Technical Perspective
Building Robust Dynamical Simulation Systems

By Dinesh Manocha

THE FIELD OF classical dynamics is regarded as one of the success stories of applied mathematics. A dynamical simulation refers to simulation of a system of objects that move according to laws of physics. The computational power of modern computers has been widely used to predict the behavior of dynamical systems using computer simulations, which are often used as an adjunct to or substitute for a dynamical system when a simple closed form analytic solution is not possible.

Dynamical simulation systems are widely used in different applications, including industrial design or engineering simulation, where such systems are used to simulate the motion of moving parts or crash testing. Computer-based surgical simulators are also used to train medical students or physicians, as they can provide safe, realistic learning environments for repeated practice. Physics-based simulation techniques are increasingly used in computer graphics or animation to generate realistic motion of rigid or deformable objects and fluids. More recently, they are used in video games to generate realistic behaviors of objects.

One major focus in the field has been on building robust and accurate dynamical simulation systems using discrete computational capabilities of current computers. The motion of underlying objects is typically governed by differential equations corresponding to Newton's Second Law of Motion. These equations tend to be nonlinear and are solved using numerical methods. These methods step through a finite time interval and integrate the equations of motion for each object over that interval.

A key challenge in these dynamical systems is to handle the collisions or contacts between various objects or parts of an object. This process includes accurate detection of collisions between the boundaries or the interior meshes of the geometric models of the simulated objects. Furthermore, techniques for robust handling of various contacts, including resting or sliding contacts, are needed. It is well known that even a single missed collision can result in an invalid simulation and noticeable visual artifacts.

The following work by Harmon et al. on *asynchronous contact mechanics* presents a robust method to reliably simulate the contacts in a dynamical system. The paper poses three fundamental requirements for a reliable simulation: that no collisions are missed (a geometric requirement); that physically conserved quantities such as energy and momentum are numerically conserved (a numerical requirement); and that the computation terminates in finite time (an algorithmic requirement). The authors present an algorithm that guarantees fulfillment of all three of these requirements.

A key aspect of this work is robust collision handling. A collision between the primitives can be regarded as a discontinuity in the motion. Prior methods to detect collisions can be classified into two categories: retroactive methods that analyze the preceding time interval and check for contacts during that interval; or predictive methods that use some kind of motion bounds or continuous techniques to estimate the first time of contact between the geometric features in the future. Many conservative methods have been proposed in the literature based on physical properties or equations of motion, which can compute a lower bound on when such a contact may occur in the future, and this bound is used to compute the size of time step.

The authors propose a contact model in which the predictive step is performed in a decoupled or asynchronous manner for every boundary element or feature (for example, a triangle) of the objects. This approach makes it possible to choose varying or large time steps for some of the elements and still guarantees nonpenetration. Moreover, by using asynchronous variational integrators from the mechanics literature, along with this novel contact model, it becomes possible to handle configurations corresponding to sharp boundaries or dispersed points of contacts. The resulting contact handling algorithm can satisfy many challenging constraints corresponding to conservation of momentum and energy. This combination results in a simulation algorithm that is reliable and correct, and does not involve any tweaking of parameters to generate the desired motion. The authors have successfully demonstrated its performance on challenging benchmarks, such as complex deformable simulation of thin objects such as cloth, including tying ribbons into a reef knot.

The authors present a major advancement in terms of building robust dynamical simulation systems for deformable objects. In posing these three desiderata and a preliminary solution, many avenues of exploration are opened: Is asynchrony fundamentally required to satisfy the three requirements? Are there other integrators, perhaps implicit, that might be used instead? The proposed simulation algorithm can robustly perform complex simulations that are regarded as nontrivial for previous methods. However, for simpler scenarios the proposed algorithm is quite slow compared to the state of the art and can take many hours of CPU time to simulate just a few seconds of deformable motion. Could it be accelerated using better collision detection algorithms, including efficient culling methods based on a continuous collision formulation, or parallelization using multicore CPUs or many-core GPUs? In summary, this work by Harmon et al. opens the door to many new directions that together have the exciting potential to achieve the robust, accurate, and fast simulation of deformable models. ⓒ

Dinesh Manocha (dm@cs.unc.edu) is currently a Phi Delta Theta/Mason Distinguished Professor of computer science at the University of North Carolina at Chapel Hill. He is a Fellow of ACM, AAAS, and IEEE.