Jiang, Schroeder, Selle, Teran, Stomakhin
The Affine Particle-In-Cell Method

1 Introduction

In the past, simulation for deformable materials (such as fluids) has been performed through the use of either Lagrangian (particle-based) or Eulerian (grid-based) methods. Particles tend to be better for resolving transport (*advection* in the case of fluids) and topological change, while grids tend to be better for computing collision reactions and updates due to forces like pressure and viscosity.

In order to acquire the advantages of both representations, many state-of-the-art methods utilize *hybrid* Lagrangian/Eulerian simulation, where the material is transformed back and forth between a particle representation and a grid representation such that kinematic steps can be done on particles and dynamic steps can be done on the grid.

Particle-In-Cell (PIC) and Fluid Implicit Particle (FLIP) are two hybrid methods which each have their own advantages and shortcomings. It's a bit of a Goldilocks problem, actually: PIC is stable but too dissipative, while FLIP is non-dissipative but noisy and sometimes unstable. Broadly, these errors are caused by transfer between the particle and grid representations.

PIC PIC directly interpolates particles from the grid, which leads to a loss of information and thereby dissipation. This dissipation also causes a loss of angular momentum; visually, this means an rotating object in free fall might (disturbingly, if you can imagine it!) just stop rotating.

FLIP FLIP, instead of directly interpolating particles from the grid, transfers increments of velocities and displacements [w.r.t. the original particle values]. However, augmenting the original values in this fashion is unsafe and can result in instability.

Note: since there are usually more particles than grid nodes, some particles are not seen by the grid and thus get no physical response. This phenomenon is called **ringing instability** and causes positional artifacts such as noise and clumping. This is more of a problem in FLIP than PIC.

PIC/FLIP We can blend PIC and FLIP in order to achieve a simulation that is both stable and energetic. In the blend, a bias toward PIC avoids instability at the expense of dissipation, while a bias toward FLIP avoids dissipation at the expense of noise and [potential] instability.

However, it can be difficult to strike a balance between the two. Therefore, in this paper we explore a *third* option for particle/grid transfer, APIC, which fixes PIC's information loss by describing each particle's velocity in a locally affine (not locally constant) manner. This stably removes PIC's dissipation and also allows for conservation of angular momentum across particle/grid transfers. Essentially, APIC is both more energetic than the original PIC and more stable than PIC/FLIP. It ends up being very useful in MPM simulations.

2 Particle-Grid Transfer

The difference between the methods has to do with the details of transfer between particles and grid. In PIC, all data flows through the grid. In FLIP, there is an additional data path from the original particle state, which reduces dissipation but can lead to instability.

When it comes to PIC, a single particle often receives data from multiple grid points, but is forced to reduce those influences to a single constant value. This leads to information loss and thereby dissipation. To remedy this, we allow particles to track a full affine representation of the local grid data.

This scheme (APIC) allows us to control noise, because the pure filter property of PIC is retained (i.e. all information is forced through the grid). And, as just mentioned, it minimizes information loss by enriching each particle with a 3×3 matrix giving a locally affine description of the flow.

APIC

A quick word on notation: a subscript p denotes a particle, a subscript of i denotes a grid index, a superscript of n means that the quantity is available at the beginning of the time step, and a superscript of $n+1$ means that the quantity is available at the beginning of the *next* time step. Lowercase unbolded quantities are scalars. Lowercase **bolded** quantities are vectors. **Uppercase** bolded quantities are matrices.

One more time: we want to prevent the loss of information that's causing all the damping. Therefore we enrich our velocity representation as being locally affine on each particle.

For this we introduce a matrix $\mathbf{C}_p^n$ for each particle which contains the particle's velocity derivatives. Then the local velocity represented by a particle at the grid position $\mathbf{x}_i$ is

$$\mathbf{v}_p^n + \mathbf{C}_p^n(\mathbf{x}_i - \mathbf{x}_p^n)$$

where $\mathbf{v}_p^n$ and $\mathbf{x}_p^n$ are, respectively, the velocity and position of particle p.

To transfer from particles to grid:

$$m_i^n \mathbf{v}_i^n = \sum_p w_{ip}^n m_p (\mathbf{v}_p^n + \mathbf{B}_p^n (\mathbf{D}_p^n)^{-1} (\mathbf{x}_i - \mathbf{x}_p^n))$$

where (in order of appearance) m_i^n is the mass at cell i, $\mathbf{v}_i^n$ is the velocity at cell i, w_{ip}^n is a weight for cell i and particle p, m_p is the mass of particle p, and $\mathbf{v}_p^n$ is the velocity of particle p. $\mathbf{B}_p^n$ is the affine state of particle p. $\mathbf{D}_p^n$ is analogous to an inertia tensor, and is defined as

$$\mathbf{D}_p^n = \sum_i w_{ip}^n (\mathbf{x}_i - \mathbf{x}_p^n)(\mathbf{x}_i - \mathbf{x}_p^n)^T$$

which can derived through the preservation of affine motion. Note that $\mathbf{C}_p^n = \mathbf{B}_p^n (\mathbf{D}_p^n)^{-1}$.

To transfer from grid to particles:

$$\mathbf{B}_p^{n+1} = \sum_i w_{ip}^n \tilde{\mathbf{v}}_i^{n+1} (\mathbf{x}_i - \mathbf{x}_p^n)^T$$

where $\tilde{\mathbf{v}}_i^{n+1}$ is the intermediate velocity at cell i.

This scheme preserves both affine velocity fields and angular momentum during each transfer. Hence our rotations will be resolved correctly.

Some limitations: APIC doesn't actually do anything about ringing instability. Also, we have to store an extra matrix per particle and perform a few extra operations during the transfers. However, the difference in cost is negligible in practice.

References

[1] C. Jiang, C. Schroeder, A. Selle, J. Teran, and A. Stomakhin. 2015. The Affine Particle-In-Cell Method. *ACM Trans. Graph.* 34, 4, Article 51 (July 2015), 10 pages.