A MOMENTUM PRESERVING FRICTIONAL CONTACT ALGORITHM BASED ON AFFINE PARTICLE-IN-CELL GRID TRANSFERS

Michael Tupek\textsuperscript{a}, Jacob Koester\textsuperscript{a}, and Matthew Mosby\textsuperscript{a}

\textsuperscript{a}Sandia National Laboratories

August 6, 2021

\textbf{ABSTRACT}

An efficient and momentum conserving algorithm for enforcing contact between solid bodies is proposed. Previous advances in the material point method (MPM) led to a fast and simple, but potentially momentum violating, strategy for enforcing contact. This was achieved through a combination of velocity transfers between background and foreground grids, and a background grid velocity field update. We propose a modified strategy which ensures conservation of both linear and angular momentum with a novel use of the affine particle-in-cell (APIC) method. Two issues common to particle-in-cell based algorithms for contact are also addressed: material bodies tend to stick at a gap which is proportional to the grid spacing; and material points tend to stick together permanently when located within the same grid cell, making material rebound and friction challenging. We show that the use of APIC, combined with a grid transfer and momentum update algorithm results in contact being enforced at essentially zero gap. For the second issue, we propose a novel iterative scheme which allows particles interacting through the background grid to naturally separate after contact and enforce friction, while still satisfying momentum conservation.

\textbf{Keywords} Particle in Cell · Material Point Method · APIC · Contact Mechanics · Friction

1 Introduction

Simulating contact between solids remains as one of the more challenging aspects of finite deformation continuum mechanics simulations. Here we propose a particle-in-cell based algorithm to enforce contact which is notable for its computational efficiency and simplicity. The affine particle-in-cell (APIC) method is leveraged and applied in a novel way to efficiently enforce contact between two discretized solid bodies, while still allowing for material rebound and frictional sliding [16, 17]. Global conservation of both linear and angular momentum is ensured at every step of the process. Our approach is inspired by and related to a recently proposed contact algorithm for enforcing contact between finite element meshes using a material point method [11].

The material point method (MPM) was first proposed in [29] and [30], with the first strategy for contact in this setting introduced shortly thereafter in [1]. Various improvements to MPM contact can also be found in [24, 14]. Several authors have developed strategies to ensure conservation of linear and angular momentum in the MPM method, but focused on the partial differential equation discretization in the absence of contact [21, 17, 9]. For the method proposed here, MPM grid transfers are solely used to enforce contact, while the PDE discretization is based on traditional Lagrangian finite elements. Linear tetrahedral elements are used, but the method works independent of the underlying weak form discretization. A strategy for introducing discontinuities into the MPM was recently suggested in [22] and shows advantages for sliding and separating contact. More broadly, material-point-like methods that use background B-spline grids are seeing increased use for a variety of mechanics, including shock, fracture, and nearly incompressible solids [2, 3, 23].

Looking beyond MPM methods, numerous strategies for enforcing contact have been proposed. Of some relevance here is the work of [27], where the source of angular momentum loss in Mortar contact methods is identified. Mortar contact algorithms are generally held to be the most accurate for contact, but suffer from being somewhat
challenging to implement, and potentially expensive to solve. This difficulty is due to the need to solve for a Lagrange multiplier field at every time step during dynamic simulations. Furthermore, standard Mortar contact implementations do not immediately conserve angular momentum during frictional sliding. Another strategy for contact involves a series of increasing energy penalties which ensures no contact overlap, enforces linear and angular momentum conservation, and has good energy conservation properties [12]. However, this approach can be computationally demanding and requires adapting the stable time step to resolve ever-increasing contact penalty frequencies. In the context of meshfree methods, a strategy was recently presented in [18] that ensures exact conservation of both linear and angular momentum, even when contact occurs with friction being enforced at a gap or with some material overlap. A meshfree contact algorithm which uses a level-set reconstruction of the contacting surfaces that can result in smoother interactions was proposed in [8]. Another approach is based on G1 continuous surface patches to ensure smoother interactions was proposed in [8].

We also comment briefly on the primary motivation of the proposed algorithm for engineering application, namely computational efficiency. Common strategies for contact involve a trade-off between accuracy, robustness, and computational efficiency. The fidelity of contact ranges from the high accuracy of Mortar contact methods [27], to, e.g., pinball contact algorithms [6, 19] which use only penalty forces and fast sphere-sphere intersection searches. The approach proposed here emphasizes computational efficiency in explicit dynamic simulations, while retaining accuracy where possible.

Efficiency is obtained here by completely avoiding the often expensive geometric search between nodes, faces, or elements, common to traditional contact strategies. With these traditional contact searches, worst-case computational complexity scales super-linearly in the number of objects [20]. We replace this with two grid transfer operations which scale linearly in the number of nodes in contact. Furthermore, the algorithm is trivially thread scalable, which enables straightforward efficient implementation on many-core and graphics processing units (GPUs). In the following, the focus is on the novel ideas which contribute to the proposed method, and its mechanical and numerical behavior. Analysis of the computational advantages is left for further work. Anecdotally the authors have observed significant performance advantages over classical search-based contact algorithms for explicit dynamic simulations in serial: about an order of magnitude. These advantages become stronger on modern GPUs, as two orders of magnitude speedup or more is observed for sufficiently large problems.

In Section 2 we introduce the proposed contact formulation and motivate the various ingredients. Semi-analytic and computational analyses of the proposed method can be found in Section 3. In addition, we numerically demonstrate energy convergence and the ability to accurately model sliding with frictional contact to verify the model in certain regimes. Finally, conclusions are in Section 4.

2 Formulation

A novel formulation for automatically detecting and penalizing contact via background grid projections is proposed. We build up the important pieces one at a time, starting with an explanation of what we call augury contact dynamics for enforcing contact, and its time discretization. This simplified starting point is then expanded upon to improve conservation and physical realism. In particular, affine particle-in-cell grid transfers are utilized to conserve momentum and improve contact gap resolution, and a novel iteration procedure is introduced to allow for material rebound and friction to occur.

Before describing the proposed contact strategy, we introduce the general problem. Consider any standard nodal finite element or meshfree discretizations of multiple solid mechanical bodies (see, e.g., [4, 15] for finite element approximations and, e.g., [25, 5, 29, 13, 7] for meshfree). For simplicity in this work we use standard linear triangular finite elements with one quadrature point per element. Given a particular spatial discretization we identify all boundary nodes which may be involved in contact, and label these $P$ nodes with index $p$. We will call these boundary nodes particles to be consistent with PIC and MPM terminology. The collection of all nodes in the mesh will be indexed by $n$. This nodal-based ‘foreground’ discretization is where the discrete solution truly exists, and its time evolution is standard:

\[ \dot{x}_n = v_n \]  
\[ v_n = a_n = \frac{f_n}{m_n} \]

where $x$ are the nodal coordinates, $v$ the nodal velocities, $a$ the nodal accelerations, $f$ the external minus internal nodal forces, and $m$ are nodal masses.

In addition to a standard nodal foreground discretization, we employ a structured Cartesian background grid. Velocity and mass fields are approximated on this background grid using B-splines. For simplicity, this work is restricted to
A momentum preserving frictional contact algorithm based on affine particle-in-cell grid transfers

Figure 1: Foreground particles and structured background grid. The background grid nodes (solid black circles) are indexed by \( i \), while the foreground (open blue circles) particles are indexed by \( p \).

using quadratic B-splines. For details, see Appendix B. The use of background B-spline grids and other smooth background functions have become fairly common in MPM and PIC discretizations [16, 17, 2, 3, 23, 28]. In contrast to the approach taken here, typically the background grid in MPM hold the degree of freedom information, while the foreground mesh only tracks or advects with the background grid motion. We label background grid nodes with index \( i \). A schematic depicting foreground particles and a background grid discretizations is shown in Figure 1.

2.1 Augury dynamics

In order to enforce contact in dynamical systems we propose a modification to the equations of motion which uses both a prediction of where the system is heading and a model for the preferred motion. In particular, we derive what we call augury acceleration correction terms which are used to take into account upcoming motions and push the system into a state which does not violate contact non-interpenetration constraints. Given a system with position, velocity, and acceleration \( x_t, v_t, a_t \) at a given time \( t \), a simple prediction of the velocity state at some time in the future \( t + \tau \), \( \tau > 0 \), is

\[
\bar{v}_{t+\tau} = v_t + a_t \tau,
\]

(3)

where we use the notation subscript \( t \) to denote the time dependence of these fields. In addition, we provide a model which considers this predicted velocity and the current position state and produces a preferred velocity field:

\[
\hat{v}_t : \mathbb{R}^{3P} \times \mathbb{R}^{3P} \rightarrow \mathbb{R}^{3P}
\]

\[
\hat{v}_{t+\tau} := \hat{v}(\bar{v}_{t+\tau}, x_t)
\]

(4)

where \( P \) is again the number of boundary particles and 3 is the spatial dimension of the velocity vector. This preferred future velocity function \( \hat{v}_{t+\tau} \) provides a model for the contact interaction behavior and different versions of it will be described in the upcoming sub-sections. We modify the equations of motion by solving for the augury acceleration corrections \( \hat{a}_t \) which move the system toward this preferred velocity state:

\[
\hat{a}_t = \frac{1}{\tau} (\bar{v}_{t+\tau} - \bar{v}_{t+\tau})
\]

(6)

which ensures

\[
\hat{v}_{t+\tau} = \bar{v}_{t+\tau} + \hat{a}_t \tau.
\]

(7)

We can simplify the equations slightly in the case where the preferred velocity is a linear function of the predicted velocity:

\[
\hat{v}(v_{t+\tau}, x_t) = H_t \bar{v}_{t+\tau},
\]

(8)

where \( H_t = H(x_t) \) is just a matrix which depends only on the particle coordinates, and the augury acceleration correction is now

\[
\hat{a}_t = \frac{1}{\tau} (H_t - I) \bar{v}_{t+\tau}.
\]

(9)
This modification to the equations of motion results in a system whose dynamics is altered gradually to track toward the desired state where the non-interpenetration conditions encoded in $\hat{v}$ are approximately satisfied.

To motivate the use of these augury correction terms which use a prediction of the state at some point in the future, consider the limiting case of $\tau = 0$. In this case we could choose to simply reset the system dynamics to the preferred state $\hat{v}_t$. However, this would require infinite contact forces to be applied and leads to unstable dynamics. Allowing a finite value for $\tau$ overcomes this limitation. In practice (see Section 3.2), we set $\tau$ proportional to the stable time step of the time discretized system.

### 2.1.1 Augury dynamics using PIC grid transfers

One option for defining the preferred velocity field $\hat{v}_t$ is to perform a velocity field grid transfer from the foreground boundary particles to a background grid, followed by another transfer back from the background to the particles. The motivation for this is to exploit the natural and efficient contact detection properties of PIC-like grid transfers. Two separate bodies with distinct particle velocity fields will project their velocity fields onto a common background grid where the velocity is necessarily continuous. This has the side-effect of having the two bodies tending to stick together as they approach each other. This is both good and bad, as contact is automatically enforced, but rebound cannot occur if the two bodies get sufficiently close. For now we will focus on the positive feature and address the sticking limitation in Section 2.3.

We review the standard particle-in-cell transfer employed here. First, mass and momentum are transferred from particles to the background grid:

$$m_i = \sum_p N_i(x_p)m_p,$$

and

$$m_i\mathbf{v}_i = \sum_p N_i(x_p)m_p\mathbf{v}_p,$$

where $\mathbf{v}_p$ are particle velocities, $m_p$ are particle masses, and the background grid spline shape functions $N_i$ for grid point $i$ are evaluated at particle locations $x_p$. The resulting grid velocities and grid masses are $\mathbf{v}_i$ and $m_i$. Second, the velocity is interpolated back to the particles

$$\bar{v}_p = \sum_i N_i(x_p)\mathbf{v}_i.$$  

This new velocity field on the particles is a smoothed version of the original field and the entire process can be thought of as a linear mapping

$$\bar{v} = H^{PIC}(\mathbf{x})\mathbf{v},$$

which means it satisfies the linear functional assumption in Equation (8).

The most basic augury dynamics strategy to enforce contact is to use this PIC smoothed field as the preferred velocity field:

$$\hat{v}(t + \tau, x_t) = H^{PIC}(x_t)\bar{v}(t + \tau).$$

The corresponding acceleration correction is then

$$\hat{a}_t = \frac{1}{\tau} \left( H^{PIC}_t - I \right) (\mathbf{v}_t + a_t\tau).$$

The standard PIC grid transfers described here are known to conserve linear momentum [17], so it follows that any velocity field $\mathbf{v}$ and its associated PIC smoothed field $\bar{v} = H^{PIC}\mathbf{v}$ have the same net linear momentum. As a result the acceleration correction Equation (15) has no net effect on the total system linear momentum. Consequently, we can think of the linear operator

$$G^{PIC} := H^{PIC} - I,$$

as one which maps any velocity field to a field with no net linear momentum.

---

1 The particle-in-cell methods in [16, 17, 11] perform their time integration on the background grid. As a result the particle-to-grid transfer occurs at a different time than the subsequent grid-to-particle transfer.
2.1.2 Augury dynamics using APIC grid transfers

While the PIC transfers used in the previous section are known to conserve linear momentum, angular momentum conservation is not guaranteed. Inspired by the affine particle-in-cell (APIC) approach proposed in [16, 17], we consider APIC grid transfers, which are (10) together with

$$m_i v_i = \sum_p N_i(x_p) m_p [v_p + B_p D_p^{-1}(x_i - x_p)],$$  \hspace{1cm} (17)

and

$$D_p = \sum_i N_i(x_p) (x_i - x_p) \otimes (x_i - x_p),$$  \hspace{1cm} (18)

where $B_p$ is a tensor that depends on the velocity, defined shortly in equation (20). Compared to more standard MPM or PIC transfers, the APIC approach enriches the particle velocities with additional degrees of freedom $B_p$, which we will call the scaled velocity gradient since $B_p D_p^{-1}$ corresponds to a local approximation of the velocity gradient field. The initial condition for $B_p$ can be solved for from the initial conditions to ensure $B_p D_p^{-1}$ matches the initial velocity gradient. The use of these extra degrees of freedom ensures that any affine motion in the particles will be exactly reproduced on the background spline grid. The grid-to-particle transfer for the APIC method is then

$$\tilde{v}_p = \sum_i N_i(x_p) v_i,$$  \hspace{1cm} (19)

and

$$B_p = \sum_i N_i(x_p) v_i \otimes (x_i - x_p).$$  \hspace{1cm} (20)

When applied sequentially in this manner, these two transfer operations combine into a linear operator $H_{APIC}: \mathbb{R}^{3P+9P} \rightarrow \mathbb{R}^{3P+9P}$, where $P$ is again the number of particles on the boundary, and the 3 and 9 are the number of components for the velocity vector and the scaled velocity gradient tensor. We call $H_{APIC}$ the APIC transformation operator. Given a generalized velocity field

$$\mathcal{V} = \{v, B\},$$  \hspace{1cm} (21)

the linear operator $H_{APIC}$ returns a new velocity field and scaled velocity gradient field which have been altered by the two grid transfers

$$\tilde{\mathcal{V}} = H_{APIC} \mathcal{V}.$$  \hspace{1cm} (22)

There are a few key properties of this linear operator. The first is that affine velocity fields are exactly preserved under $H_{APIC}$:

$$\tilde{\mathcal{V}} = H_{APIC} \tilde{\mathcal{V}},$$  \hspace{1cm} (23)

for any $\tilde{\mathcal{V}}$ with

$$\tilde{v}_p = v_0 + L x,$$  \hspace{1cm} (24)

and

$$\tilde{B}_p = L D_p,$$  \hspace{1cm} (25)

where $L$ is a constant velocity gradient tensor. A proof of this is provided in Appendix A. In addition, linear and angular momentum are both preserved under the operation $H_{APIC}$ for any $\mathcal{V}$. This follows directly from the proofs in [17], where both grid-to-particle and particle-to-grid transfers were separately shown to be both linear and angular momentum preserving.

Similar to the PIC case in Section 2.1.1, it follows from the above that we can define an operator

$$G_{APIC}: \mathbb{R}^{3P+9P} \rightarrow \mathbb{R}^{3P+9P},$$

$$G_{APIC} = H_{APIC} - I,$$  \hspace{1cm} (26)

which takes an arbitrary generalized velocity field and returns a generalized velocity field with no net linear or angular momentum.
2.2 APIC augury dynamics for contact

To enforce contact in explicit dynamic simulations, a modification to the standard equations of motion for the spatially discretized system is presented. This modification allows the use of the APIC transformation operator $H^{APIC}$ to automatically detect and enforce contact. Similar to the PIC case, $H^{APIC}$ tends to return transformed velocity fields which discourage contact penetration due to the particles sticking together within cells, an effect which is common to many MPM methods.

For particles with current velocity $v$, scaled velocity gradient $B$, and mechanical acceleration $a$ (the acceleration in the absence of contact), we can predict the generalized velocity at time $t + \tau$

$$\dot{V}_{t+\tau} = V_t + \tau A_t,$$

where $V$ is the generalized velocities as before, $\dot{V}$ is the predicted generalized velocity in the future, and $A$ is the corresponding generalized accelerations

$$A_p = \left\{ a_p, \dot{B}_p \right\}.$$  \hfill (28)

The mechanical scaled acceleration gradient rate is always set to zero, $\dot{B}_p = 0$, as there is no classical internal work on that mode. APIC augury dynamics modifies the equations of motion by applying the APIC transformation to $\dot{V}_{t+\tau}$ and solving for an additional acceleration term which supplies the APIC augury contact force correction

$$\dot{A}_t = \frac{1}{\tau} \left( H^{APIC} - I \right) \dot{V}_{t+\tau},$$

with $I$ the identity operator. This correction is the acceleration needed to reach the preferred generalized velocity, assuming all accelerations are constant over the time $t \rightarrow t + \tau$, and satisfies

$$H^{APIC} \dot{V}_{t+\tau} = V_t + \tau \left( A_t + \dot{A}_t \right).$$  \hfill (30)

The resulting APIC augury equations of motion become

$$\dot{x}_t = v_t$$

$$\dot{V}_t = A_t + \dot{A}_t,$$  \hfill (31)

where $\dot{A}_t$ is by construction a linear and angular momentum free acceleration correction and $A$ inherits its linear and angular momentum conserving properties from the foreground discretization. Note that the operator $H^{APIC}$ depends on the particle coordinates $x_p$ and masses $m_p$ at time $t$. As mentioned before $\dot{B}_p = 0$, but $\dot{\dot{B}}_p \neq 0$ in general. We integrate in time using the Newmark time stepping algorithm, with Newmark parameters chosen for explicit time integration and second order accuracy [10]

$$V^{n+1/2}_p = V^n + \frac{1}{2} \Delta t A^n_p$$

$$x^{n+1} = x^n + \Delta t V^{n+1/2}_p$$

$$A^{n+1}_p = \{ a(x^{n+1})_p, 0 \} + \dot{A}^{n+1}_p$$

$$V^{n+1/2}_p = V^n + \frac{1}{2} \Delta t A^{n+1}_p,$$  \hfill (32)

where $a(x)_p$ is the mechanical acceleration model for particle $p$ arising from the foreground finite element discretization and, e.g., particular material modeling choices. The augury correction $\dot{A}^{n+1}_p$ is the particle acceleration correction arising from (29) at time step $n + 1$. Changing timesteps between steps can readily be included, but is omitted here for brevity.

We call the resulting algorithm the *APIC augury* algorithm for contact to distinguish it from the PIC algorithm for contact in [17] which uses a background grid update step to compute the momentum impulse applied to the foreground solids. When applying this algorithm without the scaled velocity gradient term (e.g. by setting $D_p = 0$), we call the resulting algorithm the *PIC augury* algorithm. Results using these formulations are shown in Figure 2 where we simulate two blocks contacting at an angle. Details of the simulation setup are in Section 3. We observe a few interesting features. Upon contact, the two bodies become stuck together, as is expected for many PIC methods in which particles share a continuous background velocity field. A less intuitive feature of these simulations is that the APIC augury algorithm results in the two bodies sticking at essentially zero visible gap/overlap. This effect appears
to be insensitive to both foreground mesh size and background grid size. The reason for this behavior is not obvious, but it is desirable when contrasted with using a more classical PIC transfers (PIC augury dynamics), as shown in Figure 3. For the lower order PIC grid transfers, the sticking occurs at a significant offset distance, proportional to the background grid size.

In addition, we plot angular momentum over time for several cases in Figure 8. Only the APIC transfers guarantee angular momentum conservation for the simulation, as is expected from the construction of the algorithm. Linear momentum is not shown, but it is conserved for both PIC or APIC augury dynamics. This is in contrast to the method in [11] where even linear momentum conservation is not guaranteed in general. It has also been shown that APIC transfers generally dissipate less energy than standard PIC transfers [16, 9], which is consistent with the results for this problem as shown and discussed further in Section 3.

For the results presented, the background grid consists of uniform hexahedrons with edge lengths approximately twice the average finite element edge length. If the background grid size is too small, it is possible for particles to pass each other without interacting through the background grid transfers. This results in missed or inaccurate contact interactions.

Figure 2: Contacting blocks using APIC grid transfers. The background grid size is set to approximately twice the element edge length.

2.3 Material separation

The previous APIC correction is an improvement over traditional MPM transfers in terms of linear and angular momentum conservation, however there is still a clear limitation in that the two bodies completely stick after contact. To address this we propose an iterative algorithm for calculating the augury acceleration correction which is based on a
A momentum preserving frictional contact algorithm based on affine particle-in-cell grid transfers

sequence of momentum-preserving grid transfer operations. Consider a grid transfer linear operator $H$ (either PIC or APIC depending on whether angular momentum conservation is considered important):

$$\hat{\mathbf{V}} = H \mathbf{V}$$  \hspace{1cm} (33)

which we can alternatively write as

$$\hat{\mathbf{V}} = \mathbf{V} + \mathbf{G} \mathbf{V},$$ \hspace{1cm} (34)

where we use $\mathbf{G} = H - I$. The associated velocity correction associated with PIC grid transfers is

$$\Delta \mathbf{V} = \hat{\mathbf{V}} - \mathbf{V} = \mathbf{G} \mathbf{V}.$$ \hspace{1cm} (35)

As $\hat{\mathbf{V}}$ and $\mathbf{V}$ have the same momentum (including angular if APIC is used), we conclude that $\Delta \mathbf{V}$ must have no net momentum, and this must hold for any $\mathbf{V}$. As the range of $\mathbf{G}$ contains only zero net momentum fields, we can construct a new field with the same net momentum as $\mathbf{V}$ with any series of the form

$$\hat{\mathbf{V}} = \mathbf{V} + \mathbf{G}_0 \mathbf{V} + \sum_{n=1}^{N} \mathbf{G}_n \mathbf{W}_n,$$ \hspace{1cm} (36)

where $N$ is the number of so-called augury iterations, and $\mathbf{W}_n$ are arbitrary generalize velocity fields. Once again the momentum of $\hat{\mathbf{V}}$ and $\mathbf{V}$ are equal provided that the $\mathbf{G}_n$ are all associated with momentum preserving grid transfers $H_n$.

While we allow for the generality of potentially using different grid transfer operations for each term in this series, in practice we simplify to the case where each transfer operation is taken to be the same, $\mathbf{G}_i = \mathbf{G}$, $i = 0, ..., N$. 
A strategy for allowing material rebound is to construct a series of this form with desirable mechanical properties. In particular, we want to allow particle interactions through the background grid when bodies are coming into contact, but prevent their interaction when their velocities are opposing as in [11] to allow them to separate. In the following, we compute a normal vector $\mathbf{n}_p$ for each particle $p$ in the current configuration by area weighting the outward normal vectors of the boundary faces adjacent to each finite element node/particle in contact.

The iterative algorithm for enforcing frictionless sliding proceeds as follows. At step $m$ with $m < N$ we compute the currently proposed modified velocity field for the particles as:

$$\hat{\mathbf{v}}_m := \mathbf{v} + \mathbf{Gv} + \sum_{n=1}^{m} \mathbf{GW}_n, \quad m < N. \quad (37)$$

Consider $\Delta \mathbf{v}_{m,p} = \hat{\mathbf{v}}_{m,p} - \mathbf{v}_p$, for particle $p$, which is the proposed change in the particle velocity due to the PIC grid transfers up to this point. For $\Delta \mathbf{v}_{m,p} \cdot \mathbf{n}_p > 0$, the proposed change to the velocity field acts to pull this particle away from its surface. This is undesirable, as it indicates that particles are moving away from each other and we want the bodies capable of separation after contact. To alleviate this, the undesirable component of the proposed velocity change can be iteratively smoothed out by setting

$$W_{m+1,p} = \begin{cases} \left[ \frac{\Delta \mathbf{v}_{m,p}}{\Delta \mathbf{B}_{m,p}} \right] & \text{if } \Delta \mathbf{v}_{m,p} \cdot \mathbf{n}_p > 0 \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \text{otherwise.} \end{cases} \quad (38)$$

For the next term in the series, $W_{m+1}$ is acted upon by the operator $\mathbf{G}$. This has the effect of gradually removing the contribution $W_{m+1}$ from the series $\hat{\mathbf{v}}_{m+1}$. The intuition behind this effect is discussed below, but first a definition.

We define the quasi-null-space of a grid transfer operator $\mathbf{H}$ as consisting of the generalized velocity fields $\mathbf{V}$ where $\mathbf{V} = \mathbf{HV}$ results in $\hat{\mathbf{v}} = \mathbf{0}$. In other words, the generalized velocity fields which result in a zero physical velocity field (and any scaled velocity gradient) after the operation $\mathbf{H}$ has been applied. For $\mathbf{V}$ in the quasi-null-space of $\mathbf{H}$, $\mathbf{W}' = \mathbf{GW} = \mathbf{HV} - \mathbf{W}$ results in a field with

$$\mathbf{w}' = -\mathbf{w}, \quad (39)$$

so the operation $\mathbf{G}$ flips the sign of the physical part of the velocity for any field $\mathbf{W}$ in the quasi-null-space of the grid transfer linear operator $\mathbf{H}$. A simple example of such a field is two equal mass particles with equal and opposite velocities interacting symmetrically in the same background cell. In this case their momentum cancel, resulting in a transformed velocity field with zero particle velocities $\mathbf{v}_p = \mathbf{0}$, but may still have $\mathbf{B}_p \neq \mathbf{0}$. Additionally, there are fields $\mathbf{W}$ in the quasi-null-space of $\mathbf{G}$, the most obvious example of this is any affine generalized velocity field, see equations (24)–(25). The intuition is that the operator $\mathbf{G}$ tends to flip the sign of the physical part of the velocity field for fields which are non-affine, e.g., small wave-length features, but yields zero physical velocities for affine fields and near zero velocities for fields changing slowly over large wave-lengths.

More analysis is needed to better understand the spectral properties of the operator $\mathbf{H}^{APIC}$ and how quickly we expect these iterations to converge in practice. For this work, we only apply a single additional augury iteration, $N = 1$. We have not formally analyzed the convergence of the algorithm, but a small number of iterations produced reasonable results in the cases tried. More work in this area is certainly warranted, but beyond the current scope. Results repeating the earlier two block simulation using this iterative rebounding strategy are shown in Figure 4 using APIC transfers. We observe that the blocks bounce cleanly off each-other while sharply capturing the contact interaction, and that both linear and angular momentum are conserved. Furthermore, while there is energy dissipation associated with the collision, the energy loss decreases under mesh and grid refinement. This improved energy conservation behavior under refinement will be discussed further in Section 3.2.

### 2.4 Sliding and friction

We extend this iterative strategy to frictional contact, where the goal is to have particle velocity changes $\Delta \hat{\mathbf{v}}_{m,p}$ which satisfy

$$\Delta \hat{\mathbf{v}}_p \cdot \mathbf{n}_p \leq 0 \quad (40)$$

$$||\Delta \hat{\mathbf{v}}_p|| \leq \mu \Delta \hat{\mathbf{v}}_p \cdot \mathbf{n}_p, \quad (41)$$
A momentum preserving frictional contact algorithm based on affine particle-in-cell grid transfers

where $\mu$ is the Coulomb friction coefficient, and $\Delta \hat{v}_p^t = \Delta \hat{v}_p n_p \otimes n_p$ is the velocity change perpendicular to the contact surface. Here we drop the $m$ index to simplify syntax when needed. At a given step $m$ we compute the undesirable portion of the generalized velocity, $\Delta V^u_p$. After evaluating

$$
\Delta v^f_p = \Delta v_p n_p \otimes n_p
$$

(42)

and

$$
s^f_p = \min (\| \Delta v^f_p \|, -\mu \Delta v_p \cdot n_p)
$$

(43)

we compute the undesirable part of the physical velocity change

$$
\Delta v^u_p = \begin{cases} 
\Delta v_p - (\Delta v_p \cdot n_p) n_p - s^f_p \frac{\Delta v^f_p}{\| \Delta v^f_p \|} & \text{for } \Delta v_p \cdot n_p \leq 0 \\
\Delta v_p & \text{otherwise}
\end{cases}
$$

(44)

and the undesirable part of the scaled velocity gradient change

$$
\Delta B^u_p = \Delta B_p.
$$

(45)

We then set

$$
W_{m+1,p} = \begin{bmatrix} 
\Delta v^u_p \\
\Delta B^u_p
\end{bmatrix}.
$$

(46)

This iteratively enforces a rate form of the Coulombic friction model. The choice to always treat the entire scaled velocity gradient difference as undesirable is debatable, but the intuition is to try and find a momentum conserving, contact enforcing solution with minimal changes to these enriched velocity gradient modes.
3 Results

We apply this new contact algorithm on a few simple example problems to assess its behavior. We begin by analyzing a two-particle system in 1D to gain insight into the algorithm's behavior. Next, a block sliding with friction down an inclined plane is simulated to assess the accuracy and convergence properties of the friction model. Finally, we demonstrate the smoothness properties of the proposed algorithm.

3.1 Contact gap analysis for two particles in a background grid

A well known behavior of MPM-like methods is that two bodies coming into contact tend to stick together at a finite distance proportional to the background grid size. This is attributed to the continuous velocity field approximation in the background grid cells which results in the contacting particles following the same flow field. Typically, when two particles enter the same cell the relative velocity between them will reach zero before the particles come into direct contact. This error can artificially stiffen the response of multiple interacting bodies, and lead to larger total volumes than physically expected. While this error does go away as the background grid is refined, it can remain as a significant source of error for certain applications.

Remarkably, numerical experiments using the proposed APIC method found that this issue is significantly eliminated. Contact (though sticky) automatically occurs at near zero visible gap. This involved no additional algorithmic heuristics, and is an especially unexpected feature as the particles interacting through contact are completely unaware of each other’s locations and velocities. The only mechanism for communication between particles is through the background grid. While we must admit that this feature is not yet fully understood, we provide a simple two particle analysis which helps unravel at least a piece of this fortunate mystery. Consider the case where we have two particles of equal mass \( m \) in 1D with initial coordinates \( x_0 \) and \(-x_0\), and initial velocities \(-v_0\) and \(v_0\), respectively. We consider the dynamics of these particles as they enter the same background cell which extends from \(-L\) to \(L\) (see Figure 5). When considering the APIC augury dynamics of particles in the absence of mechanical forces, the equations of motion (31) simplify to

\[
\begin{align*}
\dot{x}_t &= v_t \\
\dot{v}_t &= \frac{1}{\tau} G^{APIC}(v_t) \quad (48)
\end{align*}
\]

We can non-dimensionalize by a change of variables

\[
\begin{align*}
x &\leftarrow \frac{x}{L} \\
v &\leftarrow \frac{v}{v_0} \\
\tau &\leftarrow \frac{\tau v_0}{L} \\
B &\leftarrow \frac{B}{v_0 L}
\end{align*}
\]

so that we only consider \( v_0 = 1 \) and \( L = 1 \). To simplify the analysis we assume linear shape functions for the background grid approximation, \( N_0(x) = \frac{1}{2} (1 - x) \) and \( N_1(x) = \frac{1}{2} (1 + x) \). By applying the grid transfers (17) and (19), we find that APIC augury dynamics in this case simplifies to

\[
\begin{bmatrix}
\dot{x}_t \\
\dot{v}_t \\
\dot{B}_t
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & 0 \\
x^2 - 1 & -x & 0 \\
x - x^2 & x^3 - x & B
\end{bmatrix}
\begin{bmatrix}
v \\
B
\end{bmatrix}, \quad (49)
\]

where \( x, v, B \) are the non-dimensionalized coordinates, velocities, and scaled velocity gradient of the left particle. Using standard PIC grid-transfers by setting \( B_p = 0 \) in equation (17) and removing its evolution, PIC augury dynamics

\footnote{For the full-scale simulation later in the paper, we always employ quadratic B-Splines for the background grid approximation space. This 1D analysis is therefore only to gain intuition about a possible mechanism for the zero contact gap behavior.}
is simply
\[
\begin{bmatrix}
\dot{x}_t \\
\dot{v}_t \\
\dot{B}_t
\end{bmatrix} = \begin{bmatrix}
1 & 0 & G_{11}(x) \\
0 & 1 & G_{12}(x) \\
0 & 0 & G_{21}(x) & G_{22}(x)
\end{bmatrix} \begin{bmatrix}
v \\
B
\end{bmatrix}.
\]
(50)

We gain intuition about the difference between the PIC and APIC augury dynamics by looking at the fixed points of the dynamical system. With the PIC equations, we find if \( v = 0 \), then \( \dot{v} = 0 \) for any \( x \). This indicates that the particles may come to rest at any distance from each other. For the APIC equations, things are a bit more interesting, as for any \( B \neq 0 \) if \( v = 0 \), then \( \dot{v} = 0 \) only when \( x = 0 \). Furthermore, if \( x = 0 \), then \( \dot{B} = 0 \), so \( v = x = 0 \), is the only fixed point of this dynamical system when \( B \neq 0 \). In other words, the particle can only come to rest once they are co-located, provided \( B \neq 0 \). Numerical evidence suggests that this picture of the dynamics is qualitatively correct: standard PIC results in particles halting at a finite distance, while APIC results in particles halting only when they are co-located.

Some numerically computed solutions to these equations of motion are shown in Figures 6 and Figures 7 for APIC and PIC dynamics respectively. The key images are in Figures 6.a and Figures 7.a, where the particle coordinates vs time are plotted for various values of the non-dimensional parameter \( \tau_{\text{vol}}/L \). It is evident for PIC dynamics that the particles in general stick at a nonzero gap which is related to both the initial velocity and grid size. For APIC, the particles come to rest only once they are both at \( x = 0 \), and therefore the steady-state contact is enforced at zero gap. We emphasize that this is a feature only of the steady-state behaviour. It is still the case with APIC augury dynamics that the particles begin to have forces applied between them as soon as they enter the same grid cell and well before then finally stop at \( x = 0 \). The remainder of images in Figures 6 and Figures 7 show the solution for the other terms in the equations of motion for completeness.

These results indicate that the essential ingredient for obtaining perfect contact gaps with background cell methods is the addition of the APIC scaled velocity gradient term, while the background grid shape functions may be of secondary importance. In fact, the dynamics for an arbitrary background grid approximations space are generically written:
\[
\begin{bmatrix}
\dot{x}_t \\
\dot{v}_t \\
\dot{B}_t
\end{bmatrix} = \begin{bmatrix}
1 & 0 & G_{11}(x) \\
0 & 1 & G_{12}(x) \\
0 & 0 & G_{21}(x) & G_{22}(x)
\end{bmatrix} \begin{bmatrix}
v \\
B
\end{bmatrix}.
\]
(51)

So long as \( G_{21} \neq 0 \) for some \( x \) is the path of the dynamics (to induce a change in \( B \)), and \( G_{12}(x) = 0 \) if and only if \( x = 0 \) and \( G_{22}(0) = 0 \), the same arguments as above will hold. In higher dimension, we believe smoothness of the background grid also likely plays an important role in the robustness and quality of the solution.

The analysis in this section has a few important caveats that are worth mentioning. The first is that as the initial velocity dimensionless parameter \( \tau_{\text{vol}}/L \) increases, the interacting particles can actually pass each other, lose contact, and go unstable. This highlights a possible limitation for the algorithm for high velocities relative to grid size. It is therefore essential in practice to choose a \( \tau \) which would ensure the system dynamics remains stable. Another caveat is that it is not straightforward to reproduce a similar dynamical system behavior using a background 1D B-spline grid (namely sticking at zero gap), so the conclusions drawn from this analysis may be limited. The mechanism for zero-gap sticking presented here may indeed be a large part of the explanation, but an understanding of the full mechanism in 3D using background smooth spline grids requires additional investigation. We speculate that there is also a multi-dimensional aspect of this phenomena which is essential to explain the zero-gap sticking behavior in the 3D simulations. Despite these limitation, the general trend of sticking at nearly zero gap, and loss of contact enforcement at high velocity to mesh-size ratios is consistent with 3D simulation results.

### 3.2 Evaluation of conservation of momentum and energy

To assess conservation of angular momentum and energy in a reasonably general scenario, we simulate two blocks impacting each other at an angle. Both blocks are 3x3x3 cubes. Block 1 is axis aligned and centered around the origin. Block 2 begins coincident with Block 1, is first translated by 3.8 in \( y \), then rotated by 22° about the global \( x \)-axis. The material property of both blocks is a density of 2700, a Young’s modulus of \( 1 \times 10^8 \), and a Poisson’s ratio of 0.3. A compressible Neo-Hookean hyperelastic material model is used. Block 1 has an initial velocity of \( v_0 = (0, 35, 0) \), while block 2 has an initial velocity of \( v_0 = (-15, -35, -15) \). Different foreground mesh sizes are simulated with \( h = 2^k \) for \( k = \{0, -1, -2, -3, -4\} \). For a given \( h \), the background grid size is always set to \( 2h \). Similarly the fixed time step is taken as \( \Delta t = 1 \times 10^{-4}h \), and the augury time for each case is taken as \( \tau = \Delta t \). Example shap-shots of these simulations are shown in Figure 4.

Results showing conservation of angular momentum for different formulations are shown in Figure 8. Only the APIC versions exactly conserve angular momentum. Results for total energy conservation are shown in Figure 9. We see that as the mesh is refined, the APIC versions do a significantly better job of conserving energy, though all are somewhat
A momentum preserving frictional contact algorithm based on affine particle-in-cell grid transfers

Figure 6: Particle dynamics of APIC augury equations with varying $\tau$.

Figure 7: Particle velocity of PIC augury equations with varying $\tau$. 
A momentum preserving frictional contact algorithm based on affine particle-in-cell grid transfers

dissipative. For APIC, the dissipation is due to the fact that any component of the velocity field which is non-affine will be damped slightly due to the APIC grid transfers. For PIC, any non-constant velocity fields will be damped over time. The difference in initial energies is due to the simulation setup where the two bodies are instantly interacting through grid transfers, so their initial energies are slightly damped. This effect decreases as the mesh and background grids are refined.

While the proposed method shows good promise for the regimes demonstrated here, we acknowledge that there are regimes in which it is still unstable. This typically manifests as the contacting objects eventually losing their repulsive contact forces, interpenetrating and falling into one another. This is currently a limiting factor in the practical use of this MPM-based contact algorithm for certain applications. This means additional efforts are required to fully exploit the efficiency of this style of contact, and we believe the analysis in 3.1 may help shed light on this current limitation. Another possibility is that the iterative strategy for enforcing friction and rebound may be too aggressive at adjusting nodal velocities and may incidentally be removing desirable contact forces. Despite these remaining challenges, the novel ideas and approaches presented here provide a unique and promising step forward relative to previous MPM contact strategies.

Figure 8: Angular momentum conservation with different grid-transfer strategies.

For computing kinetic energy, we are presently neglecting a contribution associated with the scaled velocity gradient field on the finite element mesh. This means that the energy conservation for the APIC cases is actually a bit better than indicated, so this omission does not diminish the essential conclusions presented here. In addition, this contribution to the total kinetic energy is expected to vanish as the mesh is refined.

---

3For computing kinetic energy, we are presently neglecting a contribution associated with the scaled velocity gradient field on the finite element mesh. This means that the energy conservation for the APIC cases is actually a bit better than indicated, so this omission does not diminish the essential conclusions presented here. In addition, this contribution to the total kinetic energy is expected to vanish as the mesh is refined.
A momentum preserving frictional contact algorithm based on affine particle-in-cell grid transfers

Figure 9: Total energy conservation with different grid-transfer strategies.
3.3 Frictional sliding down a ramp

A frictional sliding example demonstrates the method for the case of sliding with frictional contact. The geometry consists of a 1x1x1 unit cubic block sliding down a ramp which is tilted at a slope of 30°, as shown in Figure 10. The ramp is 4 units long, 1.5 wide, and 0.25 thick. The bottom face of the ramp is fully fixed. The density, Young’s modulus, and Poisson’s ratio for both ramp and block are $1 \times 10^6$, $1 \times 10^{10}$, and 0.3 respectively. A gravity load of $g = -100$ is applied, and the time step is chosen to be $\Delta t = 2.5 \times 10^{-5}h$, for approximate mesh size $h = \{0.25, 0.33, 0.5, 1.0\}$. The MPM grid size is taken to be $0.5h$, and the augury time is taken as $\tau = \Delta t$. Results showing the displacement history of the block for varying mesh sizes and friction coefficients are shown in Figure 11. The analytic solution is also shown in black. The results using both an PIC and APIC transfer are shown. Clearly the use of APIC helps improve accuracy for these cases, and there appears to be improvement toward the exact solution as the mesh and background grid are refined.

![Figure 10: Snapshots of block sliding down a ramp using APIC and $\mu = 0.2$.](image)

4 Conclusions

The proposed APIC contact algorithm has two clear advantages over established contact algorithms based on PIC methods: contact enforcement at near zero gap, and exact conservation of both linear and angular momentum. Furthermore, the proposed algorithm has two advantages over established contact algorithms for explicit dynamic finite element simulations. PIC-based contact algorithms require no object-to-object search, as all information is simply exchanged through a common background grid. PIC-based algorithms have linear computational complexity and are naturally both shared memory and distributed memory scalable. These two performance advantages results in a more efficient algorithm for enforcing contact than is traditionally possible. While the novel ideas presented here show significant promise, opportunities for improvement remain. Ongoing work includes: continuing to analyze the theoretical properties of the algorithm, and minimizing energy dissipation by generalizing the algorithm to work effectively even when the background grid is small relative to foreground mesh.

Acknowledgments

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

Conflict of interest

On behalf of all authors, the corresponding author states that there is no conflict of interest.

A Linear completeness

We prove one of the properties claimed in Section 2: that affine velocity fields are preserved under grid transfer operation $\mathbf{H}^{APIC}$. First we have to establish two standard properties of our background grid approximations, namely partition of unity:

$$\sum_i N_i(\mathbf{x}) = 1, \quad (52)$$
A momentum preserving frictional contact algorithm based on affine particle-in-cell grid transfers

\[ \sum_i N_i(x) x_i = x. \quad (53) \]

The specific form of quadratic B-Splines which we use are described in B. Suppose each particle velocity starts with an affine velocity field \( \mathbf{v}(x) = \mathbf{v}_0 + Lx \), where \( \mathbf{v}_0 \) is the velocity at \( x = 0 \) and \( L \) is a constant velocity gradient tensor, so

\[ \mathbf{v}_p = \mathbf{v}_0 + Lx_p, \quad (54) \]

and

\[ \mathbf{B}_p = L\mathbf{D}_p. \quad (55) \]

Figure 11: Displacement in x for a block sliding down a ramp with varying friction coefficient.
A momentum preserving frictional contact algorithm based on affine particle-in-cell grid transfers

The particle-to-grid transfer results in grid velocities
\[ m_i \mathbf{v}_i = \sum_p N_i(x_p) m_p \left[ \mathbf{v}_0 + \mathbf{L} x_p + \mathbf{L} (x_i - x_p) \right] \]
\[ = \sum_p N_i(x_p) m_p \mathbf{v}_0 \]
\[ = \sum_p N_i(x_p) m_p \tilde{\mathbf{v}}(x_i) \]
\[ = m_i \tilde{\mathbf{v}}(x_i). \] \hfill (56)

The subsequent grid-to-particle transfer then gives
\[ \mathbf{v}_p = \sum_i N_i(x_p) \mathbf{v}_i \]
\[ = \sum_i N_i(x_p) \tilde{\mathbf{v}}(x_i) \]
\[ = \sum_i N_i(x_p) \left( \mathbf{v}_0 + \mathbf{L} x_i \right) \]
\[ = \mathbf{v}_0 + \mathbf{L} x_p \]
\[ = \tilde{\mathbf{v}}(x_p), \] \hfill (57)

where we use (52) and (53). Finally we confirm that the scaled velocity gradient term is unchanged by the grid transfers:
\[ \mathbf{B}_p = \sum_i N_i(x_p) \tilde{\mathbf{v}}(x_i) \otimes (x_i - x_p) \]
\[ = \sum_i N_i(x_p) \left( \mathbf{v}_0 + \mathbf{L} x_i \right) \otimes (x_i - x_p) \]
\[ = \mathbf{v}_0 \otimes \sum_i N_i(x_p) (x_i - x_p) + \sum_i N_i(x_p) \mathbf{L} x_i \otimes (x_i - x_p) \]
\[ = \mathbf{v}_0 \otimes (x_p - x_p) + \mathbf{L} \sum_i N_i(x_p) x_i \otimes (x_i - x_p) \]
\[ = \mathbf{L} \sum_i N_i(x_p) x_i \otimes (x_i - x_p) \]
\[ = \mathbf{L} \left( \sum_i N_i(x_p) x_i - x_p \right) \otimes (x_i - x_p) \]
\[ = \mathbf{L} \mathbf{D}_p, \] \hfill (58)

where we use
\[ \sum_i N_i(x_p) x_p \otimes (x_i - x_p) = x_p \otimes \sum_i N_i(x_p) (x_i - x_p) = x_p \otimes (x_p - x_p) = 0. \] \hfill (59)

This proves that affine velocity fields are preserved by APIC grid transfers.

**B Quadratic B-Spline Basis**

For the background grid approximation space, quadratic B-splines are employed. We adjust the standard quadratic B-splines slightly to ensure both partition of unity and linear completeness [31], as is required for the proof in A. Only the 1D shape functions are defined here. The full 3D approximation space is constructed from the standard outer product of 1D spline spaces. For a point \( x \) in a 1D grid with grid size \( h \), we compute the grid point location just to the left as
\[ x_i = h \operatorname{floor} \left( \frac{x}{h} \right). \] \hfill (60)
Using
\[ \xi = \frac{x}{h} \text{ and } i = \frac{x_i}{h}, \]
the shape functions which cover \( x \) are
\[ \begin{align*}
N_{i-1}(\xi) &= \frac{1}{4} a_i(\xi) \\
N_i(\xi) &= \frac{1}{8} (a_i(\xi) + b_i(\xi)) \\
N_{i+1}(\xi) &= \frac{1}{8} (b_i(\xi) + c_i(\xi)) \\
N_{i+2}(\xi) &= \frac{1}{4} c_i(\xi),
\end{align*} \]
for \( i = 0, 1, 2, \ldots, N \).

With
\[ \begin{align*}
a_i(\xi) &= (i + 1 - \xi)^2, \\
b_i(\xi) &= (-i + 1 + \xi)(i + 1 - \xi) + (i + 2 - \xi)(\xi - i), \\
c_i(\xi) &= (\xi - i)^2.
\end{align*} \]

For this background grid, the value for the tensor \( D_p \) associated with computing the scaled velocity gradient tensor from Section 2.1.2 is constant over the entire grid, and has value \( D_p = \frac{1}{2} h^2 I \).

References

[1] Bardenhagen, S., Brackbill, J., Sulsly, D.: The material-point method for granular materials. Computer Methods in Applied Mechanics and Engineering 187(3), 529 – 541 (2000). DOI https://doi.org/10.1016/S0045-7825(99)00338-2. URL http://www.sciencedirect.com/science/article/pii/S0045782599003382
[2] Bazilevs, Y., Kamran, K., Moutsanidis, G., Benson, D.J., Oñate, E.: A new formulation for air-blast fluid–structure interaction using an immersed approach. part i: basic methodology and fem-based simulations. Computational Mechanics 60(1), 83–100 (2017)
[3] Bazilevs, Y., Moutsanidis, G., Bueno, J., Kamran, K., Kamensky, D., Hillman, M.C., Gomez, H., Chen, J.S.: A new formulation for air-blast fluid–structure interaction using an immersed approach: part ii—coupling of iga and meshfree discretizations. Computational Mechanics 60(1), 101–116 (2017)
[4] Belytschko, T., Liu, W.K., Moran, B.: Nonlinear Finite Element for Continua and Structures, 1 edn. Wiley (2000)
[5] Belytschko, T., Lu, Y.Y., Gu, L.: Element-free Galerkin methods. International Journal for Numerical Methods in Engineering 37(2), 229–256 (1994)
[6] Belytschko, T., Neal, M.O.: Contact-impact by the pinball algorithm with penalty and lagrangian methods. International Journal for Numerical Methods in Engineering 31(3), 547–572 (1991)
[7] Chen, J.S., Hillman, M., Chi, S.W.: Meshfree methods: Progress made after 20 years. Journal of Engineering Mechanics 143(4), 04017001 (2017)
[8] Chi, S.W., Lee, C.H., Chen, J.S., Guan, P.C.: A level set enhanced natural kernel contact algorithm for impact and penetration modeling. International Journal for Numerical Methods in Engineering 102(3-4), 839–866 (2015)
[9] Fu, C., Guo, Q., Gast, T., Jiang, C., Teran, J.: A polynomial particle-in-cell method. ACM Transactions on Graphics 36(6), 1–12 (2017). DOI 10.1145/3130800.3130878. URL http://dx.doi.org/10.1145/3130800.3130878
[10] Gründemann, H.: Belytschko, t. hughes, t. j. r. (eds.), computational methods for transient analysis. vol. 1 in computational methods in mechanics. amsterdam-new york, north-holland publ. co. 1983. ix, 523 s., us $ 69.00. df1. 225.00. isbn 0-444-86479-2. ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik 66(9), 394–394 (1986). DOI 10.1002/zamm.19860660905. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/zamm.19860660905
[11] Han, X., Gast, T.F., 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), 1–24 (2019)
A momentum preserving frictional contact algorithm based on affine particle-in-cell grid transfers

[12] Harmon, D., Vouga, E., Smith, B., Tamstorf, R., Grinspun, E.: Asynchronous contact mechanics. Communications of the ACM 55(4), 102–109 (2012). DOI 10.1145/2133806.2133828. URL http://dx.doi.org/10.1145/2133806.2133828

[13] Hillman, M., Chen, J.S., Chi, S.W.: Stabilized and variationally consistent nodal integration for meshfree modeling of impact problems. Computational Particle Mechanics 1(3), 245–256 (2014)

[14] Huang, P., Zhang, X., Ma, S., Huang, X.: Contact algorithms for the material point method in impact and penetration simulation. International Journal for Numerical Methods in Engineering 85(4), 498–517 (2011). DOI 10.1002/nme.2981. URL http://onlinelibrary.wiley.com/doi/10.1002/nme.2981

[15] Hughes, T.: The Finite Element Method: Linear Static and Dynamic Finite Element Analysis. Dover Civil and Mechanical Engineering. Dover Publications (2012)

[16] Jiang, C., Schroeder, C., Selle, A., Teran, J., Stomakhin, A.: The affine particle-in-cell method. ACM Trans. Graph. 34(4) (2015)

[17] Jiang, C., Schroeder, C., Teran, J.: An angular momentum conserving affine-particle-in-cell method. Journal of Computational Physics 338, 137 – 164 (2017)

[18] Kamensky, D., Behzadinasab, M., Foster, J.T., Bazilevs, Y.: Peridynamic modeling of frictional contact. Journal of Peridynamics and Nonlocal Modeling (2019)

[19] Kamensky, D., Xu, F., Lee, C.H., Yan, J., Bazilevs, Y., Hsu, M.C.: A contact formulation based on a volumetric potential: Application to isogeometric simulations of atrioventricular valves. Computer Methods in Applied Mechanics and Engineering 330, 522 – 546 (2018)

[20] Karras, T.: Maximizing parallelism in the construction of bvhs, octrees, and k-d trees. In: Proceedings of the Fourth ACM SIGGRAPH / Eurographics Conference on High-Performance Graphics, EGHH-HPG’12, pp. 33–37. Eurographics Association, Goslar, DEU (2012)

[21] Love, E., Sulsky, D.: An unconditionally stable, energy–momentum consistent implementation of the material-point method. Computer Methods in Applied Mechanics and Engineering 195(33-36), 3903–3925 (2006). DOI 10.1016/j.cma.2005.06.027. URL http://dx.doi.org/10.1016/j.cma.2005.06.027

[22] Moutsanidis, G., Kamensky, D., Zhang, D.Z., Bazilevs, Y., Long, C.C.: Modeling strong discontinuities in the material point method using a single velocity field. Computer Methods in Applied Mechanics and Engineering 345, 584 – 601 (2019). DOI https://doi.org/10.1016/j.cma.2018.11.005. URL http://www.sciencedirect.com/science/article/pii/S0045782518305607

[23] Moutsanidis, G., Koester, J.J., Tupek, M.R., Chen, J.S., Bazilevs, Y.: Treatment of near-incompressibility in meshfree and immersed-particle methods. Computational Particle Mechanics 7(2), 309–327 (2019). DOI 10.1007/s40571-019-00238-z. URL http://dx.doi.org/10.1007/s40571-019-00238-z

[24] Nairn, J., Guo, Y.: Material point method calculations with explicit cracks, fracture parameters, and crack propagation. 11th International Conference on Fracture 2005, ICF11 2 (2005)

[25] Nayroles, B., Touzot, G., Villon, P.: Generalizing the finite element method: Diffuse approximation and diffuse elements. Computational Mechanics 10(5), 307–318 (1992)

[26] Puso, M.A., Laursen, T.A.: A 3d contact smoothing method using gregory patches. International Journal for Numerical Methods in Engineering 54(8), 1161–1194 (2002). DOI 10.1002/nme.466. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.466

[27] Puso, M.A., Laursen, T.A.: A mortar segment-to-segment frictional contact method for large deformations. Computer Methods in Applied Mechanics and Engineering 193(45), 4891 – 4913 (2004). DOI https://doi.org/10.1016/j.cma.2004.06.001. URL http://www.sciencedirect.com/science/article/pii/S0045782504002518

[28] Song, J.U., Kim, H.G.: An improved material point method using moving least square shape functions. Computational Particle Mechanics (2020). DOI 10.1007/s40571-020-00368-9. URL https://doi.org/10.1007%2Fs40571-020-00368-9

[29] Sulsky, D., Chen, Z., Schreyer, H.L.: A particle method for history-dependent materials. Computer Methods in Applied Mechanics and Engineering 118(1), 179–196 (1994)

[30] Sulsky, D., Zhou, S.J., Schreyer, H.L.: Application of a particle-in-cell method to solid mechanics. Computer Physics Communications 87(1), 236 – 252 (1995). DOI https://doi.org/10.1016/0010-4655(94)00170-7. URL http://www.sciencedirect.com/science/article/pii/0010465594001707. Particle Simulation Methods

[31] Zhang, H., Wang, D.: Reproducing kernel formulation of B-spline and NURBS basis functions: A meshfree local refinement strategy for isogeometric analysis. Computer Methods in Applied Mechanics and Engineering 320, 474 – 508 (2017)