IOS Press , 2005

Bounds for Damping that Guarantee Stability in Mass-Spring Systems

Yogendra BHASIN [a] and Alan LIU [a,1]
[a] *The Surgical Simulation Laboratory*
National Capital Area Medical Simulation Center
Uniformed Services University
http://simcen.usuhs.mil

Abstract. Mass-spring systems are often used to model anatomical structures in medical simulation. They can produce plausible deformations in soft tissue, and are computationally efficient. Determining damping values for a stable mass-spring system can be difficult. Previously stable models can become unstable with topology changes, such as during cutting. In this paper, we derive bounds for the damping coefficient in a mass-spring system. Our formulation can be used to evaluate the stability for user specified damping values, or to compute values that are unconditionally stable.

Keywords. Mass-spring, stability bounds, Deformable models, Medical simulation,

1. Introduction

Mass-springs [1,2] are a common method for modeling deformable objects [3]. Mass-spring models are characterized by a network of point masses connected to its neighbors by massless springs. Individual masses may optionally be acted upon by external forces. A mass-spring system with n nodes can be described by the following equation,

$$m_i\, a_i^t = [\sum_{j \in N(i)} \kappa_{ij}\, \hat{d_{ij}^t}\, (l_{ij}^t - l_{ij}^0)\,] - \gamma_i\, v_i^t + f_{exti}^t \tag{1}$$

a_i^t is the acceleration of the mass m_i at time t. N(i) are all the neighbors of m_i, where a spring is connected between m_i and each neighbor. κ_{ij} is the stiffness coefficient of the spring connecting m_i and m_j. $\hat{d_{ij}^t}$ is the unit vector between m_i and m_j at time t. l_{ij}^t is the spring length between m_i and m_j at time t. l_{ij}^0 is the spring length between m_i and m_j at rest. γ_i is the viscous damping coefficient. v_i^t is the velocity of m_i at time t. f_{exti}^t is the external force acting on m_i at time t.

The first term in equation 1 denotes the internal force due to the tension of the springs connecting m_i with its neighbors. Viscous damping represents resistance due to air or any other medium in the system, and acts in the opposite direction of the velocity of the mass.

For a set of N points in 3D space, let X be a $3\ N \times 1$ column of position vectors at the current time. Then the mass-spring system can be expressed as, $M\ \ddot{X} = K\ X - Y\ \dot{X} + F_{ext}$ where, M and Y are $3N \times 3N$ diagonal mass and damping matrices respectively, F_{ext} is a $3\ N \times 1$ column of vectors representing external forces, K is a $3N \times 3N$ banded matrix of stiffness coefficients. This equation can be rewritten as a set of first order differential equations, and solved using standard numerical integration methods [4].

Mass-spring models are readily understood. Fast, efficient methods exist for solving them. Mass-spring models can be fairly large and complex without sacrificing real-time response to user interactions. For surgical simulation, the Explicit Euler scheme is widely used [5,6,7,8]. Explicit Euler integration has the benefits of simplicity and computational efficiency [9]. It can be used to update models used for both haptic and visual rendering [10].

[1]Correspondence to: Alan Liu, http://simcen.org/surgery.

Mass-spring models have disadvantages. Finding an appropriate set of parameters that is realistic can require considerable trial and error. Once a solution is found, a small time step may be necessary to maintain stability.

Previous work has addressed instability issues in various ways. Delingette [11] described a condition that can lead to numerical instability. For a dynamic mass-spring model with n nodes and total mass m_{total}, $k_c \approx \frac{m_{total}}{n\ \pi^2\ (\Delta t)^2}$, where Δt is the time step and k_c is the critical stiffness beyond which the system of equations is divergent. Provot [2] described a similar stability condition where the maximum allowed time step should be smaller than the natural period T_0 of the system, $T_0 \approx \pi\sqrt{\frac{m_i}{\kappa_i}}$. While these equations provide a better understanding of the relationship between mass-spring parameters and stability, it does not provide an accurate means of determining when instability occurs. Kacic-Alesic [12] described a stability condition for the Verlet integration method as, $\Delta t \leq 2\sqrt{\frac{m}{K}}$, where K is the combined spring coefficient of all the springs. However, the condition provided an excessively conservative bound.

In these attempts, stability was achieved at the cost of a smaller timestep, increasing computational complexity. In this paper, we approach the stability problem from a different perspective. Instead of changing the time parameter, we derive a relationship between the damping coefficient of a mass-spring system and its other three parameters (mass, stiffness coefficient, and time step). The bounds unconditionally guarantee a stable system. Section (2) describes the derivation, sections (3) and (4) outline our experiments and their results respectively. Sections (5) will show how these bounds can be used to preserve stability with minimal impact to the overall simulation. Section (6) summarizes our contribution.

2. Derivation

In this section, we motivate the rational behind our derivation, and provide the details of our formulation.Our derivation is based on Explicit Euler integration because it is widely used in medical simulation and other fields. The principal is not limited to this method however and can be easily extended to other numerical integration methods.

To simplify the discussion, we assume that the mass-spring system is unforced, and that the spring coefficient is the same for all springs, i.e., $\kappa_i = \kappa$ for $i \in [1..n]$. Equation (1) can then be rewritten as a standard second-order differential equation, i.e.,

$$m_i\ a_i^t + \gamma_i\ v_i^t - \kappa\ p_i^t = 0 \tag{2}$$

where, $p_i^t = \sum_{j \in N(i)} \hat{d_{ij}^t}\ (l_{ij}^t - l_{ij}^0)$. Using the Explicit Euler method, the position $x_i^{t+\Delta t}$ and velocity $v_i^{t+\Delta t}$ of the mass m_i at time $t + \Delta t$ is computed as,

$$v_i^{t+\Delta t} = v_i^t + (a_i^t)\Delta t \tag{3}$$

$$x_i^{t+\Delta t} = x_i^t + (v_i^{t+\Delta t})\Delta t \tag{4}$$

Lower bound for damping. Consider a mass-spring system with two nodes, m_0 and m_1. Let m_0 be fixed. Then equation (2) describes an unforced, damped system. The form of the solution to the characteristic equation (2) depends on the quantity $\gamma_1^2 - 4\ m_1\ \kappa$ [13].

When $\gamma_1^2 - 4\ m_1\ \kappa\ <\ 0$ the system is underdamped. An underdamped system is characterized by oscillatory motion. When $\gamma_1^2 - 4\ m_1\ \kappa \geq 0$, the system does not oscillate. However, the time to reach equilibrium increases as γ_1 increases. [13]. When $\gamma_1^2 - 4\ m_1\ \kappa = 0$, the system reaches equilibrium in the shortest time, and without oscillation.

In summary, underdamped systems oscillate. For large mass-spring systems, local oscillations are visually displeasing and should be avoided. Local oscillations can also produce cumulative errors that cause instability. By applying the constraint,

$$\gamma_i \geq 2\sqrt{m_i \kappa} \tag{5}$$

for $i \in [1, n]$, this situation is avoided. As will be shown in section 4, mass-spring systems that violate this constraint can also become unstable. Thus, equation (5) provides a lower bound for stability.

Upper bound for damping. When Explicit Euler integration is applied to solving the mass-spring equations, the physical properties of the system can be violated for large time steps. An explicit formulation of these constraints forms the basis for a derivation of an upper bound on the damping coefficient γ_i as a function of $m_i,\ \Delta t,\ v_i^t$ and κ.

Substituting the value for acceleration from equation (2) into equation (3) and rearranging the terms yield $v_i^{t+\Delta t} = v_i^t + (\frac{\kappa\ p_i^t - \gamma_i\ v_i^t}{m_i})\Delta t$. Let $v_{s\ i}^t = \frac{\kappa\ p_i^t}{m_i}\ \Delta t$ and $v_{d\ i}^t = \frac{\gamma_i\ v_i^t}{m_i}\ \Delta t$. Then $v_{s\ i}^t$ is the difference in velocity between time t and

$t + \Delta t$ due to spring forces. Similarly, $v^t_{d\ i}$ is the velocity difference due to damping between t and $t + \Delta t$. Damping dissipates energy from a mass-spring system. The nodes slow down as a result. Thus, between t and $t + \Delta t$, $v^t_{d\ i}$ can at most bring the node to a halt, it cannot cause the node to move faster. That is, $v^t_{d\ i}$ cannot be the dominant term. Expressing this mathematically, we get $|v^t_i + v^t_{s\ i}| \ \geq \ |v^t_{d\ i}|$. Expanding $v^t_{s\ i}$ and $v^t_{d\ i}$ yields, $|v^t_i + F^t_i \ \frac{\Delta t}{m_i}| \geq \frac{\gamma_i \Delta t}{m_i}|v^t_i|$, where $F^t_i = \kappa p^t_i$, is the internal spring force. So,

$$\gamma_i \leq \frac{|v^t_i \ \frac{m_i}{\Delta t} + F^t_i|}{|v^t_i|} \ for\ |v^t_i| \neq 0 \tag{6}$$

When $|v^t_i| = 0$, the damping coefficient has no effect on the system.

3. Experiments

Experiments were conducted to validate our theoretical development. A 20 x 20 cm planar mesh with 1300 nodes and 3750 edges was used. The mesh was aligned along the xz plane. The distance between the adjacent nodes was 0.57 cm. An additional spring was attached from each mass to its rest position so that the system always returned to its initial configuration. For all experiments, one or more nodes were perturbed by a specified amount, then the system was allowed to run until the initial configuration was restored. The model was assumed to be restored to its original configuration when all nodes were within 1% of the initial perturbation value. For each trial, all the spring coefficient and mass values were set to the same κ and m values repectively, i.e., $\kappa_i = \kappa$ and $m_i = m$ for $i \in [1..n]$.

Validating the lower bound. To validate the lower bound, an arbitrary node near the model's center was displaced by 5 cm from its rest position along the y-axis. Spring-meshes with a wide range of different parameters were tested. The spring coefficient and mass values were varied between 0.0001 to 0.1 in steps of 0.0001. For each combination of mass and spring coefficient, the damping coefficient was varied between 0 and the optimum damping value ($2\sqrt{m\kappa}$) in increments of 0.0001. The time step was set to 50 msec. In total, 1 million combinations of mass and spring coefficient values were tested. Each trial was allowed to run until either the model returned to the initial configuration, or until 1000 iterations were performed.

Validating the upper bound. We performed two set of experiments to validate the upper bound. In the first set, the damping was varied according to the upper bound value described by equation (6). In the second set, the damping was set to a value 5% larger than the upper bound. For each set of experiments, the spring coefficient and mass values were varied between 0.0001 to 0.1 in steps of 0.0001, resulting in 1 million trials. The time step was set to 50 msec. Again an arbitrary node was displaced by 5 cm from its rest position. Overdamped systems take a longer time to reach steady state. So, each trial was allowed to run for 10,000 Euler iterations, or until the initial configuration was restored.

4. Results

Validating the lower bound. This experiment evaluated a mesh with $m \in$ [0.0001, 0.1], $\kappa \in$ [0.0001, 0.1], $\gamma_i = \gamma \in$ [0, $2\sqrt{m\ \kappa}$] for i ∈ [1,n] and Δt = 50 msec. For all cases (1 million trials), the mesh behaved as predicted. When $\gamma \leq 2\sqrt{m\ \kappa}$, the mesh failed to return to its initial configuration and kept oscillating. Fig. 1 illustrates a typical case over time for $\kappa = 0.001$, $m = 0.001$ and γ = 0.0005, 0.001 and 0.002. The graph plots the absolute distances of the initially displaced node from its rest position during the simulation.

Validating the upper bound. The adaptive damping experiment examined a mesh with $m \in$ [0.0001, 0.1], $\kappa \in$ [0.0001, 0.1] and Δt = 50 msec. All one million trials produced similar results. Fig. 2 shows a typical case. The absolute distance of the initially displaced node from its rest position is plotted on a logarithmic (Base 10) scale when $\kappa = 0.1$ and $m = 0.005$. It is evident from the graph that the system with damping greater than the upper bound was unstable. Some nodes in the mesh did converge initially but ultimately became unstable. However, the system with the damping equal to the upper bound value achieved stability within 652 iterations and remained stable thereafter.

5. Discussion

In applications such as medical simulation, surgical procedures such as cutting and tearing can dynamically change the characteristics of a mass-spring model. Finding a set of parameters suitable for all situations can be challenging.

The bounds we have derived can ensure system stability while reducing the effort required to formulate plausible soft-tissue models. In this section, we discuss the significance of our results. While our discussion is made in the context of medical simulation, the implications are generalizable to other application areas.

Dynamic stability checks. Our results can be used to determine if a mass-spring system has become unstable during runtime. From equations (5) and (6), at time t, if $2\sqrt{m_i\kappa} > \frac{|v_i^t \frac{m_i}{\Delta t} + F_i^t|}{|v_i^t|}$, then no real value of damping exists that make the system stable. The ability to dynamically evaluate stability is especially useful if the model changes during runtime. A statically tuned model may exhibit different properties if its topology changes. To our knowledge, no comparable work exists that provides a closed-form solution to evaluating stability in this fashion.

Dynamic damping. The ability to evaluate stability at runtime is important. However, it may not be possible to tune a mass-spring model to exhibit the desired behavior and still be stable under all conditions. Accommodating all situations may require a suboptimal set of parameters that decreases realism. Equations (5) and (6) can be used in an adaptive damping scheme that permits the model to use user-specified damping values but when instability is detected, damping is clamped to either bound.

Limitations. Presently, our derivation describes bounds only for damping. The results cannot be used to derive practical bounds for other mass-spring parameters, such as time step. Work is ongoing to address this aspect.

6. Conclusion

This paper established a stability criterion for an unforced, damped mass-spring system using the Explicit Euler integration method as,

$$2\sqrt{m_i\kappa} \leq \gamma_i \leq \frac{|v_i^t \frac{m_i}{\Delta t} + F_i^t|}{|v_i^t|}$$

We conducted a series of experiments to validate our findings. For each case, validation was performed using one million different spring-mass configurations, with the smallest and largest values for each parameter differing by three orders of magnitude. In every case, the system behaved exactly as predicted by our equations. While it is impossible to evaluate all possible combinations of mass-spring parameters, these experiments lend greater confidence to the validity of our theoretical development.

Determining a suitable set of parameters for mass-springs can be difficult. Our results can reduce the efforts required to tune mass-springs and can be used to model any changes during runtime. The adaptive damping scheme can be used to restore stability when instability occurs.

The implications of our results were briefly discussed. Using our findings, it is possible to perform run-time instability detection, implement an adaptive damping-correction scheme to ensure system stability, and simplify the creation of realistic mass-spring models.

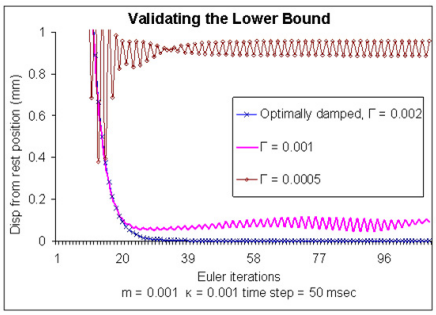

Figure 1. The behavior of a mesh with $\kappa = 0.001$, $m = 0.001$ and γ = 0.0005, 0.001 and 0.002 after a single initial impulse. γ below the lower bound (0.002) caused the mesh to oscillate indefinitely. At the computed lower bound, the mesh reached steady state after 34 iterations

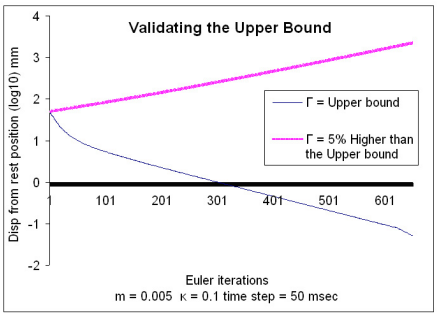

Figure 2. Validating the upper bound. The distances of the displaced node are in logarithmic (Base 10) scale

References

[1] D. Baraff and A. Witkin. Large steps in cloth simulation. In *Computer Graphics Proceedings, Annual Conference Series, SIGGRAPH, ACM Press*, pages 43–54, 1998.

[2] Xavier Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *Graphics Interface*, pages 147–154, 1995.

[3] Sarah Gibson and Brian Mirtich. A survey of deformable modeling in computer graphics. Technical Report TR-97-19, Mitsubishi Electric Research Lab, Cambridge MA, USA, 1997.

[4] D. Kincaid and W. Cheney. *Numerical Analysis: Mathematics of Scientific Computing*. 3rd edition, 2002.

[5] U. Kühnapfel, H. K. Çakmak, and H. Maaß. Endoscopic surgery training using virtual reality and deformable tissue simulation. *Computer Graphics*, 24:671–682, 2000.

[6] M. Bro-Nielsen, D. Helfrick, B. Glass, X. Zeng, and H. Connacher. Vr simulation of abdominal trauma surgery. In *Medicine Meets Virtual Reality: Art, Science, and Technology*, pages 117–123. IOS Press, 1998.

[7] M.H. Choi M.H. Kim Y.J. Choi, M. Hong. Adaptive mass-spring simulation using surface wavelet. In *Proceedings of International Conference on Virtual Systems and MultiMedia*, pages 25–33, 2002.

[8] Y. Lee, D. Terzopoulos, and K. Waters. Realistic modeling for facial animation. In *Proceedings of ACM SIGGRAPH*, volume 2, pages 55–62, 1995.

[9] A. Witkin. Physically based modeling. ACM SIGGRAPH Course Notes, 2001.

[10] L. P. Nedel and D. Thalmann. Real time muscle deformations using mass-spring systems. In *Proceedings of Computer Graphics International*, pages 156–165, Hannover, Germany, 1998.

[11] Hervé Delingette. Towards realistic soft tissue modeling in medical simulation. Technical Report RR-3506, INRIA, Sophia Antipolis, France, 2000.

[12] Z. Kacic-Alesic, M. Nordenstam, and D. Bullock. A practical dynamics system. In *Eurographics Symposium on Computer Animation*, pages 7–16, 2003.

[13] Leonard Meirovitch. *Elements of Vibration Analysis*. McGraw-Hill, 2nd edition, 1986.