ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2022
D. L. Michels and S. Pirk
(Guest Editors)

Volume 41 (2022), Number 8

Stability analysis of explicit MPM

Song Bai[1] and Craig Schroeder[1]†

[1]University of California, Riverside

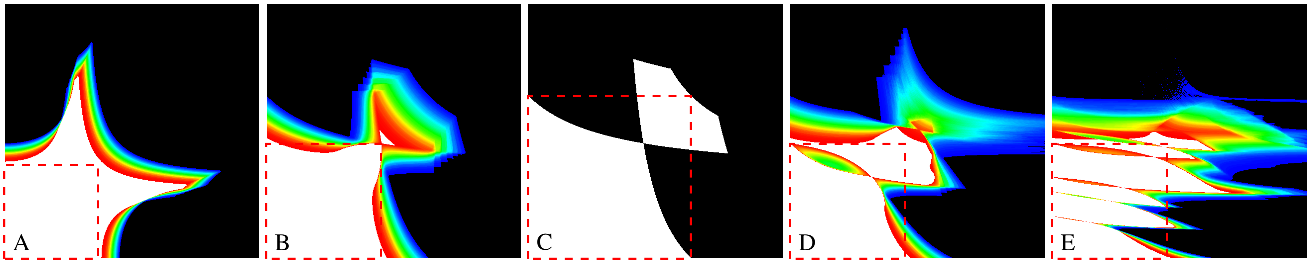

Figure 1: *Destabilizing effects of cycling between two time step sizes on stability with one particle. Δt_0 is the x axis and Δt_1 is the y axis. The dashed box indicates the stable region based on constant time steps sizes. For images A-C, time steps alternate: $\Delta t_0, \Delta t_1, \Delta t_0, \Delta t_1, \ldots$ with PIC (A), APIC (B), and CPIC (C). In (D), APIC is run with $\Delta t_0, \Delta t_1, \Delta t_1, \Delta t_0, \Delta t_1, \Delta t_1, \ldots$. In (E), Δt_0 is followed by 6 time steps at Δt_1 using APIC. In all cases, quadratic splines are used in 3D. All plots are on the same scale. White is stable, and black is unstable. If stability is particle position dependent, colors indicate the likelihood of any particular position being stable, with red being likely stable and blue being likely unstable. The stability region can become quite complex with unstable time step sequences scattered throughout the predicted stable region.*

Abstract

In this paper we analyze the stability of the explicit material point method (MPM). We focus on PIC, APIC, and CPIC transfers using quadratic and cubic splines in two and three dimensions. We perform a fully three-dimensional Von Neumann stability analysis to study the behavior within the bulk of a material. This reveals the relationship between the sound speed, CFL number, and actual time step restriction and its dependence on discretization options. We note that boundaries are generally less stable than the interior, with stable time steps generally decreasing until the limit when particles become isolated. We then analyze the stability of a single particle to derive a novel time step restriction that stabilizes simulations at their boundaries. Finally, we show that for explicit MPM with APIC or CPIC transfers, there are pathological cases where growth is observed at arbitrarily small time steps sizes. While these cases do not necessarily pose a problem for practical usage, they do suggest that a guarantee of stability may be theoretically impossible and that necessary but not sufficient time step restrictions may be a necessary and practical compromise.

CCS Concepts

• ***Computing methodologies*** → ***Physical simulation;***

1. Introduction

The material point method (MPM) was introduced into computer graphics for simulation snow [SSC*13]. Since then, MPM has grown significantly in popularity as a method for handling a wide variety of complex phenomena. These materials often involve plasticity, which is challenging to handle implicitly [YSB*15,KGP*16, TGK*17,LLJ22]. This has created interest in explicit MPM formulations, which handle plasticity easily and are quite effective when stiffness is not too high.

Implicit methods are generally favored in graphics due to their stability [BW98]. Unlike explicit methods, which often require small time steps sizes to maintain stability, implicit methods tend to be stable at very large time step sizes and are often run at one (or a few) time steps per frame, though smaller time steps may be required to resolve collisions or improve accuracy. For explicit methods, time steps must be chosen based on stability considerations. Nevertheless, explicit MPM formulations are widely used (and are

† craigs@cs.ucr.edu

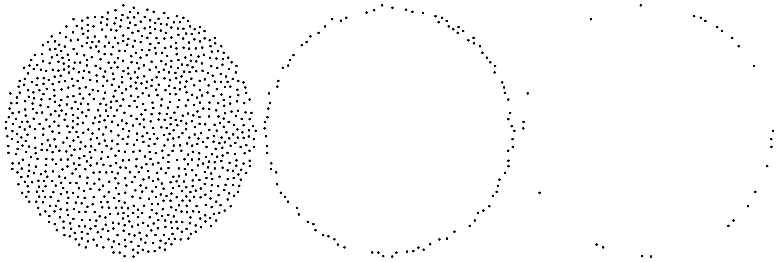

Figure 2: *(Left) This spinning disk is destabilized slightly by its boundary, requiring it to be run with a smaller time step. (Middle) If the disk is made very thin, a smaller time step is required for stability. (Right) In the limit as more particles are pruned out, particles eventually become isolated, and the single particle stability time step is required.*

the norm outside graphics) since they greatly simply the treatment of plasticity, which is difficult to handle accurately and efficiently with implicit methods. In practice, one often selects a fixed time step size by trial and error that is sufficient to ensure stability for a particular simulation [YSB*15, KGP*16]. More recently, there has been interest in using adaptive time step sizes based on the classical CFL restriction for explicit MPM [FHHJ18, SSS20]. The classical CFL restriction requires that the numerical wave speed must be at least as high as the physical wave speed. Although this is not a guarantee of stability, it is necessary [Str04]. A simulation that does not follow its CFL restriction can be expected to explode.

Stable time step selection is a fundamental challenge for numerical simulation, especially MPM [Bra16], and many tools have been developed to analyze stability for entire classes of numerical methods. Von Neumann analysis is the gold standard for proving stability for linear finite difference schemes. It is straightforward to use in this context and provides a guarantee of stability under some circumstances [Str04]. Although Von Neumann analysis assumes a linear numerical scheme and a linear partial differential equation, it can be adapted to nonlinear problems by linearizing about an equilibrium configuration. Indeed, Von Neumann analysis has been applied to MPM a few times, but this has been limited to 1D [NZ20, Gri14b], though [NZ20] does suggest some ways to apply the results to higher dimensions. In this paper, we perform a fully three-dimensional Von Neumann stability analysis on MPM near the rest configuration. This analysis includes the full algorithm with APIC [JSS*15] or CPIC [HFG*18] transfers, which has never been done in any dimension. This analysis justifies the use of sound-speed-based time step restrictions and provides insight on how details of the discretization affect the CFL numbers that can be used.

Algorithm 1 Explicit APIC time integration scheme.

```
procedure TIME_STEP
    $m_i^n \leftarrow \sum_p w_{ip}^n m_p$
    $m_i^n \mathbf{u}_i^n \leftarrow \sum_p w_{ip}^n m_p (\mathbf{u}_p^n + \mathbf{C}_p^n (\mathbf{x}_i^n - \mathbf{x}_p^n))$
    $\mathbf{f}_i^n \leftarrow -\sum_p V_p^0 (\mathbf{P}(\mathbf{F}_p^n))(\mathbf{F}_p^n)^T \nabla w_{ip}^n$
    $\tilde{\mathbf{u}}_i^{n+1} \leftarrow \mathbf{u}_i^n + \Delta t (m_i^n)^{-1} \mathbf{f}_i^n$
    $\mathbf{u}_p^{n+1} \leftarrow \sum_i w_{ip}^n \tilde{\mathbf{u}}_i^{n+1}$
    $\mathbf{x}_p^{n+1} \leftarrow \mathbf{x}_p^n + \Delta t \mathbf{u}_p^{n+1}$
    $\nabla \mathbf{u}_p^{n+1} \leftarrow \sum_i \tilde{\mathbf{u}}_i^{n+1} (\nabla w_{ip}^n)^T$
    $\mathbf{F}_p^{n+1} \leftarrow (\mathbf{I} + \Delta t \nabla \mathbf{u}_p^{n+1}) \mathbf{F}_p^n$
    $\mathbf{C}_p^{n+1} \leftarrow \xi \sum_i w_{ip}^n \tilde{\mathbf{u}}_i^{n+1} (\mathbf{x}_i^n - \mathbf{x}_p^n)^T$
```

Transfers and averaging between particles and grid have a generally stabilizing effect on MPM. On the other hand, reduced numbers of neighboring particles can often reduce stability. Indeed, MPM simulations are often observed to explode from the boundary rather than the interior. We investigate the effects of boundaries on MPM stability, which naturally leads to the limiting case where a particle becomes isolated. Indeed, a simulation with an isolated particle is actually less stable than a large system of particles. Single particle stability was a special consideration in the development of APIC [JSS*15]. It was also noted in [SSS20] that fluid simulations could become unstable if particles became isolated. They proposed a time step restriction to avoid the single particle instability in the fluid case, noting that the instability also affects solids. In this paper, we take a closer look at the single particle instability and propose a time step restriction in the solid case.

One of the big surprises we encountered during the analysis in this paper was the destabilizing effects of variable time step sizes on the stability of isolated particles when APIC or CPIC transfers are used. Conventional wisdom suggests the existence of a time restriction below which time step sizes may be chosen at will; this is a fundamental assumption that underlies adaptive time step sizes and time step restrictions. We show that this might not be true for MPM under some circumstances.

2. Related work

The material point method was developed as an extension of early hybrid particle/grid methods [Har64] to handle elastic solids [SCS94, SZS95]. Early hybrid methods were quickly found to suffer from a variety of instabilities, such as ringing [Oku72, Bra88, BL98, BK04, SKB08, Gri14a]. This lead to studies into the stability properties of particle-grid transfers, which lead to the development of better interpolation functions [BK04, WG08]. Although the original PIC was effectively implicit, being a projection-style fluid solver, early MPM methods were all explicit. Implicit formulations of MPM were developed to improve stability and take larger time steps [GW01, GW03, SK04, LS06]. Despite these advantages, explicit methods are very popular in the MPM community due to their simplicity, accuracy, and the need to take small time steps anyway for many applications.

MPM was introduced to the computer graphics community by [SSC*13] as a method for simulating snow. Within the community, MPM has grown significantly in popularity. It is used for phase change [SSJ*14], foams and gels [YSB*15, RGJ*15], dry sand [DBD16, KGP*16], wet sand [TGK*17, GPH*18], coupling with rigid bodies [HFG*18], fracture [WFL*19, WDG*19], anisotropic materials [JYM*20, CLR*20], and other complex materials [FLGJ19].

Early experiences within the community quickly discovered limitations when trying to apply the method to graphics applications, which has lead to efforts within the community to improve upon MPM, such as less noisy and less dissipative transfers [JSS*15, FGG*17], and better time integration [JST17, GSS*15, WLF*20]. Improvements in the method have in turn lead to an expansion in its adoption.

Due to the need for efficiency and stability, MPM was introduced into graphics as an implicit method [SSC*13], and most methods within graphics since have been implicit. However, implicit MPM is significantly more complex to implement [YSB*15]. Plasticity is especially difficult to solve in an implicit context due to the asymmetry of the linear systems that must be solved [KGP*16], though some progress has been made towards addressing this [Li22]. Similar difficulties exist for other phenomena, especially contacts, that are occasionally modeled using plasticity [JGT17, GHF*18, DS19, HGG*19]. This has lead to interest in explicit MPM formulations [YSC*18, FHHJ18, SSS20].

3. Stability analysis

Our principle objective in this paper is to examine the stability characteristics of explicit MPM. For this purpose, we begin with the explicit APIC scheme shown in Algorithm 1, adopting the notation used in [JSS*15]. We restrict our analysis to the case of quadratic or cubic splines for the transfer weights. These are the most common choices in graphics and they have the advantage that $\mathbf{D}_p^{n+1} = \sum_i w_{ip}^n (\mathbf{x}_i^n - \mathbf{x}_p^n)(\mathbf{x}_i^n - \mathbf{x}_p^n)^T = \frac{1}{\xi}\mathbf{I}$ takes a particularly simple form, where $\xi = \frac{4}{\Delta x^2}$ for quadratic and $\xi = \frac{3}{\Delta x^2}$ for cubic, leading to the simple update rule for $\mathbf{C}_p^{n+1}$ used in Algorithm 1 (See [JSS*15] for details). The analysis could be extended to cover other transfer kernels (such as GIMP [BK04, WG08, Wal09, NG15]), but we will not pursue this here.

We consider both 2D and 3D. We also consider PIC and CPIC [HFG*18] transfers, which we are able to do without repeating too much of the analysis. In particular, PIC can obtained by dropping the $\mathbf{C}_p^n$ portions (or by setting $\xi = 0$). CPIC can be obtained by substituting $\nabla w_{ip}^n \to \xi w_{ip}^n (\mathbf{x}_i^n - \mathbf{x}_p^n)$ everywhere it occurs. We color code the portions that differ for PIC or CPIC to make the differences clear at a glance. We also summarize the conclusions separately for each. We do not consider alternative transfer schemes such as FLIP [BR86, BKR88, SCS94], PIC/FLIP blends [SSC*13], or XPIC [HN17], which would generally be expected to have radically different stability profiles.

3.1. Von Neumann analysis

Von Neumann analysis is the simplest approach for proving stability for linear finite difference schemes under periodic boundary conditions. The idea behind Von Neumann analysis is to take the Fourier transform of the discretized PDE. When this is done, different wave-numbers decouple from one another and grow or decay with different amplification factors g. The scheme is stable if all of the amplification factors satisfy $|g| \le 1$. For simplicity, we ignore the complications surrounding $|g| = 1$ throughout this paper.

Basic ideas. To get a general feel for how the procedure is applied in practice, we first apply it to the advection equation $u_t + cu_x = 0$ in 1D, which we discretize with forward-time-forward-space as $\frac{u_j^{n+1} - u_j^n}{\Delta t} + c\frac{u_{j+1}^n - u_j^n}{\Delta x} = 0$. We begin by fixing a wave-number k and let $u_j^n = u(j\Delta x, n\Delta t) = g^n e^{ikj\Delta x}$. Here, i is the imaginary number when it does not occur in an index, and g^n is a power of g. Substituting this into the discretized equation yields $\frac{g^{n+1}e^{ikj\Delta x} - g^n e^{ikj\Delta x}}{\Delta t} + c\frac{g^n e^{ik(j+1)\Delta x} - g^n e^{ikj\Delta x}}{\Delta x} = 0$. Solving for g gives us $g = 1 + \frac{c\Delta t}{\Delta x}(1 - e^{ik\Delta x})$. Let $\nu = \frac{c\Delta t}{\Delta x}$ so that $|g|^2 = 1 + 2\nu(\nu+1)(1 - \cos(k\Delta x))$. To have $|g| \le 1$ for all k we must have $\nu(\nu+1) \le 0$ so that $-1 \le \nu \le 0$. That is, the scheme is only stable if $c \le 0$ and $\Delta t \le \frac{\Delta x}{|c|}$. This classical conclusion mirrors the CFL restriction. Here, c is the advection velocity. Since the scheme only looks right (it uses u_{j+1}^n but not u_{j-1}^n), it can only be stable if information moves right to left ($c \le 0$). Since at each time step information travels only one grid node, the numerical wave speed is $\frac{\Delta x}{\Delta t}$, so that $\frac{\Delta x}{\Delta t} \le c$ is the time step restriction.

MPM Von Neumann setup. In the simple example above, our state was a simple scalar u_i^n that lives on a regular grid with pe-

$$h_q = \sum_i w_{ip}^n e^{i(\mathbf{x}_p^n - \mathbf{x}_i^n)\cdot\mathbf{z}}$$

$$\mathbf{k}_q = \sum_i \nabla w_{ip}^n e^{i(\mathbf{x}_p^n - \mathbf{x}_i^n)\cdot\mathbf{z}} \qquad \mathbf{e}_q = \xi \sum_i w_{ip}^n (\mathbf{x}_i^n - \mathbf{x}_p^n) e^{i(\mathbf{x}_p^n - \mathbf{x}_i^n)\cdot\mathbf{z}}$$

$$w = \sum_p w_{ip}^n = N_g \qquad b = \sum_q (\overline{h_q} h_q + \xi^{-1} \overline{\mathbf{e}_q}^T \mathbf{e}_q)$$

$$\mathbf{b} = \sum_p w_{ip}^n (\mathbf{u}_q + \mathbf{C}_q(\mathbf{x}_i^n - \mathbf{x}_p^n)) e^{i(\mathbf{x}_p^n - \mathbf{x}_i^n)\cdot\mathbf{z}} = \sum_q (h_q \mathbf{u}_q + \xi^{-1} \mathbf{C}_q \mathbf{e}_q)$$

$$\mathbf{B} = \sum_p \nabla w_{ip}^n \overline{\mathbf{k}_q}^T e^{i(\mathbf{x}_p^n - \mathbf{x}_i^n)\cdot\mathbf{z}} = \sum_q \mathbf{k}_q \overline{\mathbf{k}_q}^T$$

$$\mathbf{c} = \sum_p \mathbf{K}_q \nabla w_{ip}^n e^{i(\mathbf{x}_p^n - \mathbf{x}_i^n)\cdot\mathbf{z}} = \sum_p \mathbf{K}_q \mathbf{k}_q$$

$$\mathbf{M} = \frac{\partial \mathbf{P}}{\partial \mathbf{F}} \qquad \mathbf{K}_q = \mathbf{M} : \mathbf{F}_q \qquad A_{ik} = M_{ijkl} B_{jl}$$

$$\tau = \frac{\Delta t g}{g - 1} \qquad \mathbf{u} = \frac{\mathbf{b}m - \Delta t V^0 \mathbf{c}}{wmg} \qquad \beta = \frac{\Delta t V^0}{m(b - wg)}$$

$$s = \frac{b}{w} \qquad c = \frac{\Delta t^2 V^0}{mw} \qquad r = \frac{c\sigma - s - 1}{2}$$

Figure 3: *Definitions of intermediates. Note that* $\mathbf{b}$, w, $\mathbf{B}$, *and* $\mathbf{c}$ *lack grid indices. Since all grid cells are the same, these quantities do not explicitly depend on the grid index.* $\overline{\mathbf{k}_q}$ *is the complex conjugate of* $\mathbf{k}_q$, *so that* $\mathbf{B}$ *is Hermitian.*

riodic boundaries. The situation with MPM is far more complex. The PDE is nonlinear, and there is much more state, which lives on potentially moving particles. In order to apply Von Neumann analysis, we must assume a (quite restrictive) setup. (A1) All cells contain the same number of particles in the same arrangement. (A2) Periodic boundary conditions are used. (A3) All particles are identical; in particular, they have the same mass ($m_p = m$) and volume ($V_p^0 = V^0$). (A4) The configuration is near the rest configuration, so that $\mathbf{u}_p^n \approx \mathbf{0}$ and positions can be considered to be stationary. (A5) The particle distribution is symmetrical in the sense that for each grid node i and particles p there is a particle $\overline{p}$ such that $(\mathbf{x}_{\overline{p}}^n - \mathbf{x}_i^n) = -(\mathbf{x}_p^n - \mathbf{x}_i^n)$. Stated another way, if the grid is mirrored along the x, y, and z directions, there will be particles in the same locations. This assumption will simplify parts of the analysis by making certain quantities real-valued.

We will use index p to refer to a per-particle quantity ($\mathbf{u}_p^n$). Since each cell has the same particle layout (by assumption (A1)), we introduce the index q to refer to a particle within some *canonical* cell. This is useful for quantities that are not constant across particles but are the same for corresponding particles in different grid cells. Each p has a unique q associated with it, and each q corresponds with a p for each grid cell.

Initial particle state. The analysis begins by fixing a wavenumber $\mathbf{z}$ and setting the particle state to be of the appropriate form.

$$\mathbf{u}_p^n = \varepsilon g^n \mathbf{u}_q e^{i\mathbf{x}_p^n \cdot \mathbf{z}} \qquad \mathbf{C}_p^n = \varepsilon g^n \mathbf{C}_q e^{i\mathbf{x}_p^n \cdot \mathbf{z}} \qquad \mathbf{F}_p^n = \mathbf{I} + \varepsilon g^n \mathbf{F}_q e^{i\mathbf{x}_p^n \cdot \mathbf{z}}.$$

Here, ε represents a small perturbation from equilibrium. At equilibrium ($\varepsilon = 0$), the particles are stationary ($\mathbf{u}_p^n = \mathbf{0}$, $\mathbf{C}_p^n = \mathbf{0}$) and experience no strain ($\mathbf{F}_p^n = \mathbf{I}$) (assumption (A4)). In the definitions above, we have also introduced $\mathbf{u}_q$, $\mathbf{C}_q$, and $\mathbf{F}_q$. These provide a magnitude and direction for the perturbation, which may be different within a cell. Note the use of the index q. As in the motivating example, g^n is a power of the growth factor g and i is the imaginary number when not used as an index.

During the course of the Fourier analysis, we will find it convenient to introduce many intermediate quantities in order to keep expressions manageable. The definitions of all of these intermediates are listed in Figure 3 for easy reference.

P2G transfers. We begin the time step by transferring particle state to the grid.

$$m_i^n = \sum_p w_{ip}^n m_p = wm$$

$$m_i^n \mathbf{u}_i^n = \sum_p w_{ip}^n m_p (\mathbf{u}_p^n + \mathbf{C}_p^n (\mathbf{x}_i^n - \mathbf{x}_p^n))$$

$$\mathbf{u}_i^n = \frac{\varepsilon g^n}{w} e^{i\mathbf{x}_i^n \cdot \mathbf{z}} \sum_p w_{ip}^n (\mathbf{u}_q + \mathbf{C}_q (\mathbf{x}_i^n - \mathbf{x}_p^n)) e^{i(\mathbf{x}_p^n - \mathbf{x}_i^n) \cdot \mathbf{z}} = \frac{\varepsilon g^n \mathbf{b}}{w} e^{i\mathbf{x}_i^n \cdot \mathbf{z}}$$

Note that the grid velocity follows the same wave form as the state variables ($e^{i\mathbf{x}_p^n \cdot \mathbf{z}}$ vs. $e^{i\mathbf{x}_i^n \cdot \mathbf{z}}$); all of the quantities will take this form. The intermediate $\mathbf{b}$ provides the direction for perturbations to $\mathbf{u}_i^n$.

Forces. With transfers completed, we must propagate our perturbations through the force computations. We compute forces through the first Piola-Kirchhoff stress tensor $\mathbf{P}$. $\mathbf{M}$ is the unperturbed stress derivative (See Figure 3). $\mathbf{K}_q$ is the direction of stress perturbation. Neglecting higher order terms in ε (so that we are effectively analyzing linear elasticity), the stress is

$$\mathbf{P}(\mathbf{F}_p^n) = \mathbf{P}(\mathbf{I}) + \mathbf{M} : (\mathbf{F}_p^n - \mathbf{I}) = \varepsilon g^n e^{i\mathbf{x}_p^n \cdot \mathbf{z}} (\mathbf{M} : \mathbf{F}_q) = \varepsilon g^n e^{i\mathbf{x}_p^n \cdot \mathbf{z}} \mathbf{K}_q$$

We next compute grid forces from the stresses

$$\begin{aligned} \mathbf{f}_i^n &= -\sum_p V_p^0 (\mathbf{P}(\mathbf{F}_p^n))(\mathbf{F}_p^n)^T \nabla w_{ip}^n \\ &= -\sum_p V_p^0 (\varepsilon g^n e^{i\mathbf{x}_p^n \cdot \mathbf{z}} \mathbf{K}_q)(\mathbf{I} + \varepsilon g^n \mathbf{F}_q e^{i\mathbf{x}_p^n \cdot \mathbf{z}})^T \nabla w_{ip}^n \\ &= -\varepsilon V^0 g^n \sum_p e^{i\mathbf{x}_p^n \cdot \mathbf{z}} \mathbf{K}_q \nabla w_{ip}^n = -\varepsilon V^0 g^n e^{i\mathbf{x}_i^n \cdot \mathbf{z}} \mathbf{c}. \end{aligned}$$

Grid and particle velocities. We can now apply forces to the grid.

$$\begin{aligned} \tilde{\mathbf{u}}_i^{n+1} &= \mathbf{u}_i^n + \Delta t (m_i^n)^{-1} \mathbf{f}_i^n \\ &= \frac{\varepsilon \mathbf{b}}{w} g^n e^{i\mathbf{x}_i^n \cdot \mathbf{z}} - \Delta t (wm)^{-1} \varepsilon V^0 g^n e^{i\mathbf{x}_i^n \cdot \mathbf{z}} \mathbf{c} = \varepsilon g^{n+1} e^{i\mathbf{x}_i^n \cdot \mathbf{z}} \mathbf{u} \end{aligned}$$

Transferring back to particles,

$$\mathbf{u}_p^{n+1} = \sum_i w_{ip}^n \tilde{\mathbf{u}}_i^{n+1} = \sum_i w_{ip}^n \varepsilon g^{n+1} e^{i\mathbf{x}_i^n \cdot \mathbf{z}} \mathbf{u} = \varepsilon g^{n+1} \overline{h_q} e^{i\mathbf{x}_p^n \cdot \mathbf{z}} \mathbf{u}.$$

From our original assumption on the form for $\mathbf{u}_p^{n+1}$ we have $\mathbf{u}_p^{n+1} = \varepsilon g^{n+1} \mathbf{u}_q e^{i\mathbf{x}_p^n \cdot \mathbf{z}}$, so that $\varepsilon g^{n+1} \mathbf{u}_q e^{i\mathbf{x}_p^n \cdot \mathbf{z}} = \varepsilon g^{n+1} \overline{h_q} e^{i\mathbf{x}_p^n \cdot \mathbf{z}} \mathbf{u}$ and thus $\mathbf{u}_q = \overline{h_q} \mathbf{u}$. Thus, we see that the particle velocity directions $\mathbf{u}_q$ are aligned but may differ in scale.

Affine state. Next, we update the affine state $\mathbf{C}_p^{n+1}$ on particles.

$$\begin{aligned} \mathbf{C}_p^{n+1} &= \xi \sum_i w_{ip}^n \tilde{\mathbf{u}}_i^{n+1} (\mathbf{x}_i^n - \mathbf{x}_p^n)^T \\ &= \xi \sum_i w_{ip}^n \varepsilon g^{n+1} e^{i\mathbf{x}_i^n \cdot \mathbf{z}} \mathbf{u} (\mathbf{x}_i^n - \mathbf{x}_p^n)^T = \varepsilon g^{n+1} e^{i\mathbf{x}_p^n \cdot \mathbf{z}} \mathbf{u} \overline{\mathbf{e}_q}^T \end{aligned}$$

As with velocities, this must match our definition for $\mathbf{C}_p^{n+1} = \varepsilon g^{n+1} \mathbf{C}_q e^{i\mathbf{x}_p^n \cdot \mathbf{z}}$, which leads to $\mathbf{C}_q = \mathbf{u} \overline{\mathbf{e}_q}^T$. With this simplified form for $\mathbf{C}_q$, we may simplify $\mathbf{b}$ as well.

$$\begin{aligned} \mathbf{b} &= \sum_p w_{ip}^n (\mathbf{u}_q + \mathbf{C}_q (\mathbf{x}_i^n - \mathbf{x}_p^n)) e^{i(\mathbf{x}_p^n - \mathbf{x}_i^n) \cdot \mathbf{z}} \\ &= \sum_q (h_q \mathbf{u}_q + \xi^{-1} \mathbf{C}_q \mathbf{e}_q) = \sum_q (h_q \overline{h_q} \mathbf{u} + \xi^{-1} \mathbf{u} \overline{\mathbf{e}_q}^T \mathbf{e}_q) = b\mathbf{u} \end{aligned}$$

Deformation gradient. The final remaining step in the transfers from grid to particles is the deformation gradient.

$$\begin{aligned} \nabla \mathbf{u}_p^{n+1} &= \sum_i \tilde{\mathbf{u}}_i^{n+1} (\nabla w_{ip}^n)^T = \sum_i \varepsilon g^{n+1} e^{i\mathbf{x}_i^n \cdot \mathbf{z}} \mathbf{u} (\nabla w_{ip}^n)^T \\ &= \varepsilon g^{n+1} e^{i\mathbf{x}_p^n \cdot \mathbf{z}} \mathbf{u} \overline{\mathbf{k}_q}^T \\ \mathbf{F}_p^{n+1} &= (\mathbf{I} + \Delta t \nabla \mathbf{u}_p^{n+1}) \mathbf{F}_p^n = (\mathbf{I} + \Delta t \varepsilon g^{n+1} e^{i\mathbf{x}_p^n \cdot \mathbf{z}} \mathbf{u} \overline{\mathbf{k}_q}^T) \mathbf{F}_p^n \end{aligned}$$

Substituting in the definition for $\mathbf{F}_p^n$ and neglecting higher order terms we have

$$\begin{aligned} \mathbf{I} + \varepsilon g^{n+1} \mathbf{F}_q e^{i\mathbf{x}_p^n \cdot \mathbf{z}} &= \left(\mathbf{I} + \Delta t \varepsilon g^{n+1} \mathbf{u} \overline{\mathbf{k}_q}^T e^{i\mathbf{x}_p^n \cdot \mathbf{z}}\right) (\mathbf{I} + \varepsilon g^n \mathbf{F}_q e^{i\mathbf{x}_p^n \cdot \mathbf{z}}) \\ \varepsilon g^{n+1} \mathbf{F}_q e^{i\mathbf{x}_p^n \cdot \mathbf{z}} &= \Delta t \varepsilon g^{n+1} e^{i\mathbf{x}_p^n \cdot \mathbf{z}} \mathbf{u} \overline{\mathbf{k}_q}^T + \varepsilon g^n \mathbf{F}_q e^{i\mathbf{x}_p^n \cdot \mathbf{z}} \\ g \mathbf{F}_q &= \Delta t g \mathbf{u} \overline{\mathbf{k}_q}^T + \mathbf{F}_q \\ \mathbf{F}_q &= \tau \mathbf{u} \overline{\mathbf{k}_q}^T \end{aligned}$$

Since positions are assumed not to move appreciably (assumption (A4)), we have completed the time step. In the process of keeping the equations relatively short, we have defined many intermediate quantities, which we must now eliminate to solve for the amplification factor g.

Eliminating unknown vectors and matrices. When we started, we assumed V^0, m, and Δt were known. The grid and particle locations ($\mathbf{x}_p^n$ and $\mathbf{x}_i^n$) were fixed in a regular arrangement, and the weights are known. We also fixed a wave-number $\mathbf{z}$, which for now we assume is known. This allows us (in principle at least) to directly calculate h_q, $\mathbf{e}_q$, b, $\mathbf{k}_q$, and $\mathbf{M}$. The quantities $\mathbf{b}$ and $\mathbf{F}_q$ are expressed in terms of other quantities. The matrix and vector equations we have left are the definitions of $\mathbf{K}_q$, $\mathbf{c}$, $\mathbf{u}$. We can eliminate $\mathbf{u}$ by introducing a new scalar β and substituting in $\mathbf{b}$.

$$\mathbf{u} = \frac{b\mathbf{u}m - \Delta t V^0 \mathbf{c}}{wmg}$$

$$\mathbf{u} = \frac{\Delta t V^0}{m(b - wg)}\mathbf{c} = \beta \mathbf{c}$$

Next, we eliminate $\mathbf{F}_q$ and $\mathbf{K}_q$ from $\mathbf{c}$, dropping into indexing notation to deal with tensors.

$$\begin{aligned}
\mathbf{c} &= \sum_p \mathbf{K}_q \nabla w_{ip}^n e^{i(\mathbf{x}_p^n - \mathbf{x}_i^n)\cdot \mathbf{z}} \\
&= \sum_p (\mathbf{M} : (\tau \mathbf{u} \overline{\mathbf{k}_q}^T)) \nabla w_{ip}^n e^{i(\mathbf{x}_p^n - \mathbf{x}_i^n)\cdot \mathbf{z}} \\
c_i &= \tau \sum_p M_{ijkl} u_k \overline{k_{ql}} (\nabla w_{ip}^n)_j e^{i(\mathbf{x}_p^n - \mathbf{x}_i^n)\cdot \mathbf{z}} \\
c_i &= \tau M_{ijkl} u_k \sum_p \overline{k_{ql}} (\nabla w_{ip}^n)_j e^{i(\mathbf{x}_p^n - \mathbf{x}_i^n)\cdot \mathbf{z}} \\
c_i &= \tau M_{ijkl} u_k B_{jl} = \tau A_{ik} u_k \\
\mathbf{c} &= \tau \mathbf{A}\mathbf{u}
\end{aligned}$$

We can now eliminate $\mathbf{c}$ and $\mathbf{d}$ from the formula for $\mathbf{u}$ giving

$$\mathbf{u} = \beta \mathbf{c} = \beta \tau \mathbf{A}\mathbf{u}.$$

We are left with the eigenvalue problem $\mathbf{A}\mathbf{u} = \sigma \mathbf{u}$, where $\mathbf{u}$ is an eigenvector with eigenvalue σ. Note that all of the pieces for $\mathbf{A}$ can be computed, so that $\mathbf{A}$, σ, and $\mathbf{u}$ are now available. We also have $\sigma\beta\tau = 1$.

Eliminating the scalars. We have now computed all of the vector and matrix quantities. We are however left with the intermediate scalars τ and β, which we must eliminate to get g, which is ultimately what tells us whether we are stable. σ was computed by solving an eigenvalue problem and is assumed to be available. The remaining equations are

$$\tau = \frac{\Delta t g}{g-1} \qquad \beta = \frac{\Delta t V^0}{m(b - wg)} \qquad \sigma\beta\tau = 1.$$

Eliminating τ and β we get a quadratic for g,

$$g^2 + \left(\frac{\Delta t^2 V^0 \sigma}{mw} - \frac{b}{w} - 1\right) g + \frac{b}{w} = g^2 + 2rg + s = 0.$$

What remains is to examine the roots g.

scheme	spline	Analytic f	Sim 2D f	Sim 3D f
APIC	quad	1.0000	1.0007	1.0017
CPIC	quad	1.0000	1.0011	1.0029
PIC	quad	0.7071	0.7133	0.7182
APIC	cubic	1.7042	1.7055	1.7072
CPIC	cubic	1.3952	1.3993	1.4028
PIC	cubic	1.4033	1.4055	1.4130

Figure 4: *The Von Neumann time step restriction is $\Delta t \le f\Delta x/c$, where c is the sound speed. This table shows the value of the constant f for each version of the scheme. The analytical value was computed exactly (see the technical document for exact values). The two simulated columns were obtained by running a periodic simulation with one particle per cell, using binary search to find the largest stable CFL number. f is the same in 2D and 3D.*

Real or complex. The quantities w, b, ξ, $\mathbf{M}$, s, and c are always real. Under the symmetry assumption (A5), for each grid node i and each particle p there is a particle $\overline{p}$ such that $(\mathbf{x}_{\overline{p}}^n - \mathbf{x}_i^n) = -(\mathbf{x}_p^n - \mathbf{x}_i^n)$. Note that $w_{ip}^n = w_{i\overline{p}}^n$ but $\nabla w_{ip}^n = -\nabla w_{i\overline{p}}^n$. Then,

$$h_{\overline{q}} = \sum_i w_{i\overline{p}}^n e^{i(\mathbf{x}_{\overline{p}}^n - \mathbf{x}_i^n)\cdot \mathbf{z}} = \sum_i w_{ip}^n e^{-i(\mathbf{x}_p^n - \mathbf{x}_i^n)\cdot \mathbf{z}} = \overline{h_q}.$$

Similarly, we also have $\mathbf{k}_{\overline{q}} = -\overline{\mathbf{k}_q}$, and $\mathbf{e}_{\overline{q}} = -\overline{\mathbf{e}_q}$. From this we conclude that $\mathbf{B}$ is real and thus symmetric. Then $\mathbf{A}$ is also real and symmetric, so that its eigenvalues σ and eigenvectors $\mathbf{u}$ are real. Note that $\mathbf{A}$ is symmetric so that σ is real. Since the equilibrium configuration should be stable (a consequence of assumption (A4)), $\sigma > 0$. From this, r is also real.

Stability. If $r^2 < s$ then g is complex and the roots are a complex conjugate pair whose product is $g\overline{g} = |g|^2 = s$. One can show that $s \le 1$, so that the complex case is always stable. We conclude that the limit of stability (where $|g| = 1$) is reached when $g = \pm 1$. The case $g = 1$ leads to $c = 0$ or $\Delta t = 0$, which only requires a nonnegative time step. The case $g = -1$ yields our time step restriction

$$c \le \frac{2(s+1)}{\sigma} \qquad \Delta t \le \sqrt{\frac{mwc}{V^0}} = \sqrt{\frac{mw}{V^0}}\sqrt{\frac{2(s+1)}{\sigma}},$$

which must be true for all $\mathbf{z}$. The dependence on $\mathbf{z}$ is entirely within the second factor, where both s and σ depend on $\mathbf{z}$.

Isotropic. For an isotropic constitutive model near the rest configuration, $M_{ijkl} = \lambda\delta_{ij}\delta_{kl} + \mu\delta_{ik}\delta_{jl} + \mu\delta_{il}\delta_{jk}$. Then, $\mathbf{A} = (\lambda + \mu)\mathbf{B} + \mu \operatorname{tr}(\mathbf{B})\mathbf{I}$ has the same eigenvectors as $\mathbf{B}$. Since the trace is the sum of the eigenvalues, the eigenvalues of $\mathbf{A}$ are bounded by $(\lambda + 2\mu)\operatorname{tr}(\mathbf{B})$ with equality in the rank-one case. This leads to the bound

$$\Delta t \le \Delta x \sqrt{\frac{m/V^0}{\lambda + 2\mu}} \sqrt{\frac{2(s+1)}{\Delta x^2 \operatorname{tr}(\mathbf{B})/w}} = \frac{f\Delta x}{c}. \qquad (1)$$

The first factor is independent of $\mathbf{z}$, the number of particles, or their placement. It is in fact just inverse of the sound speed c. The second term f is dependent only on $\mathbf{z}$, the splines, and the particle placement.

$$f^2 = \frac{2(s+1)}{\Delta x^2 \operatorname{tr}(\mathbf{B})/w} = \frac{2\sum_q(|h_q|^2+\xi^{-1}\|\mathbf{e}_q\|^2+1)}{\Delta x^2\sum_p\|\mathbf{k}_q\|^2} \le \max_q \frac{2(|h_q|^2+\xi^{-1}\|\mathbf{e}_q\|^2+1)}{\Delta x^2\|\mathbf{k}_q\|^2},$$

where we have used the fact that if $a,b,c,d>0$ then $\frac{a}{c} \le \frac{a+b}{c+d} \le \frac{b}{d}$ or $\frac{b}{d} \le \frac{a+b}{c+d} \le \frac{a}{c}$. We conclude that the smallest possible value of f, and thus the tightest time step restriction, is obtained with a single particle. The quantity f with a single particle depends only on the two vectors $\mathbf{x}_p^n$ and $\mathbf{z}$, which due to periodicity are both bounded. f can therefore be bounded numerically or analytically.

Bounds. The constants f depend on the underlying scheme and are given in Figure 4 under the "Analytic f" column. The exact analytic expressions for these bounds are given in the technical document. Although the constant f depends on the scheme (PIC, APIC, CPIC) and the splines (quadratic, cubic), it does not depend on dimension. We were surprised that Von Neumann analysis predicted that PIC is actually *less stable* than APIC, but the numerical experiments bear this out. It also suggests that CFL numbers larger than 1 should be okay for cubic splines.

Numerical validation. It turns out that in all cases, this bound is realized with one particle placed in the center of each cell. This makes the bounds easy to verify numerically. To validate the correctness of the conclusions from the analysis, we ran MPM simulations with one particle in the middle of each cell with periodic boundaries on a 32^2 or 32^3 grid. We initialized the state ($\mathbf{v}_p^n$, $\mathbf{C}_p^n$, $\mathbf{F}_p^n$) by adding a random perturbation in the range $[-10^{-4}, 10^{-4}]$ to the rest configuration. We used Young's modulus $10^3\,Pa$ and Poisson's ratio 0.3 with a Neo Hookean constitutive model and simulated to time $10s$. We deemed a simulation to be unstable if the maximum velocity magnitude grew by a factor of 10. We used binary search to identify the largest stable CFL number for each sim in both 2D and 3D. These results are shown in the last two columns of Figure 4 and are in very close agreement with the analytically derived bounds. The numerical bounds are in fact always slightly higher than the true bound; binary search is actually searching for the largest CFL number for which the simulation is not unstable enough to fail the stability criterion over the bounded duration of the simulation.

The time step restriction based on Von Neumann stability analysis is not a heuristic in the sense that there exists a particle layout for which this time step is in fact required for stability. What is perhaps unusual about Von Neumann analysis is that it constructs a simulation where that bound is required. Of course, this is a worst case bound under the assumptions of the analysis. With more favorable particle coverage, the same simulation setup is stable at a larger time step size. As we will explore in more detail later, in the presence of boundaries, the stable time step size can be worse.

Generalizations We performed our analysis using the five rather restrictive assumptions A1-A5. It is natural to ask whether the analysis can be generalized. Assumptions (A1) and (A2) are essential to Von Neumann analysis, since otherwise the Fourier transform will not diagonalize the PDE. Assumption (A3) simplifies the analysis, but relaxing it could be useful for analyzing mixed fluid phases. Relaxing assumption (A4) by linearizing about a non-rest equilibrium configuration could be very interesting. This straightforward generalization amounts to selecting a time step size such that the equilibrium is *dynamically stable*, which is problematic since the equilibrium may be physically unstable, such as when buckling or tearing. Assumption (A5) should be removable. It causes $\mathbf{A}$ to be symmetrical and have the same eigenvectors as $\mathbf{B}$, which facilitates the derivation of (1). The assumption is not needed elsewhere.

3.2. Single particle instability

It is possible for a simulation that satisfies its CFL restriction to be unstable. One particularly simple case where this is observed is for a single particle. This case occurs in practice when a particle becomes isolated, such as when water splashes or sand spreads out. In [SSS20], an effective time step restriction for the single particle instability was proposed in the case of fluids. In that paper, no solution was proposed in the more difficult case of solids. In this section, we derive a time step restriction for solids in the vicinity of the rest configuration ($\mathbf{F}_p^n \approx \mathbf{I}$). In this context, this assumption is not as limiting as it might seem. Particles that are at risk of becoming isolated tend to be near the surface, where they would not be expected to experience significant strain.

Our strategy for deriving a time step restriction is to consider the stability of perturbations from the rest configuration. The momentum of an isolated particle is constant in the absence of outside forces, so that its velocity does not change. Ignoring grid-related effects, the velocity of the particle is decoupled from the rest of the dynamics. In practice, the location of a particle relative to the grid can affect its stability, with some placements within a cell requiring a smaller time step for stability. A particle traveling through the grid would experience a average of the local stability along its trajectory. To handle this, we evaluate the stability of a stationary particle at an arbitrary location within a cell and then select a time step size that is stable for all such locations.

Stability formulation. At the beginning of each time step, the isolated particle has state m_p, $\mathbf{x}_p^n$, $\mathbf{v}_p^n$, $\mathbf{C}_p^n$, and $\mathbf{F}_p^n$. The mass m_p does not change. Since we are assuming that the particle is stationary, $\mathbf{v}_p^n = \mathbf{0}$ and $\mathbf{x}_p^n$ is constant. We assume a small perturbation from the rest configuration, so that $\mathbf{C}_p^n = \varepsilon\mathbf{A}^n$ and $\mathbf{F}_p^n = \mathbf{I} + \varepsilon\mathbf{E}^n$. After the time step, we will have a new state $\mathbf{C}_p^{n+1} = \varepsilon\mathbf{A}^{n+1}$ and $\mathbf{F}_p^{n+1} = \mathbf{I} + \varepsilon\mathbf{E}^{n+1}$. The new state variables ($\mathbf{E}^{n+1}, \mathbf{F}_p^{n+1}$) are related to the originals ($\mathbf{E}^n, \mathbf{F}_p^n$) by a matrix $\mathbf{N}$. Since the changing portion of the state consists of two 3×3 matrices, $\mathbf{N}$ will be an 18×18 matrix. A stable time step size is one such that the spectral radius (largest eigenvalue magnitude) of $\mathbf{N}$ is no larger than 1.

P2G and grid update. We begin by transferring the particle's state to the grid, which yields

$$m_i^n = m_p w_{ip}^n \qquad \mathbf{u}_i^n = \mathbf{C}_p^n(\mathbf{x}_i^n - \mathbf{x}_p^n) = \varepsilon\mathbf{A}^n(\mathbf{x}_i^n - \mathbf{x}_p^n).$$

Next, we compute particle forces. Let $\mathbf{M} = \frac{\partial \mathbf{P}}{\partial \mathbf{F}}(\mathbf{F}_p^n)$ and $\mathbf{K} = \mathbf{M} : \mathbf{E}^n$ so that $\mathbf{P}(\mathbf{F}_p^n) = \varepsilon \mathbf{M} : \mathbf{E}^n = \varepsilon \mathbf{K}$. Neglecting higher order terms,

$$\mathbf{f}_i^n = -\sum_p V_p^0 (\mathbf{P}(\mathbf{F}_p^n))(\mathbf{F}_p^n)^T \nabla w_{ip}^n = -\varepsilon V_p^0 \mathbf{K} \nabla w_{ip}^n.$$

Finally, we can compute the new grid velocities

$$\tilde{\mathbf{u}}_i^{n+1} = \mathbf{u}_i^n + \Delta t (m_i^n)^{-1} \mathbf{f}_i^n = \varepsilon \mathbf{A}^n (\mathbf{x}_i^n - \mathbf{x}_p^n) - \frac{\varepsilon \Delta t V_p^0}{w_{ip}^n m_p} \mathbf{K} \nabla w_{ip}^n.$$

G2P transfers. A quick calculation shows that $\mathbf{u}_p^{n+1} = \sum_i w_{ip}^n \tilde{\mathbf{u}}_i^{n+1} = \mathbf{0}$, so that particles do not move. Let

$$\mathbf{R} = \frac{1}{\xi} \sum_i \frac{\nabla w_{ip}^n (\nabla w_{ip}^n)^T}{w_{ip}^n}.$$

We now update the deformation gradient on particles.

$$\begin{aligned}
\nabla \mathbf{u}_p^{n+1} &= \sum_i \tilde{\mathbf{u}}_i^{n+1} (\nabla w_{ip}^n)^T \\
&= \varepsilon \mathbf{A}^n \sum_i (\mathbf{x}_i^n - \mathbf{x}_p^n)(\nabla w_{ip}^n)^T - \frac{\varepsilon \Delta t V_p^0}{m_p} \mathbf{K} \sum_i \frac{\nabla w_{ip}^n (\nabla w_{ip}^n)^T}{w_{ip}^n} \\
&= \varepsilon \mathbf{A}^n - \frac{\varepsilon \xi \Delta t V_p^0}{m_p} \mathbf{K} \mathbf{R} \\
\mathbf{F}_p^{n+1} &= (\mathbf{I} + \Delta t \nabla \mathbf{u}_p^{n+1}) \mathbf{F}_p^n \\
&= \left(\mathbf{I} + \varepsilon \Delta t \mathbf{A}^n - \frac{\varepsilon \xi \Delta t^2 V_p^0}{m_p} \mathbf{K} \mathbf{R} \right) (\mathbf{I} + \varepsilon \mathbf{E}^n) \\
&= \mathbf{I} + \varepsilon \Delta t \mathbf{A}^n - \frac{\varepsilon \xi \Delta t^2 V_p^0}{m_p} \mathbf{K} \mathbf{R} + \varepsilon \mathbf{E}^n \\
\mathbf{E}^{n+1} &= \Delta t \mathbf{A}^n - \frac{\xi \Delta t^2 V_p^0}{m_p} (\mathbf{M} : \mathbf{E}^n) \mathbf{R} + \mathbf{E}^n
\end{aligned}$$

Finally we update the affine state.

$$\begin{aligned}
\mathbf{C}_p^{n+1} &= \xi \sum_i w_{ip}^n \tilde{\mathbf{u}}_i^{n+1} (\mathbf{x}_i^n - \mathbf{x}_p^n)^T \\
&= \xi \varepsilon \mathbf{A}^n \sum_i w_{ip}^n (\mathbf{x}_i^n - \mathbf{x}_p^n)(\mathbf{x}_i^n - \mathbf{x}_p^n)^T \\
&\quad - \frac{\varepsilon \xi \Delta t V_p^0}{m_p} \mathbf{K} \sum_i \nabla w_{ip}^n (\mathbf{x}_i^n - \mathbf{x}_p^n)^T = \varepsilon \mathbf{A}^n - \frac{\varepsilon \xi \Delta t V_p^0}{m_p} \mathbf{M} : \mathbf{E}^n \\
\mathbf{A}^{n+1} &= \mathbf{A}^n - \frac{\xi \Delta t V_p^0}{m_p} \mathbf{M} : \mathbf{E}^n
\end{aligned}$$

Let $\hat{\mathbf{M}} = \frac{\xi \Delta t^2 V_p^0}{m_p} \mathbf{M}$. Then, we can express these as a matrix equation using indexing notation

$$\begin{pmatrix} \Delta t A_{ij}^{n+1} \\ E_{ij}^{n+1} \end{pmatrix} = \underbrace{\begin{pmatrix} \delta_{ik}\delta_{jl} & -\hat{M}_{ijkl} \\ \delta_{ik}\delta_{jl} & \delta_{ik}\delta_{jl} - \hat{M}_{imkl} R_{mj} \end{pmatrix}}_{\mathbf{N}} \begin{pmatrix} \Delta t A_{kl}^n \\ E_{kl}^n \end{pmatrix}$$

We define $\mathbf{N}$ to be the resulting 18×18 matrix (8×8 in 2D).

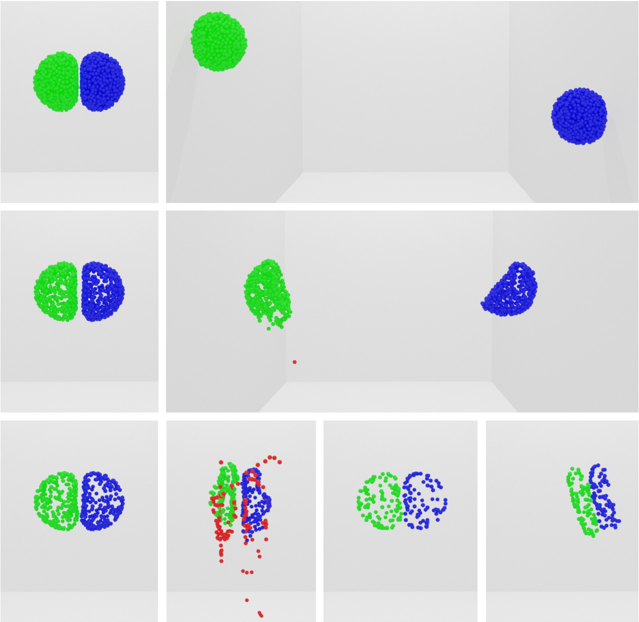

Figure 5: *(Top) Colliding two filled balls is stable. (Middle) Removing the core of the ball reduces the number of particle neighbors, causing the sim to become unstable and eject a particle. Lowering the time step stabilizes the sim. (Bottom left) Thinning out the object further causes it to become unstable again. (Bottom right) Enforcing the single particle time step restriction stabilizes the simulation, even down to isolated particles.*

Simplifying the system. The matrix $\mathbf{N}$ is defined in terms of two unitless quantities $\hat{\mathbf{M}}$ and $\mathbf{R}$. $\hat{\mathbf{M}}$ depends on forces and time step size, while $\mathbf{R}$ encodes the dependence on particle position. In the case of CPIC, $\mathbf{R} = \mathbf{I}$, so that particle position does not matter. In the case of APIC, $\mathbf{R}$ is diagonal. Let $a = \frac{x}{\Delta x}$ so that

$$\mathbf{R}_{11} = \frac{3}{3 - 4a^2} \qquad -\frac{1}{2} \leq a \leq \frac{1}{2} (quad)$$

$$\mathbf{R}_{11} = \frac{-2(9a^4 - 18a^3 + 2a^2 + 7a + 2)}{(3a^3 - 6a^2 + 4)(3a^3 - 3a^2 - 3a - 1)} \qquad 0 \leq a \leq 1 (cubic)$$

The other entries $\mathbf{R}_{22}$ and $\mathbf{R}_{33}$ are similar. Note that the entries are decoupled. That is, $\mathbf{R}_{11}$ depends on x but not y or z. Let $r_i = R_{ii}$ be the diagonal entries. For CPIC, $r_i = 1$. For APIC with quadratic splines, $1 \leq r_i \leq \frac{3}{2}$. For APIC with cubic splines, $1 \leq r_i \leq 1.0456$. We denote the relevant upper bound by r so that $1 \leq r_i \leq r$.

Assuming an isotropic constitutive model, we have $M_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu \delta_{ik} \delta_{jl} + \mu \delta_{il} \delta_{jk}$. We define new unitless scalars p and s so that $s = \frac{\xi \Delta t^2 V_p^0 \mu}{m_p}$ and $\lambda = p\mu$. The scalar $p = \frac{2\nu}{1-2\nu}$ is a function of the Poisson's ratio; it is an unknown positive constant for our purposes. $\mathbf{N}$ now depends on only s, p, r_i.

Stability as a polynomial problem. The characteristic polynomial $P(\lambda; s, p, r_i)$ of $\mathbf{N}$ (ignoring the denominator) is

$$\begin{aligned}
P(\lambda; s, p, r_i) &= (1-\lambda)^6 Q(\lambda; s, p, r_i) \\
&\qquad S(\lambda; s, r_1, r_2) S(\lambda; s, r_2, r_3) S(\lambda; s, r_1, r_3) \\
S(\lambda; s, r_i, r_j) &= \lambda^2 + (r_i s + r_j s - 2)\lambda - s r_i - s r_j + 2s + 1,
\end{aligned}$$

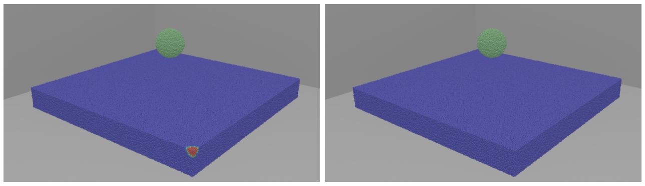

Figure 6: *(Left) A ball drops on a block sitting on the ground. The corners have fewer neighboring particles, which causes them to become unstable. Blue particles are stable, and red particles indicate instability. A particle is later ejected from the corner (not shown). (Right) Enforcing the single particle stability criterion stabilizes the simulation, including the corners.*

where $Q(\lambda; s, p, r_i)$ is a polynomial of degree 6 in λ (the polynomial is omitted here for brevity but is provided in the technical document). The goal is to find

$$s^* = \min_{1 \le r_i \le r} s \qquad \text{such that} \qquad P(\lambda; s, p, r_i) = 0, |\lambda| = 1, s > 0.$$

The time step restriction is then readily deduced from s^*. Note that the factors of $P(\lambda; s, p, r_i)$ can be treated separately. The $(1-\lambda)^6$ factor does not impose a time step restriction and can be discarded. There are now two high level cases to consider: $S(\lambda; s, r_i, r_j) = 0$ or $Q(\lambda; s, p, r_i) = 0$.

Case $S(\lambda; s, r_i, r_j) = 0$. We break this case into three further cases: $\lambda = \pm 1$ and complex conjugate λ. We first consider the case of complex conjugate solutions. Note that $(\lambda - \kappa)(\lambda - \overline{\kappa}) = \lambda^2 - (\kappa + \overline{\kappa})\lambda + \kappa\overline{\kappa}$, so that the square of the magnitude of the eigenvalues are just the constant term of the quadratic. That is, we need $1 = |\lambda|^2 = -sr_i - sr_j + 2s + 1$ which leads to $s(2 - r_i - r_j) = 0$. This case does not lead to a useful time step restriction ($s = 0$ implies $\Delta t = 0$). The case $\lambda = 1$ leads to $0 = S(1; s, r_1, r_2) = 2s$, which also does not lead to a useful solution. The case $\lambda = -1$ implies $0 = S(-1; s, r_1, r_2) = 2(s + 2 - r_1 s - r_2 s)$, which leads to a meaningful time step restriction $s^* \le A$ where

$$s^* \le \frac{2}{r_1 + r_2 - 1} \le \frac{2}{2r - 1} = A. \tag{2}$$

Case $Q(\lambda; s, p, r_i) = 0$. Next, we consider the case where $Q(\lambda; s, p, r_i) = 0$. As before, we consider $\lambda = \pm 1$ and complex conjugate λ separately. $0 = Q(1; s, p, r_i)$ has only the solution $s = 0$. The complex conjugate case is very complex and does not lead to the tightest time step restriction, so we omit it here. The case is addressed in the technical document. This leaves the case $Q(-1; s, p, r_i) = 0$, which is a cubic polynomial in s, which could in theory be solved for $s(p, r_1, r_2, r_3)$. This would then need to be minimized over all feasible r_1, r_2, r_3 to produce the bound $s^*(p)$. This approach is infeasible, but there is a simpler way. Note that p is constant and r_i are independent. If r_k is not at a bound, then $0 = \frac{\partial}{\partial r_k} Q(-1; s, p, r_i) = \frac{\partial Q}{\partial s}(-1; s, p, r_i)\frac{\partial s}{\partial r_k} + \frac{\partial Q}{\partial r_k}(-1; s, p, r_i)$. Since $\frac{\partial s}{\partial r_k} = 0$, we have $\frac{\partial Q}{\partial r_k}(-1; s, p, r_i) = 0$. For each r_k, we must choose between $r_k = 1$, $r_k = r$, or $\frac{\partial Q}{\partial r_k}(-1; s, p, r_i) = 0$. This leads to many cases, but we can reduce these somewhat. The r_k are equivalent, so permutations need not be considered.

Partially unconstrained. Consider that at least one r_i is unconstrained (say, r_1). Then, $Q(-1; s, p, r_i) = \frac{\partial Q}{\partial r_1}(-1; s, p, r_i) = 0$. Eliminating r_1, s, p from this system of equations leads to the equation $(2r_3 s - s - 2)^2(2r_2 s - s - 2)^2 = 0$, which implies $2r_3 s - s - 2 = 0$ or $2r_2 s - s - 2 = 0$. Both of these lead back to (2). We are left with the case that all r_i are at their bounds 1 or r.

Boundary cases. There are four boundary cases (depending on how many of the r_i are 1 and r). $0 = Q(-1; s, p, r_1, r_1, r_1) = -4(2r_1 s - s - 2)^2(6pr_1 s - 3ps + 4r_1 s - 2s - 4)$. The middle factor leads to (2). The last factor produces the restriction $s^* \le B$ where

$$s^* \le \frac{4}{(2r_1 - 1)(3p + 2)} \le \frac{4}{(2r - 1)(3p + 2)} = B, \tag{3}$$

which is more strict than (2) since $0 < B < A$. $0 = Q(-1; s, p, r_1, r_2, r_2) = -4(2r_2 s - s - 2)u(s, p, r_1, r_2)$ where $u(s, p, r_1, r_2) = (2r_2 - 1)(2r_1 - 1)(3p + 2)s^2 + (-4pr_1 - 8pr_2 + 6p - 8r_1 - 8r_2 + 8)s + 8$. The factor $(2r_2 s - s - 2)$ leads to (2). The last factor $u(s, p, r_1, r_2)$ is a quadratic in s where $u(B, p, r_1, r_2) > 0$, $u_s(B, p, r_1, r_2) < 0$, and $u_{ss}(B, p, r_1, r_2) > 0$ for all $1 \le r_1, r_2 \le r$ and $p > 0$. We conclude that any roots of $u(B, p, r_1, r_2)$, if they are real, must be larger than B. Thus, $s^* \le B$ from (3) is the time step restriction for the single particle instability. Noting $s = \frac{\xi \Delta t^2 V_p^0 \mu}{m_p}$ we have our final single-particle time step restriction

$$\Delta t \le \sqrt{\frac{m_p / V_p^0}{\xi(r - \frac{k}{2})(\mu + \frac{3}{2}\lambda)}}, \tag{4}$$

where k is introduced to summarize the PIC case ($k = 0$) and APIC/CPIC cases ($k = 1$). The PIC and 2D cases are handled in the technical document but omitted here. Figure 1 shows the single particle stability regions for PIC, APIC, and CPIC.

Alternating time step sizes. Conventional wisdom suggests that a simulation will be stable for any combination of time step sizes provided all of the time step sizes are below some critical value. This assumption is fundamental to the idea of using adaptive time step sizes. At least when simulating stationary isolated particles, *this assumption is not true*. There appear to be sequences of arbitrarily small time step sizes such that long term growth is observed.

We can demonstrate the problem of variable time step sizes by simulating a single stationary particle using CPIC transfers and $\mathbf{A}^n = a^n \mathbf{I}$ and $\mathbf{E}^n = e^n \mathbf{I}$, where a^n and e^n are scalars at time step n. Assuming an isotropic constitutive model, $\mathbf{M} : \mathbf{E}^n = (3\lambda + 2\mu)e^n \mathbf{I}$. Letting $m = \frac{\xi V_p^0}{m_p}(3\lambda + 2\mu)$ and recalling that $\mathbf{R} = \mathbf{I}$ for CPIC we have

$$\mathbf{A}^{n+1} = \mathbf{A}^n - \frac{\xi \Delta t V_p^0}{m_p}\mathbf{M} : \mathbf{E}^n = a^n \mathbf{I} - \Delta t m e^n \mathbf{I}$$

$$\mathbf{E}^{n+1} = \Delta t \mathbf{A}^n - \frac{\xi \Delta t^2 V_p^0}{m_p}(\mathbf{M} : \mathbf{E}^n)\mathbf{R} + \mathbf{E}^n = \Delta t a^n \mathbf{I} - \Delta t^2 m e^n \mathbf{I} + e^n \mathbf{I}$$

We can express this as a matrix

$$\begin{pmatrix} a^{n+1} \\ e^{n+1} \end{pmatrix} = \begin{pmatrix} 1 & -\Delta t m \\ \Delta t & 1 - \Delta t^2 m \end{pmatrix} \begin{pmatrix} a^n \\ e^n \end{pmatrix}. \tag{5}$$

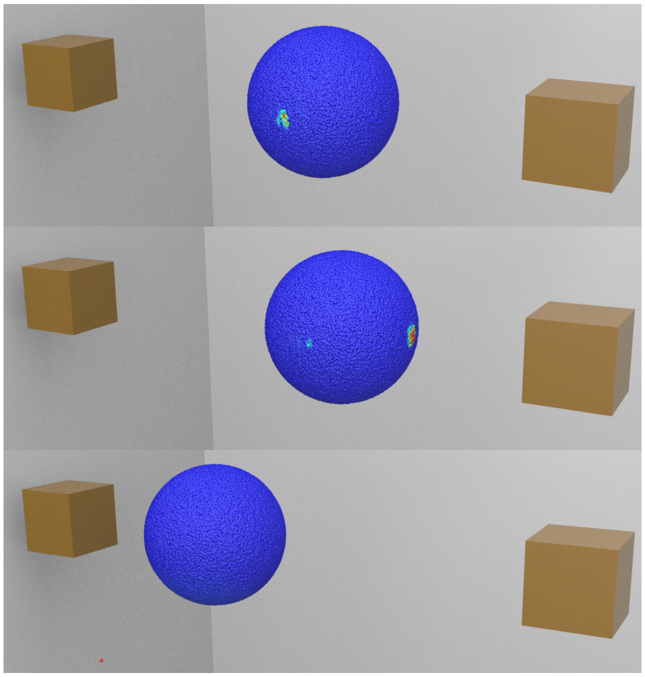

Figure 7: *A ball bounces back and forth between two blocks without the single particle time step restriction. Localized instabilities are observed on the boundary of the sphere (top and middle), and a particle is eventually ejected from the ball (red particle at bottom left). Enforcing the proposed time step restriction stabilizes the simulation.*

If we take two steps with different time steps $\Delta t_0 < \Delta t_1$ then

$$\begin{pmatrix} a^{n+2} \\ e^{n+2} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & -\Delta t_1 m \\ \Delta t_1 & 1-\Delta t_1^2 m \end{pmatrix} \begin{pmatrix} 1 & -\Delta t_0 m \\ \Delta t_0 & 1-\Delta t_0^2 m \end{pmatrix}}_{\mathbf{B}} \begin{pmatrix} a^n \\ e^n \end{pmatrix}. \tag{6}$$

If we alternate between these two time steps sizes, then the eigenvalues λ_0, λ_1 of $\mathbf{B}$ determine the stability. The matrix in (5) has determinant 1, so that $\lambda_0\lambda_1 = \det(\mathbf{B}) = 1$. If the eigenvalues are complex, then $\lambda_0 = \overline{\lambda_1}$ and $|\lambda_0| = |\lambda_1| = 1$, which is stable. If the eigenvalues are real and $\lambda_0 \neq \lambda_1$ then $0 < \lambda_0 < 1 < \lambda_1$ or $0 > \lambda_0 > -1 > \lambda_1$, which is unstable. The simulation is thus stable when $|\lambda_1 + \lambda_2| = |\mathrm{tr}(\mathbf{B})| \leq 2$.

Alternating time step bound. Let $\Delta t_0 = k\Delta t_1$ with $0 < k < 1$. The case $\mathrm{tr}(\mathbf{B}) \leq 2$ leads to $\Delta t_1 \leq \frac{k+1}{k\sqrt{m}}$. This time step restriction is tightest when $k \to 1$ and leads to $\Delta t_1 \to \frac{2}{\sqrt{m}} = T$, where T is the single particle instability time step restriction (4). The case $\mathrm{tr}(\mathbf{B}) \geq -2$ leads to $\Delta t_1^4 k^2 m^2 - \Delta t_1^2 (k+1)^2 m + 4 \geq 0$. This stability requirement fails for all Δt_1 in an interval that includes $\Delta t_1 = \frac{k+1}{k\sqrt{2m}}$. As $k \to 1$, $\Delta t_1 \to \frac{T}{\sqrt{2}}$. Plugging in the limit itself shows it to be stable for any k, so the bound is tight. *Surprisingly, if we simply alternate between two nearly equal time step sizes, we must reduce the time step size by a factor of* $\sqrt{2}$.

Cycled time steps have sharp teeth. Optimizing the trace while obeying the single particle instability criterion shows that the largest magnitude eigenvalue of $\mathbf{B}$ occurs when $3\Delta t_0 = \Delta t_1 = T$. (That is, alternate between taking the single particle time step bound and a time step one-third the size.) With these, $\mathbf{B}$ has eigenvalues $-\frac{1}{3}$ and -3. Every two time steps, the state grows in magnitude by a factor of three. This growth rate is fast enough for a particle that separates from the main bulk to explode before colliding with an obstacle or other particles.

Longer cycles. What happens for longer cycles of time step sizes? The analysis above can in principle be repeated for any number of time step sizes. Cycling 3 nearly equal time step sizes admits unstable time step sizes smaller than (4) by a factor of $2-\varepsilon$. Cycling a sequence of 49 time steps at Δt followed by a step at $0.99\Delta t$ is unstable for $\Delta t = 0.031417T$. For 99+1 time steps, $\Delta t = 0.0157089T$ is unstable. Although this cycling scheme seems contrived, this is exactly what happens when using a fixed time step size that does not evenly divide the frame. *It seems unlikely that the single particle instability can be absolutely avoided with CPIC or APIC with variable time step sizes.* Of course, the situation is not quite as bleak as that; the unstable eigenvalues in these two extreme cases are only -1.00002 and -1.000005, which would not be noticed in practice. Figure 1 shows the effects of non-constant time step sizes on the stability of a single particle.

4. Numerical results

In the analysis above, we derived two time step restrictions. The first was based on Von Neumann analysis, and the second was based on a single particle instability. In this section we probe at the relationships between these time step restrictions and the time step sizes at which actual simulations go unstable. In particular, *both time step restrictions are sharp in the sense that there are simulations that go unstable at that time step size.*

A few observations about the two restrictions are worth pointing out. The first difference is that the Von Neumann stability prediction is related to the classical CFL restriction and depends on the P-wave modulus $(\lambda + 2\mu)$. The single particle time step (4), however, depends on the bulk modulus $(\lambda + \frac{2}{3}\mu)$. (For completeness, (2) depends on the shear modulus μ, which also happens to be equal to the S-wave modulus). That means the ratio between the two depends on the Poisson's ratio and is not a simple fixed number (and thus a simple rule such as "use CFL 0.8" is not adequate). In our simulations, we always use the correct sound speed for our CFL (not the linearization derived from the Von Neumann stability analysis). With this, the single particle time step size usually falls between around CFL number 0.45 and 0.95, but it can be larger than 1.

Effects of boundaries. In Figure 2, we show a simple rotating circle of radius 0.3 in 2D. This circle is run with a Neo Hookean constitutive model with Young's modulus 10^3 and Poisson's ratio 0.3. The angular velocity is 0.4. We use APIC transfers and quadratic splines so that the classical CFL agrees with the results of the Von Neumann analysis. We begin by running this simulation with CFL 0.9. While this simulation does not explode when run at CFL 1, the velocities do occasionally twitch. This suggests that the simulation

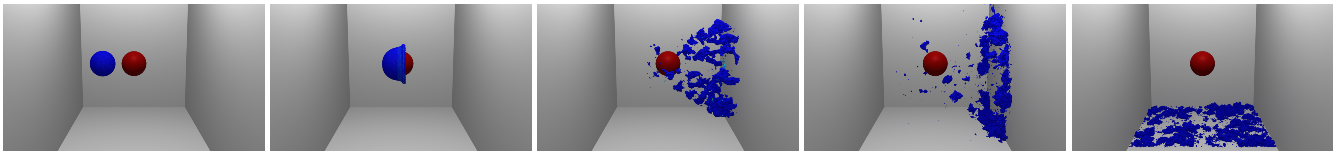

Figure 8: *Sphere collides with an obstacle, fracturing into small pieces. This simulation explodes if the single particle instability time step is not enforced. Enforcing the time step stabilizes the simulation.*

briefly exceeds its stable time step size in certain local areas of the simulation for brief periods of time. The simulation is fully stable at CFL 0.8. What is perhaps a bit unexpected is that if we simply change the geometry by removing all particles except the outer 0.015 of the circle (leaving the other particles in exactly the same places), the simulation explodes. The stable time step size actually depends on the *geometry*, not merely factors such as particle spacing or stiffness. We can "fix" the unstable simulation by reducing the CFL number to 0.7. However, if we prune out more particles (leaving only 0.05 of the circle), the simulation explodes even at this CFL. Reducing the time step size to the single particle time step size (CFL number 0.59 in this case) stabilizes the simulation. If run at a time step 5% larger, the simulation eventually explodes.

This simulation illustrates an important feature of MPM: neighboring particles have a stabilizing influence on MPM simulations. In the absence of boundaries, the simulation above would have been stable at CFL 0.9 and even CFL 1.0. The mere existence of a boundary (where there are fewer neighboring particles) reduced the stable time step size significantly. As the number of neighboring particles is reduced gradually by thinning the geometry, the stable time step size drops. In the limit as more and more particles are removed, we eventually end up with isolated particles, which by definition obey the single particle time step restriction. One may view the single particle time step size as a worst case scenario.

Figure 5 shows a similar experiment with colliding shells. As the shell gets thinner, the simulation requires smaller time step sizes. Generally, 3D simulations appear to be somewhat less susceptible to the effects of boundaries, likely since particles typically have more neighbors.

Instability due to boundaries. Instability when following the CFL is also occasionally observed even under seemingly innocuous conditions. In Figure 7, a sphere bounces between objects without following the proposed time step restriction, leading to instabilities and eventually an ejected particle. Obeying (4) stabilizes the simulation.

Instability due to corners. Corners of objects have even fewer neighboring particles than normal boundary particles, which can cause them to be especially sensitive. In Figure 6, the corner of a block becomes unstable, shown red. A particle is eventually ejected from the corner. Enforcing the single particle stability criterion stabilizes the simulation.

Instability due to fracture. Fracturing causes particles to have fewer neighbors than normal, which may reduce their stability. In Figure 8, a sphere fractures against an obstacle, causing particles to become more separated and correspondingly less stable. This simulation is unstable and explodes if the single particle time step is not followed. If the time step restriction is followed, the simulation behaves normally.

Sand. Figure 9 shows two sand spheres colliding. This simulation uses a standard Drucker Prager sand model. Plasticity can help to stabilize simulations, especially ones that might otherwise be unstable due to the single particle time step restriction. This instability is primarily a feedback loop between pressure and compression. This causes the deformation gradient to cycle between compression and expansion. Drucker Prager plasticity yields on expansion, breaking the feedback loop. This does, however, lead to divergence of the plastic component of the deformation gradient, which is not required for simulation. Because of this, simulations of sand generally do not become unstable on isolated particles.

Stiffer objects. The proposed time step properly accounts for stiffness and works fine for stiff objects, as shown in Figure 10. If run without the proposed restriction, instabilities are observed in the velocities on this simulation. In this case, the instabilities do not affect the appearance of the object, since we are not rendering velocity.

5. Conclusions and Limitations

In this paper, we have analyzed the stability of explicit MPM simulations in two complementary ways. Von Neumann analysis describes the stability of particles is a bulk medium. The single particle instability describes the stability of particles in the limit where particles have no neighbors and serves as a bound on the destabilizing effects of object boundaries. These complementary restrictions seem to work well at stabilizing non-isolated particles. Both analyses fully account for the discretization details of the algorithm, are practical, and are straightforward to implement. Both analyses are based on a linearization about the rest configuration, so the time step sizes may not be as reliable away from the rest configuration. Nevertheless, we have found them to be quite effective in practice.

Isolated particles are a significant problem for explicit MPM. If time steps are fixed, (4) solves the problem. Otherwise, there does not seem to be a practical way to stabilize the simulation using time step restrictions alone. The time step restrictions from [SSS20] can help somewhat, but they may take effect after some damage has already been done. *A pragmatic solution is to simply detect isolated particles and reset (or decay) their state* ($\mathbf{C}_p^n \leftarrow \mathbf{0}, \mathbf{F}_p^n \leftarrow \mathbf{I}$). Alternatively, one may prefer to retain the skew-symmetric part of $\mathbf{C}_p^n$ to conserve angular momentum (similar to an APIC-RPIC blend,

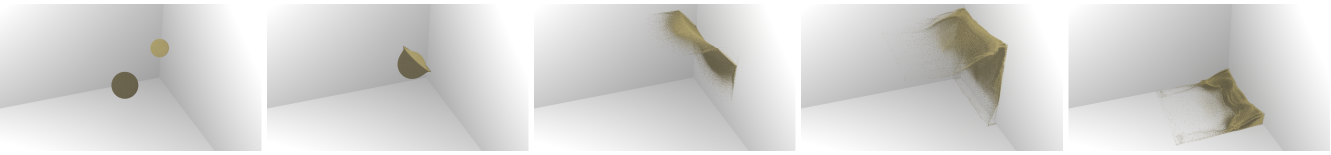

Figure 9: *Two sand spheres collide, scattering sand against a wall. This simulation creates many isolated particles.*

but only for isolated particles). It seems reasonable to assume that isolated particles should not be experiencing significant strain. This does not eliminate the need to follow (4), but it can mitigate the effects of variable time step sizes. Implicit time integration is another option for improving stability.

The time step restrictions in this paper are tight (simulations exist that require them) but do not guarantee stability. In fact, it appears that no guaranteed stable time step size exists for APIC or CPIC that does not depend on the positions of the MPM particles. In particular, an attempt at using full nonlinear stability analysis to construct a guaranteed stable time step size with APIC or CPIC transfers is unlikely to succeed. It is unclear how limiting this is in practice. Bulk MPM simulations do not seem to exhibit such pathological behavior provided particles do not become isolated. Further, isolated particles are easy to identify, so a simple corrective procedure could be employed to stabilize them. In practice, we have found the single particle time step restriction along with the restrictions from [SSS20] to be quite effective.

6. Acknowledgments

The simulations in this paper were rendered using SideFX Houdini. We would like to thank Tamar Shinar for many helpful discussions relating to this paper. This work was funded by NSF - 2006570.

References

[BK04] Bardenhagen S., Kober E.: The generalized interpolation material point method. *Comp Mod in Eng and Sci 5*, 6 (2004), 477–496. 2, 3

[BKR88] Brackbill J., Kothe D., Ruppel H.: FLIP: A low-dissipation, PIC method for fluid flow. *Comp Phys Comm 48* (1988), 25–38. 3

[BL98] Brackbill J., Lapenta G.: Particle-in-cell magnetohydrodynamics. In *16th Int Conf on the Numer Sim of Plasmas* (1998). 2

[BR86] Brackbill J., Ruppel H.: FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. *J Comp Phys 65* (1986), 314–343. 3

[Bra88] Brackbill J.: The ringing instability in particle-in-cell calculations of low-speed flow. *J Comp Phys 75*, 2 (1988), 469–492. 2

[Bra16] Brannon R.: The kinematic anomaly in mpm. https://csmbrannon.net/2016/01/04/the-kinematic-anomaly-in-mpm/, 2016. Accessed: 2020-01-18. 2

[BW98] Baraff D., Witkin A.: Large steps in cloth simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1998), SIGGRAPH '98, ACM, pp. 43–54. 1

[CLR*20] Cao W., Lyu L., Ren X., Zhang B., Yang Z., Wu E.: Fracture patterns design for anisotropic models with the material point method. In *Computer Graphics Forum* (2020), vol. 39, Wiley Online Library, pp. 93–104. 3

[DBD16] Daviet G., Bertails-Descoubes F.: A semi-implicit material point method for the continuum simulation of granular materials. *ACM Trans Graph 35*, 4 (jul 2016). 3

[DS19] Ding O., Schroeder C.: Penalty force for coupling materials with coulomb friction. IEEE. 3

[FGG*17] Fu C., Guo Q., Gast T., Jiang C., Teran J.: A polynomial particle-in-cell method. *ACM Transactions on Graphics (TOG) 36*, 6 (2017), 222. 3

[FHHJ18] Fang Y., Hu Y., Hu S.-M., Jiang C.: A temporally adaptive material point method with regional time stepping. In *Computer graphics forum* (2018), vol. 37, Wiley Online Library, pp. 195–204. 2, 3

[FLGJ19] Fang Y., Li M., Gao M., Jiang C.: Silly rubber: an implicit material point method for simulating non-equilibrated viscoelastic and elastoplastic solids. *ACM Transactions on Graphics (TOG) 38*, 4 (2019), 1–13. 3

[GHF*18] Guo Q., Han X., Fu C., Gast T., Tamstorf R., Teran J.: A material point method for thin shells with frictional contact. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 147. 3

[GPH*18] Gao M., Pradhana A., Han X., Guo Q., Kot G., Sifakis E., Jiang C.: Animating fluid sediment mixture in particle-laden flows. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 149. 3

[Gri14a] Gritton C.: *Ringing Instabilities in Particle Methods*. PhD thesis, The University of Utah, 2014. 2

[Gri14b] Gritton C. E.: *Ringing instabilities in particle methods*. PhD thesis, School of Computing, University of Utah Salt Lake City, UT, 2014. 2

[GSS*15] Gast T., Schroeder C., Stomakhin A., Jiang C., Teran J.: Optimization integrator for large time steps. *IEEE transactions on visualization and computer graphics 21*, 10 (2015), 1103–1115. 3

[GW01] Guilkey J., Weiss J.: An implicit time integration strategy for use with the material point method. In *Proceedings from the First MIT Conference on Computational Fluid and Solid Mechanics* (2001). 2

[GW03] Guilkey J. E., Weiss J. A.: Implicit time integration for the material point method: Quantitative and algorithmic comparisons with the finite element method. *Int J Numer Meth Eng 57*, 9 (2003), 1323–1338. 2

[Har64] Harlow F.: The particle-in-cell method for numerical solution of problems in fluid dynamics. *Meth Comp Phys 3* (1964), 319–343. 2

[HFG*18] Hu Y., Fang Y., Ge Z., Qu Z., Zhu Y., Pradhana A., Jiang C.: A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 150. 2, 3

[HGG*19] Han X., Gast T., Guo Q., Wang S., Jiang C., Teran J.: A hybrid material point method for frictional contact with diverse materials. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 2*, 2 (2019), 1–24. 3

[HN17] Hammerquist C. C., Nairn J. A.: A new method for material point method particle updates that reduces noise and enhances stability.

Computer Methods in Applied Mechanics and Engineering 318 (2017), 724–738. 3

[JGT17] Jiang C., Gast T., Teran J.: Anisotropic elastoplasticity for cloth, knit and hair frictional contact. *ACM Transactions on Graphics (SIGGRAPH 2017) 36*, 4 (2017), 152. 3

[JSS*15] Jiang C., Schroeder C., Selle A., Teran J., Stomakhin A.: The affine particle-in-cell method. *ACM Trans Graph 34*, 4 (2015), 51:1–51:10. 2, 3

[JST17] Jiang C., Schroeder C., Teran J.: An angular momentum conserving affine-particle-in-cell method. *J Comp Phys 338* (2017), 137–164. 3

[JYM*20] Joshuah W., Yunuo C., Minchen L., Yu F., Ziyin Q., Jiecong L., Meggie C., Chenfanfu J.: Anisompm: Animating anisotropic damage mechanics. *ACM Transactions on Graphics (TOG) 39*, 4 (2020), 37–1. 3

[KGP*16] Klár G., Gast T., Pradhana A., Fu C., Schroeder C., Jiang C., Teran T.: Drucker-prager elastoplasticity for sand animation. *ACM Transactions on Graphics (SIGGRAPH 2016).* (2016). 1, 2, 3

[Li22] Li X. L. M. J. C.: Energetically consistent inelasticity for optimization time integration. *ACM Transactions on Graphics (TOG)* (2022). 3

[LLJ22] Li X., Li M., Jiang C.: Energetically consistent inelasticity for optimization time integration. *ACM TOG 41*, 4 (2022). 1

[LS06] Love E., Sulsky D.: An unconditionally stable, energy-momentum consistent implementation of the the material point method. *Comp Meth App Mech Eng 195* (2006), 3903–3925. 2

[NG15] Nairn J. A., Guilkey J. E.: Axisymmetric form of the generalized interpolation material point method. *International Journal for Numerical Methods in Engineering 101*, 2 (2015), 127–147. 3

[NZ20] Ni R., Zhang X.: A precise critical time step formula for the explicit material point method. *International Journal for Numerical Methods in Engineering 121*, 22 (2020), 4989–5016. 2

[Oku72] Okuda H.: Nonphysical noises and instabilities in plasma simulation due to a spatial grid. *J Comp Phys 10*, 3 (1972), 475–486. 2

[RGJ*15] Ram D., Gast T., Jiang C., Schroeder C., Stomakhin A., Teran J., Kavehpour P.: A material point method for viscoelastic fluids, foams and sponges. In *Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2015), ACM, pp. 157–163. 3

[SCS94] Sulsky D., Chen Z., Schreyer H.: A particle method for history-dependent materials. *Comp Meth in App Mech Eng 118*, 1 (1994), 179–196. 2, 3

[SK04] Sulsky D., Kaul A.: Implicit dynamics in the material-point method. *Comp Meth in App Mech Eng 193*, 12 (2004), 1137–1170. 2

[SKB08] Steffen M., Kirby R., Berzins M.: Analysis and reduction of quadrature errors in the material point method (MPM). *Int J Numer Meth Eng 76*, 6 (2008), 922–948. 2

[SSC*13] Stomakhin A., Schroeder C., Chai L., Teran J., Selle A.: A material point method for snow simulation. In *ACM Transactions on Graphics (SIGGRAPH 2013)* (2013), pp. 102:1–10. 1, 3

[SSJ*14] Stomakhin A., Schroeder C., Jiang C., Chai L., Teran J., Selle A.: Augmented mpm for phase-change and varied materials. *ACM Transactions on Graphics (SIGGRAPH 2014)* (2014), 1–11. 3

[SSS20] Sun Y., Shinar T., Schroeder C.: Effective time step restrictions for explicit mpm simulation. In *Computer Graphics Forum* (2020), vol. 39, Wiley Online Library, pp. 55–67. 2, 3, 6, 10, 11

[Str04] Strikwerda J. C.: *Finite difference schemes and partial differential equations*. SIAM, 2004. 2

[SZS95] Sulsky D., Zhou S., Schreyer H.: Application of a particle-in-cell method to solid mechanics. *Comp Phys Comm 87*, 1 (1995), 236–252. 2

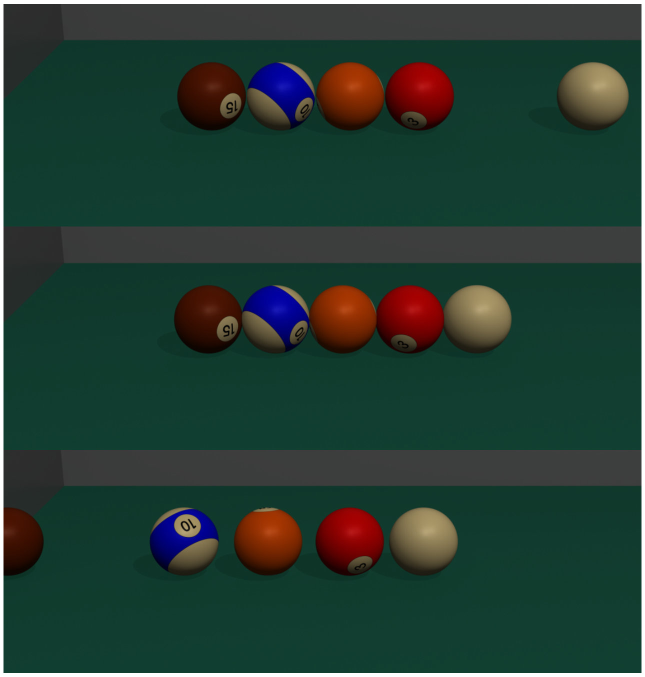

Figure 10: *The time step restriction properly accounts for stiffness and works well for stiff objects.*

[TGK*17] Tampubolon A., Gast T., Klár G., Fu C., Teran J., Jiang C., Museth K.: Multi-species simulation of porous sand and water mixtures. *ACM Transactions on Graphics (SIGGRAPH 2017) 36*, 4 (2017), 105. 1, 3

[Wal09] Wallstedt P.: *On the order of accuracy of the generalized interpolation material point method*. 2009. 3

[WDG*19] Wang S., Ding M., Gast T. F., Zhu L., Gagniere S., Jiang C., Teran J. M.: Simulation and visualization of ductile fracture with the material point method. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 2, 2 (2019), 1–20. 3

[WFL*19] Wolper J., Fang Y., Li M., Lu J., Gao M., Jiang C.: Cd-mpm: continuum damage material point methods for dynamic fracture animation. *ACM Transactions on Graphics (TOG) 38*, 4 (2019), 1–15. 3

[WG08] Wallstedt P., Guilkey J.: An evaluation of explicit time integration schemes for use with the generalized interpolation material point method. *J Comp Phys 227*, 22 (2008), 9628–9642. 2, 3

[WLF*20] Wang X., Li M., Fang Y., Zhang X., Gao M., Tang M., Kaufman D. M., Jiang C.: Hierarchical optimization time integration for cfl-rate mpm stepping. *ACM Transactions on Graphics (TOG) 39*, 3 (2020), 1–16. 3

[YSB*15] Yue Y., Smith B., Batty C., Zheng C., Grinspun E.: Continuum foam: a material point method for shear-dependent flows. *ACM Trans Graph 34*, 5 (2015), 160:1–160:20. 1, 2, 3

[YSC*18] Yue Y., Smith B., Chen P., Chantharayukhonthorn M., Kamrin K., Grinspun E.: Hybrid grains: Adaptive coupling of discrete and continuum simulations of granular media. In *SIGGRAPH Asia 2018 Technical Papers* (2018), ACM, p. 283. 3