difficulties and much room for improvement. One of the major difficulties in Molecular Dynamics is the transferability of models used to calculate the interactions between bodies. The assumption that a molecule is subject to classical forces is an approximation and as such is only valid in a certain domain (in certain circumstances, or conditions). Creating models that are both accurate and transferable is an active area of research. Part of the attractiveness of Molecular Dynamics is its ability to handle systems, in theory, with an arbitrary number of degrees of freedom. This new ability comes at a cost: the calculation of forces and integration of equations of motion must be done numerically. Whenever you pass from the domain of analytical solutions to numerical solutions, numerical methods become an issue. When calculating many individual particle trajectories in very large systems, efficient and stable numerical algorithms are a must. Several different algorithms have been used to integrate the equations of motion in Molecular Dynamics. Explicit integrators have found common use because they are easy to implement and work well under many conditions. However, in other classical mechanics applications that require the numerical solution of the equations of motion for complex systems, explicit methods have encountered major difficulties. In these cases, the state of the art relies on implicit methods to produce accurate solutions quickly. In many ways the equations of classical mechanical system simulation and Molecular Dynamics are very similar since they both draw on Newtons second law. Consequently, it seems natural to try to apply recent numerical methods developed in the realm of classical mechanical system simulation to problems in Molecular Dynamics. The premise of this work is that methods that are suitable for efficient simulation of mechanical systems will lead to significant gains when used in Molecular Dynamics. The goal of the proposed work is to investigate this assumption. To this end, we (a) propose a set of two benchmark problems used to validate new numerical solution techniques and (b) present simulation results obtained with a new class of implicit integrators (generalized alpha) and compare its performance against current explicit MD integrators when used in conjunction with the proposed benchmark problems. Accuracy, efficiency and stability are all important measures of the worthiness of a numerical integrator, and all these characteristics will be considered in assessing the potential of the proposed methods. Implicit methods are stable under large time steps and

therefore can be used to decrease the number of integration steps necessary for a simulation, resulting in an overall reduction of CPU time. Reducing the amount of time required to run a simulation effectively increases the dimension of the problem which can be considered. Although Molecular Dynamics has already been applied to many problems on short time and length scales, many interesting phenomena, such as biochemical processes, currently reside on the border or outside of what is feasible to simulate. In order to explore these areas, new more efficient numerical methods must be developed and tested.

2 Background

In protein structure investigations, homology models are refined at high resolution using an all-atom force field and Molecular Dynamics (MD) as a search engine. On numerical stability grounds [4], the standard numerical methods for explicit integration of the equations of motion [12, 13, 15] are limited to femtosecond-order time steps due to the short period for some vibration modes, such as C-H bond stretching. An alternative approach that offers much improved numerical stability and thus allows for much greater time steps is implicit integration, which is extensively used in mechanical engineering for the time domain analysis of finite element models [5]. Due to good stability properties and very attractive high frequency damping properties, the most widely used implicit integrators belong to the Hilber-Hughes-Taylor (HHT) family of methods. These are A-stable [6] and have an adjustable damping factor that filters out spurious high frequency vibration modes (which in the Finite Element method are artifacts of the intermediate finite element discretization step [6]), while preserving low frequency, large amplitude motions [5]. Furthermore, this ability to adjust the level of numerical damping in the HHT family of integrators addresses the numerical dissipation concerns typically associated in MD with the use of other implicit numerical methods [12]. The numerical damping of an HHT integration formula can actually be controlled to produce anything from a method with no numerical damping (the trapezoidal rule), to a method with a fair amount of numerical damping that effectively filters out spurious high frequency content dynamic behavior. The adoption of implicit integrators has led to clear benefits in other applications areas such as the well established Computer Aided Engineering (CAE) field. CAE simulations have to accommodate larger dimension models (up to 111 million degrees of freedom [2]) than the ones currently encountered in the typical protein folding problem, and in spite of the complexity and computational constraints faced, the most competitive multi-body dynamics simulation package (ADAMS), as well as the most popular linear and non-linear finite element codes (NASTRAN, ABAQUS) use implicit schemes. Since for the implicit HHT integration formulas considered in this work the stability is not limiting the step-size, the only attribute that factors into the step-size control process is the accuracy requirement. The proposed implicit integrator will contain an error-control mechanism that at all times during a simulation will determine the largest value of the step-size that yet attains the level of accuracy imposed by the user [9]. Thus, when the level of accuracy requested is crude, the integrator

will be capable of advancing the simulation with large steps and expeditiously produce results that can be later refined. The ability to control the integration step-size becomes essential when the physical system experiences rapid transients that are of short duration relative to the overall length of the simulation. In this case, if the step-size is kept fixed during simulation it will be limited to a value that can accommodate the most demanding transients that might be only briefly or periodically present in the physical process. When there are significant time intervals during which the nature of the problem is smooth and uneventful, being able to increase the step-size quickly and significantly (by orders of magnitude) is expected to result in significant CPU time reduction. When using an implicit method for the time integration of a dynamic system, the integration formula transforms the ordinary differential equation (ODE) problem into an equivalent problem of solving a set of non-linear algebraic equations. The solution of the non-linear system is found by means of a Newton-like method, where an iterative process gradually improves the quality of the numerical solution. As pointed out by Zhang and Schlick [16], the number of iterations for reaching acceptable accuracy with this approach is sometimes very large (tens or even hundreds), and it is quite often true that for non-linear high frequency systems the process might not converge at all. This problem can be effectively addressed following an idea discussed in depth in Lubich [7], where a very stiff force is replaced with a geometric constraint. Thus, rather than facing the convergence challenges posed by the non-linear algebraic solution, the dynamics of the problem qualitatively remain the same [14] if certain very strong and stiff covalent bonds are replaced with geometric constraints [11]. This treatment of stiff forces amounts to a generalization of the SHAKE procedure in GROMOS where we propose to adjust the positions of all particles simultaneously as part of the overall solution procedure employed to solve the nonlinear system that results from the implicit integration formulation. The proposed implicit integration method applied in the context of this equivalent constrained dynamics problem will answer two challenges: (a) it will allow for large integration steps (the new method will not be limited to small step-sizes on convergence failure grounds), and (b) it will efficiently and effectively handle the geometric constraints in an approach that has been used in the multi-body dynamics community for more than two decades [10], and it is at the foundation of the most widely used dynamics simulation package [8] in the CAE market. Jacobian and force computation are the two most CPU expensive MD tasks. While force computation is common to both explicit and implicit integration schemes, the Jacobian computation is only required by the latter. Here the Jacobian is used by the Newton method that solves the non-linear algebraic system obtained after discretization of the index three differential-algebraic equations (DAEs) by means of the HHT integrator. The Jacobian computation will not be carried out at each integration time-step as this approach was shown in practice to significantly increase the simulation CPU time [9]. Rather, the Jacobian is recycled for many time-steps, which effectively leads to a quasi-Newton method [1]. At the same time, in an MPI parallel computational framework [3], the state of the art in classical multibody dynamics simulations recommends using a separate computational thread for

continuous Jacobian update and factorization, thus effectively decoupling the tasks of numerical integration and Jacobian computation/factorization with additional CPU savings.

3 Models

Using the techniques of Molecular Dynamics it is possible to simulate the motion of any set of bodies between which the instantaneous forces can be computed. It is currently being used to address two large classes of problems: those in the Biological and Materials sciences. These two types of problems can be very different in the way in which they are formulated and what they require from a numerical integrator. In order to better probe the usefulness of the proposed methods we will test them on one model from each of the Biological and Materials sciences. The ability to simulate chemical and biological systems is already impacting the way in which we understand life and has an extremely rich potential for diagnosing and treating disease. Simulation of nucleic acids and proteins are of particular importance. Nucleic acids are used to store the genetic code and proteins provide structure to our tissues and are responsible for the many transport, storage and enzymatic processes that sustain life. Proteins are made of amino acids joined together by covalent peptide bonds. Finding which amino acids make up a certain protein and in what order they occur is a well established procedure known as sequencing. Once a proteins amino acid sequence, or primary structure, is known, the next step in understanding its function is to see how it hydrogen bonds to itself, which is known as the proteins secondary structure. The tertiary structure of a protein is how the secondary structure folds up on itself. Finally, some large proteins have quaternary structure which has to do with the interaction of different tertiary structures within a protein. Given the end goal of determining a proteins three dimensional structure from its primary sequence, there are several methods currently being employed. Experimental crystallography gives a direct look at higher order structure of proteins without referencing the sequence. Once several proteins have been sequenced and imaged, it is possible to infer how the primary structure of a protein affects higher order structures and function. There are disadvantages to this method, however, not the least of which is the large degree of uncertainty as to what causes a given primary structure to fold up in a certain way. Rather than relying on inferences, it may seem natural to attempt directly to simulate the folding dynamics of a protein. This would allow you to determine not only the final structure a protein assumes, but also the pathway by which reaches its final state. Modeling interactions of proteins with other molecules would also follow naturally from such a simulation framework. Finally, and perhaps most importantly, arbitrary modification (such as changing the amino acid sequence or solvent) would be exceedingly facile. With all these potential benefits, it should come as no surprise that Molecular Dynamics simulations of proteins is a very active area of research. The primary difficulty in performing the aforementioned simulations is not determining how the atoms within a protein interact, although some computational gains may be made by using valid assumptions to simplify

the calculation of forces. Rather, the difficulty comes in calculating the forces quickly enough (measured in CPU time) to be able to run a simulation on meaningful time and length scales. A typical amino acid has on the order of ten atoms and proteins can have several hundreds or even thousands of amino acids. It is simply not possible to simulate such a large system with existing numerical methods. The calculation of forces must be done at every time step of the simulation. Using numerical methods that are stable under long time steps is an indirect means of reducing the amount of force computation that must be performed per unit time being simulated. The biological portion of our study will focus on decalanine, a protein consisting of two alanine amino acids joined by a peptide bond. This simple system has many of the important features of larger proteins, is relatively inexpensive to simulate, and has been used in the past to test many aspects of Molecular Dynamics methods. The other large class of problems where Molecular Dynamics is currently being applied is those in Materials Science. Materials science deals with the design and testing of materials. Any situation that would benefit from having a material with certain properties is a potential application for Materials science. Materials Science increasingly depends on simulation for the virtual prototyping of materials. This process involves creating a representation of a material on the computer, exposing the material to specified conditions or tests and seeing how it reacts. It is essential that the computer representation of the material reacts in the same way as the real material would if it were exposed to the same tests. These computer experiments are designed to be cheaper, more flexible and faster than ordinary laboratory experiments. They also provide a more complete description of the situation than a simple measurement. The need for the computer experiments to be accurate means that a simulation must capture all relevant phenomena. In order to do so, the simulation must run for a time that is comparable to the timescale on which the phenomena are occurring. The current standard for integrators in Materials Science is stable only under very short time steps. The sheer number of iterations that would be necessary to run a simulation on large timescales makes using these methods impractical. The Materials Science portion of our study will focus on a lattice of atoms interacting via the Lennard-Jones potential. Lennard-Jones is a widely used interaction potential that comes very close to measurement data and has been used successfully in many Molecular Dynamics applications. Once our two models are in place, we will run several numerical experiments using our newly implemented integrators and compare the results for accuracy and CPU time against the current standard. The proposed implicit integrator will use large time steps to speed up the simulation but will avoid the common pitfalls of taking long time steps by ensuring not only stability, but also any level of accuracy inputted by the user. It does so by dynamically adjusting the time steps during the integration to suit the needs of the system. If the system is undergoing very fast vibrations, then the time step will be reduced to ensure a faithful representation of the motion. On the other hand, if the motion of the system is relatively smooth, or if the amplitude of the oscillation is within a user prescribed tolerance and thus can be safely ignored, the integration step-size is increased to reduce the number of time steps necessary

to complete the simulation.

4 Numerical Experiments

4.1 Mass Matrix versus Full Jacobian

This numerical experiment compares the use of a system's mass matrix versus a full Jacobian computed by numerical differencing. This matrix is used in the Newton Method to solve for corrections to the accelerations on each atom until the corrections are small. The results are summarized below.

Jacobian?	Step Size (fs)	Num. Steps	Simulation Time (ps)	CPU (s)
No	2.6	1923	5	331.46
Yes	6.2	806	5	366.96

The step sizes were chosen to be the maximum step size that was stable given the matrix that was used. The number of time steps was chosen such that each simulation would be approximately 5ps real time. The figures below show a plot of energy verus time for each experiment. The results indicate similar results quality and a slight time savings using just the mass matrix versus the full jacobian.

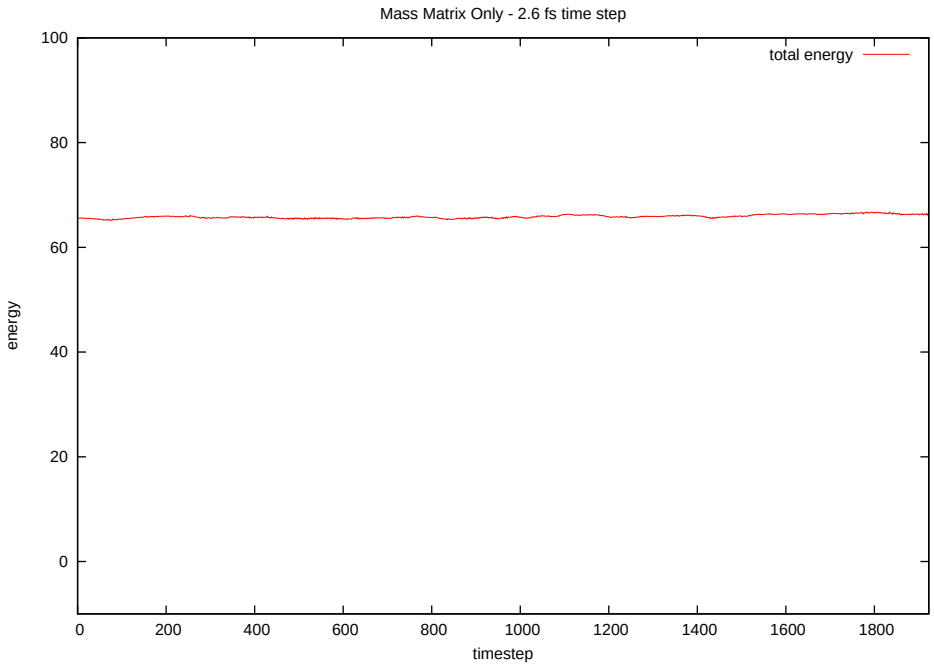

Figure 1: Mass Matrix Only

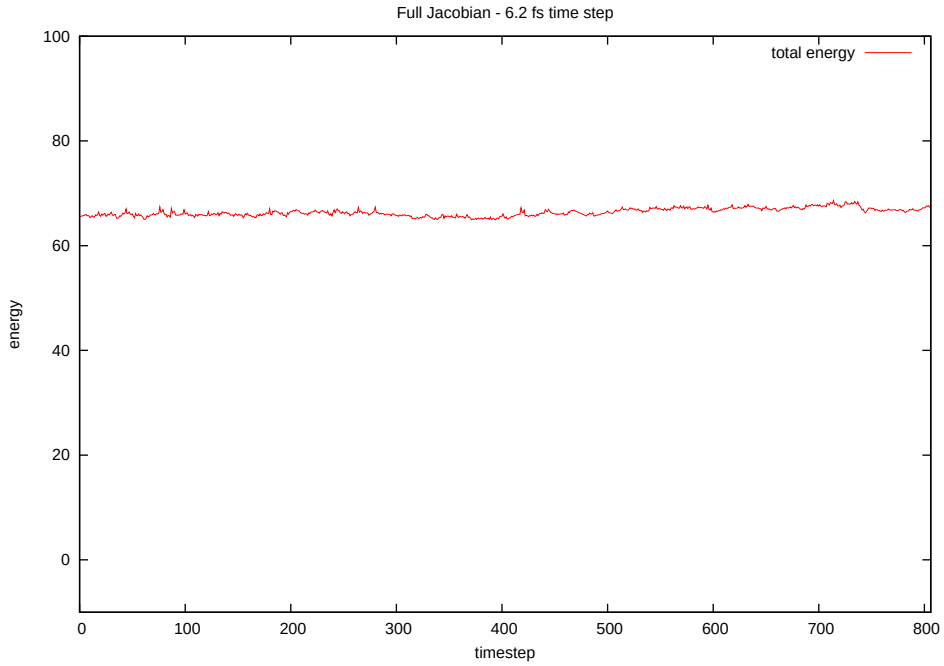

Figure 2: Full Jacobian

References

[1] K.E. Atkinson. *An introduction to numerical analysis*. John Wiley and Sons Inc., 1989.

[2] L. Elliott. Complex analyses need high-end fea. *Desktop engineering elements of analysis*, (10):4–7, 2004.

[3] W. Gropp, E. Lusk, and R. Thakur. *Using MPI-2: Advanced Features of the Message-Passing Interface*. MIT Press, 1999.

[4] E. Hairer, S.P. Norsett, and G. Wanner. Solving ordinary differential equations. *Computational Mathematics*, 1, 1987.

[5] H.M. Hilber, T.J.R. Hughes, and R.L. Taylor. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Eng. and Struct. Dynamics*, (5):283–292, 1977.

[6] T.J.R. Hughes. *Finite Element Method - Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall, 1987.

[7] C. Lubich. Integration of stiff mechanical systems by runge-kutta methods. *ZAMP*, (44):1022–1053, 1993.

[8] MSCsoftware. Msc.adams online help system. 2005.

[9] D. Negrut, R. Rampalli, G. Ottarsson, and A. Sajdak. On an implementation of the hilber-hughes-taylor method in the context of index 3 differential-algebraic equations of multibody dynamics. *J. Computational and Nonlinear Dynamics*, (2):73–82, 2007.

[10] N. Orlandea, M.A. Chace, and D.A. Calahan. A sparsity-oriented approach to the dynamic analysis and design of mechanical systems part i. *Transactions of the ASME Journal of Engineering for Industry*, pages 773–779, 1977.

[11] J.P. Ryckaert, G. Ciccotti, and H.J.C. Berendsen. Numerical integration of the cartesian equation of motion of a system with constraints: molecular dynamics of n-alkanes. *J. Comp. Phys.*, (23):327–341, 1977.

[12] T. Schlick. *Molecular Modeling and Simulation*. Springer, 2002.

[13] T. Schlick, R. Skeel, A. Brunger, L. Kale, J.A. Board Jr, J. Hermans, and K. Schulten. Algorithmic challenges in computational molecular biophysics. *Journal of Computational Physics*, (151):9–48, 1999.

[14] W.F. van Gunsteren and M. Karplus. Effects of constraints on the dynamics of macromolecules. *Macromolecules*, (15):1528–1544, 1982.

[15] L. Verlet. Computer experiments on classical fluids i.thermodynamical properties of lennard-jones molecules. *Phys. Rev.*, (159):98–103, 1967.

[16] G. Zhang and T. Schlick. Implicit discretization schemes for langevin dynamics. *Mol. Phys*, (84):1077–1098, 1995.