

Generalized Contact and Improved Frictional Heating in the Material Point Method

J. A. Nairn¹ · S. G. Bardenhagen² · G. D. Smith²

the date of receipt and acceptance should be inserted later

Abstract The material point method (MPM) has proven to be an effective particle method for computational mechanics modeling of problems involving contact, but all prior applications have been limited to Coulomb friction. This paper generalizes the MPM approach for contact to handle any friction law with examples given for friction with adhesion or with a velocity-dependent coefficient of friction. Accounting for adhesion requires an extra calculation to evaluate contact area. Implementation of velocity-dependent laws usually needs numerical methods to find contacting forces. The friction process involves work which can be converted into heat. This paper provides a new method for calculating frictional heating that accounts for interfacial acceleration during the time step. The acceleration terms is small for many problems, but temporal convergence of heating effects for problems involving vibrations and high contact forces is improved by the new method. Fortunately, the new method needs few extra calculations and therefore is recommended for all simulations.

Keywords Material point method · MPM · friction · imperfect interfaces · contact

1 Introduction

One potential advantage of the material point method (MPM) over other computational mechanics tools is for handling problems involving contact. The original development of MPM used a single velocity field on a grid

[17]. Because such a field enforces a continuous velocity field on the particles, single-field MPM automatically prevents particle-particle penetration. But this “contact-for-free” feature can only model contact by stick conditions meaning surfaces move in the same velocity field when in contact but move independently when apart. The contact capabilities of MPM were enhanced by moving to a multimaterial mode MPM where particles are designated as belonging to a material type and each material extrapolates its own velocity field to the grid [2]. Nodes that end up with information from a single material proceed by standard MPM while nodes with more than one material need to resolve and implement contact physics on that node. Subsequent enhancements in MPM contact have included an improved contact algorithm [4], alternate methods for finding the crucial contacting normal vector [10, 13], extension to axisymmetric contact [14], and modeling of imperfect interface laws [13].

Most prior MPM contact modeling has considered only simple Coulomb friction. This paper generalizes the MPM contact methods to allow any contact law and specifically considers friction with adhesion, friction laws with a velocity-dependent coefficient of friction, and implementation of static and dynamic coefficients of friction. The generalizations are mostly straightforward but require some extra calculations and require a subtle modification when implementing velocity-dependent laws. Implementation of some new contact laws were verified by comparison to analytical solutions.

Another motivation for this work was to improve the methods for modeling frictional heating. The target problem was rapid loading of plastic-bonded explosives (PBXs) [1]. These materials are composite materials of energetic particles (such as HMX particles [16]) held together by a plastic binder. During rapid loading,

DISTRIBUTION A. Approved for public release, distribution unlimited. (96TW-2016-0188)

1. Oregon State University, Wood Science & Engineering, 112 Richardson Hall, Corvallis, OR 97330, USA E-mail: John.Nairn@oregonstate.edu

2. Wasatch Molecular, Salt Lake City, UT 84103, USA

it is likely that weak interfaces or cracks in particles will slide as stress waves induce extensive, high force, and rapid sliding. If frictional heating resulting from this sliding causes sufficient temperature increase, the material may develop hot spots that cause detonation [6]. MPM is likely an excellent tool for modeling hot spots and other effects in PBXs [3]. In our preliminary work, however, we noticed that frictional heating calculations had slow temporal convergence requiring a much smaller time step than needed to resolve the specimen's mechanical response. We traced the issue to calculation of frictional heating being equal to frictional force times sliding velocity, which was assumed constant during the time step. We were able to get improved convergence by modifying the frictional heating calculation to account for sliding acceleration and for slip and stick conditions during the time step. The resulting frictional heating needs minimal extra calculations and potentially offers improved convergence for heating effects in PBXs or any other systems experiencing oscillating, high-force contact.

2 Multimaterial Contact

2.1 Multimaterial Mode Extrapolations

In MPM, bodies are discretized into particles on a background grid [17]). Each time step involves extrapolating quantities to the grid, implementing contact mechanics (if applicable), solving the momentum equation, and then updating all needed quantities on the particles. This paper will not review the MPM algorithm (it has appeared in many papers [5]) but instead focuses on multimaterial mode MPM and only those algorithm steps associated with contact mechanics [3].

In multimaterial MPM, each material extrapolates to its own velocity field. The extrapolated quantities needed for contact calculations on node i for material j 's field are [13]:

$$m_{i,j} = \sum_{p \in j} S_{ip} m_p \quad (1)$$

$$\mathbf{p}_{i,j} = \sum_{p \in j} S_{ip} \mathbf{p}_p \quad (2)$$

$$\mathbf{f}_{i,j} = \sum_{p \in j} \left(-m_p \frac{\boldsymbol{\tau}_p^{(n)} \cdot \mathbf{G}_{ip}^{(n)}}{\rho_0} \right) + \mathbf{f}_{i,j}^{ext} \quad (3)$$

$$m_{i,j} \mathbf{x}_{i,j} = \sum_{p \in j} S_{ip} m_p \mathbf{x}_p \quad (4)$$

$$\Omega_{i,j} = \begin{cases} \sum_{p \in j} S_{ip} V_p & 3D \\ \sum_{p \in j} S_{ip} A_p & 2D \end{cases} \quad (5)$$

$$\mathbf{g}_{i,j} = \begin{cases} \sum_{p \in j} \mathbf{G}_{ip} V_p & 3D \\ \sum_{p \in j} \mathbf{G}_{ip} A_p & 2D \end{cases} \quad (6)$$

Here $m_{i,j}$, $\mathbf{p}_{i,j}$, and $\mathbf{f}_{i,j}$ are material j 's mass, momentum, and force. The sums over particles are only for particles of material type j . Prior modeling on contact has depended only on $m_{i,j}$ and $\mathbf{p}_{i,j}$ [2, 4], but the new frictional heating method described below needs $\mathbf{f}_{i,j}$ as well. $\boldsymbol{\tau}_p^{(n)}$ is Kirchoff stress on the particles and $\mathbf{f}_{i,j}^{ext}$ are external forces due to body forces (*e.g.*, gravity) or boundary conditions. The extrapolated position, $\mathbf{x}_{i,j}$, domain, $\Omega_{i,j}$, and domain gradient, $\mathbf{g}_{i,j}$ are specific to contact calculations and are used to improve contact detection and to find the contacting normal vector and contact area [13]. Finally, S_{ip} and \mathbf{G}_{ip} are generalized shape functions and shape function gradients commonly used in MPM [5].

2.2 Contact Mechanics

Once all quantities are extrapolated to the grid, each node is examined to see if it has only one material or more than one material. All nodes with a single material proceed by standard MPM. All nodes with more than one material must determine if the materials are in contact. If they are not in contact, they proceed independently (*i.e.*, they are not interacting). If contact is detected, however, the extrapolated quantities (namely momenta and forces) must be changed to implement the modeled contact mechanics. Note that multimaterial contact in MPM handles contact between different materials (although two contacting objects can be different "materials" with the same properties), but does not handle self contact. Although MPM automatically handles self contact, such contact follows a stick contact law. Implementation of self contact with other contact laws would require alternate methods [8].

2.3 Finding Contacting Normal Vector

We first describe contact between exactly two, non-rigid materials — material a and b as illustrated in Fig. 1 (the changes for more than two materials and for rigid materials are given later). The first task is to find the normal vector at the contacting surface. This task was initially done by finding mass gradient of the material [2], but this approach could give inaccurate results. In our experience, the most common reason MPM fails in problems involving contact is that the normal is inaccurate. This observation means that contact calculations need sufficient resolution to resolve normals and may need to use different methods for different problems. The issue is discussed thoroughly in Ref. [13] and summarized here. The normal vector calculated from material a information alone is found from volume gradient (in 3D) or area

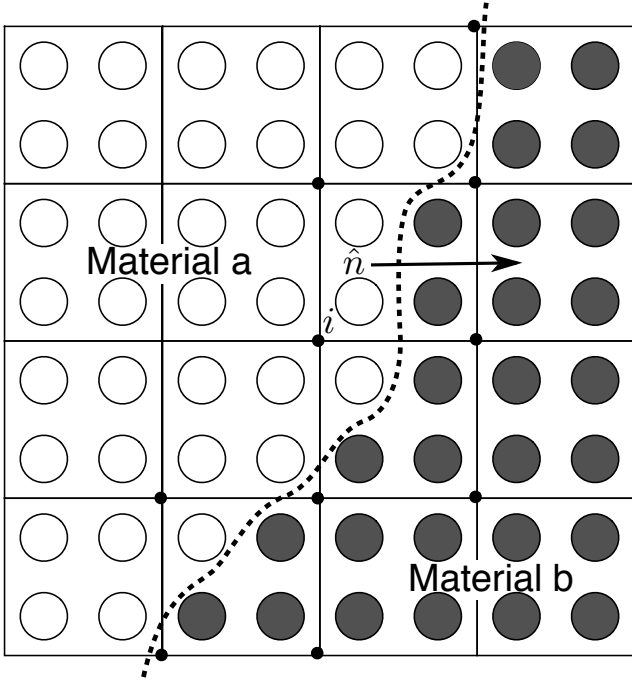


Fig. 1 Material points in a 2D MPM model having two different material types on either side of a contact surface. The normal vector \hat{n} is defined as directed from material *a* to material *b*. The black dots indicate nodes (such as node *i*) that have multimaterial fields.

gradient (in 2D):

$$\|\mathbf{n}_{i,a}\| \mathbf{n}_{i,a} = \mathbf{g}_{i,a} \quad (7)$$

In axisymmetric MPM, the gradient is a spatial term that must be determined from particle *area* in the *r-z* plane and not the particle volume [14]. Furthermore, to deal with some edge effects, any edge that is known to be on a symmetry plane (such as the $r = 0$ plane in axisymmetric MPM), should set the component of the normal perpendicular to that plane to be zero.

For two materials, the contacting normal can be found from either material *a* or *b*, but in general they will be different. When each material was treated independently with their own normals, the end result was non-conservation of momentum. For two materials in contact, the solution is to find a single normal vector. The two most useful approaches are [13]:

Maximum Volume Gradient (MVG): find the normal from material *a* or *b*, whichever has the domain gradient with the largest magnitude.

Average Volume Gradient (AVG): find the normal from a domain-weighted average of gradients from the two materials — $(\Omega_{i,a}\mathbf{g}_{i,a} - \Omega_{i,b}\mathbf{g}_{i,b})/(\Omega_a + \Omega_b)$.

A third, and the most accurate option when available, is to *specify the normal* (SN). This option is typically only available to simple problems (and is used below in several friction examples).

2.4 Detecting Contact

Once the normal vector, \hat{n} , is found, the next task is to detect if the two materials are separated or in contact. The initial MPM method was to assume contact occurs whenever $\Delta \mathbf{v}_i \cdot \hat{n} < 0$, which as shown below corresponds to the surface stress being in compression. This method alone, however, often leads to premature detection of contact. The contact calculations are improved by also looking at material positions and calculating separation of material surfaces using

$$\delta_n = (\mathbf{x}_{i,a} - \mathbf{x}_{i,b}) \cdot \hat{n} - 0.8h_\perp \quad (8)$$

where h_\perp is an effective thickness of the contacting volume perpendicular to the contacting surface [13]. The subtraction of $0.8h_\perp$ accounts for the spatial effect that particle surfaces come into contact before the centers (which are used to calculate $\mathbf{x}_{i,j}$). For a regular grid with equal element dimension in all directions, h_\perp is equal to element side Δx . The modifications when using rectangular elements with different dimensions along different axes are given in Ref. [13]. The extension to a non-regular grid would require more modifications.

A useful option in MPM problems with modest deformation and when all interfaces that come into contact start out in contact (a situation common, for example, in composite materials), is to change the separation calculation to use particle displacements. In this approach, the position extrapolation is changed to a displacement extrapolation:

$$m_{i,j} \Delta \mathbf{x}_{i,j} = \sum_{p \in j} S_{ip} m_p (\mathbf{x}_p - \mathbf{x}_p^{(0)}) \quad (9)$$

where $\mathbf{x}_p^{(0)}$ is the initial particle position. The surface separation is then found from extrapolated displacements

$$\delta_n = (\Delta \mathbf{x}_{i,a} - \Delta \mathbf{x}_{i,b}) \cdot \hat{n} \quad (10)$$

and note that the spatial correction, h_\perp , is not applied when using displacements.

Finally, the surfaces are assumed to be in contact when both $\Delta \mathbf{v}_i \cdot \hat{n} < 0$ and $\delta_n < 0$, otherwise the surfaces are separated.

2.5 Finding Contact Area

Although many friction laws can be implemented in MPM without needing to know the contact area, those including strength terms depend on that area. The area calculation needs to account for orientation of the contact normal and the total number of nodes along an interface — that number of nodes will be different for

an interface near grid lines compared to an interface in the middle of grid cells. An area calculation was developed in a previous paper [13]. The contact area at node i for 2D MPM in terms of extrapolated quantities defined above is:

$$A_c = \frac{t\sqrt{2\Omega_i \min(\Omega_{i,a}, \Omega_{i,b})}}{h_\perp} \quad (11)$$

where t is particle thickness and h_\perp accounts for tilted contact surfaces in grid with different element dimensions in the axis directions. In 3D problems t is dropped. In axisymmetric MPM [14], the t is replaced by r_i , which is the radial position of the node to give area per radian.

2.6 Implementing Contact Mechanics

When contact surfaces are determined to be in contact, the final task is to modify nodal momenta and forces to reflect the implemented contact mechanics. The first step in this task is to calculate the momentum change applied to material a that would be required for that material to move in the center of mass velocity ($\mathbf{v}_i^{(c)}$):

$$\Delta \mathbf{p}_{i,a} = m_{i,a} \mathbf{v}_i^{(c)} - \mathbf{p}_{i,a} = \frac{m_{i,a} \mathbf{p}_{i,b} - m_{i,b} \mathbf{p}_{i,a}}{m_{i,a} + m_{i,b}} \quad (12)$$

where

$$\mathbf{v}_i^{(c)} = \frac{\sum_j \mathbf{p}_{i,j}}{m_i^{(c)}} = \frac{\sum_j m_{i,j} \mathbf{v}_{i,j}}{m_i^{(c)}} \quad (13)$$

and $m_i^{(c)} = \sum_j m_{i,j}$ is the total nodal mass. In other words, if $\Delta \mathbf{p}_{i,a}$ was applied to material a and $-\Delta \mathbf{p}_{i,a}$ applied to material b , the two materials would stick together, or move in a single velocity field as if they were one material. This “stick” calculation is equivalent MPM treatment of self contact, but multimaterial MPM extrapolations provide information that allows implementation of new contact conditions.

The difference in unmodified velocities for materials a and b can be related to this momentum change:

$$\Delta \mathbf{v}_i = \mathbf{v}_{i,b} - \mathbf{v}_{i,a} = \frac{\mathbf{p}_{i,b}}{m_{i,b}} - \frac{\mathbf{p}_{i,a}}{m_{i,a}} = \frac{\Delta \mathbf{p}_{i,a}}{m_{i,red}} \quad (14)$$

where $m_{i,red} = m_{i,a}m_{i,b}/(m_{i,a} + m_{i,b})$ is reduced mass at node i . The momentum change can be interpreted as apparent contacting forces by rewriting $\Delta \mathbf{p}_{i,a}$ as

$$\begin{aligned} \Delta \mathbf{p}_{i,a} &= (\Delta \mathbf{p}_{i,a} \cdot \hat{\mathbf{n}}) \hat{\mathbf{n}} + (\Delta \mathbf{p}_{i,a} \cdot \hat{\mathbf{t}}) \hat{\mathbf{t}} = d_n \hat{\mathbf{n}} + d_t \hat{\mathbf{t}} \\ &= (-NA_c \Delta t) \hat{\mathbf{n}} + (S_{stick} A_c \Delta t) \hat{\mathbf{t}} \end{aligned} \quad (15)$$

where N is the contacting normal pressure and S_{stick} is the sliding traction for preventing frictional sliding (or d_n and d_t are the normal and tangential momentum

changes required to stick in that direction). Notice that contact detection requires $\Delta \mathbf{v}_i \cdot \hat{\mathbf{n}} = -NA_c \Delta t / m_{i,red} < 0$, which guarantees that all contact situations have compression on the surface with $N > 0$.

To implement friction, we calculate the tangential traction that would be experienced by frictional sliding:

$$S_{slide} = f(N, A_c, \Delta \mathbf{v}_i', \dots) \quad (16)$$

where $f(N, A_c, \Delta \mathbf{v}_i', \dots)$ is any arbitrary friction law which may depend on various parameters such as normal pressure, contact area, relative sliding velocity, and maybe more parameters. Notice that realistic velocity-dependent friction laws will depend on the final sliding velocity $\Delta \mathbf{v}_i'$, and not the initial unmodified velocity difference $\Delta \mathbf{v}_i$ (the consequences of this difference are discussed later). The full frictional algorithm becomes:

1. If $S_{stick} < S_{slide}$, then the driving forces for frictional sliding are too small to overcome sticking meaning the two materials stick rather than slide. Set final momentum change to $\Delta \mathbf{p}_{i,a}' = \Delta \mathbf{p}_{i,a}$.
2. But, if $S_{stick} > S_{slide}$, the surface would rather slide at the lower S_{slide} then stick. Set final momentum change to

$$\Delta \mathbf{p}_{i,a}' = d_n \hat{\mathbf{n}} + (S_{slide} A_c \Delta t) \hat{\mathbf{t}} \quad (17)$$

3. Change the momenta for materials a and b to

$$\mathbf{p}_{i,a}' = \mathbf{p}_{i,a} + \Delta \mathbf{p}_{i,a}' \text{ and } \mathbf{p}_{i,b}' = \mathbf{p}_{i,b} - \Delta \mathbf{p}_{i,a}' \quad (18)$$

4. To keep forces consistent with changed momenta in contact calculations done after updating nodal momenta, the nodal forces should be changed to:

$$\mathbf{f}_{i,a}' = \mathbf{f}_{i,a} + \frac{\Delta \mathbf{p}_{i,a}'}{\Delta t} \text{ and } \mathbf{f}_{i,b}' = \mathbf{f}_{i,b} - \frac{\Delta \mathbf{p}_{i,a}'}{\Delta t} \quad (19)$$

Note that some papers claim improved contact by working with forces rather than momenta and show better simulation results compared to the momentum approach generalized here [9, 15]. Examination of the physics, however, shows that the effects of contact appear in *both* momenta and forces and can be implemented through either provided all is done self-consistently throughout the MPM time step. We suggest their differences in methods were caused by improper implementation of the momentum approach rather than some advantage of using forces.

2.7 Coulomb Friction

For simple Coulomb friction, $S = \mu N$ where μ is the coefficient of friction, which can be written as $S_{slide} A_c \Delta t$

$= -\mu d_n$. During frictional sliding, the momentum change becomes

$$\Delta \mathbf{p}'_{i,a} = d_n(\hat{\mathbf{n}} - \mu \hat{\mathbf{t}}) \quad (20)$$

This equation is identical to previous implementations of friction in MPM [4, 13]. Notice that the law is independent of the contact area and the sliding velocity.

2.8 Velocity-Dependent Friction

To implement velocity-dependent friction, we need to calculate $\Delta \mathbf{v}'_i$ in the final sliding condition

$$\begin{aligned} \Delta \mathbf{v}'_i &= \frac{\mathbf{p}'_{i,b}}{m_{i,b}} - \frac{\mathbf{p}'_{i,a}}{m_{i,a}} = \frac{\Delta \mathbf{p}_{i,a} - \Delta \mathbf{p}'_{i,a}}{m_{i,red}} \\ &= \frac{(d_t - S_{slide} A_c \Delta t)}{m_{i,red}} \hat{\mathbf{t}} \end{aligned} \quad (21)$$

Because $\Delta \mathbf{v}'_i$ depends on S_{slide} , any function for S_{slide} will need to be inverted to find sliding traction. For example, if the coefficient of friction is linearly dependent on the sliding velocity ($\mu = \mu_0 + k|\Delta \mathbf{v}'_i|$) then:

$$S_{slide} = \left(\mu_0 + \frac{k(d_t - S_{slide} A_c \Delta t)}{m_{i,red}} \right) N \quad (22)$$

which can be solved to give

$$S_{slide} = \frac{m_{i,red} \mu_0 + k d_t}{m_{i,red} + k d_n} N \quad (23)$$

By finding $S_{slide} A_c \Delta t$ needed for momentum change (see Eq. (17)), which only depends on $d_n = -N A_c \Delta t$ and d_t , this velocity-dependent Coulomb's law is independent of the contact area.

A seemingly simple velocity-dependent effect to implement is static and dynamic friction. The calculation is to find $S_{slide} = \mu_s N$ where μ_s is the static coefficient of friction. If S_{stick} is less than this S_{slide} , the surfaces stick together; otherwise S_{slide} is changed to $S_{slide} = \mu_d N$ where μ_d is the dynamic coefficient of friction (or to some other dynamic friction law) and then make the changes needed for frictional sliding. The Results and Discussion section finds inaccuracies for this implementation and suggests an alternative.

Finally, the above velocity-dependent laws are among the few that yield to close-formed expressions for S_{slide} . For friction laws that have non-linear or arbitrary dependence on velocity, the analogous expression to Eq. (22) will usually require numerical methods to solve for S_{slide} .

2.9 Contact Law with Adhesion

Sometimes friction modeling adds an adhesion term to Coulomb friction or $S = S_a + \mu N$ where S_a is a shear adhesion strength of the interface [19]. In this law the material will stick unless S_{stick} exceeds the adhesion strength, and then will slide with an offset Coulomb law (this component could depend on velocity if desired). The sliding force needed to implement this law is

$$S_{slide} A_c \Delta t = S_a A_c \Delta t - \mu d_n \quad (24)$$

The main difference from previous laws is that for non-zero S_a the final $S_{slide} A_c \Delta t$ calculation depends on contact area A_c .

If the interface has shear adhesion strength, it might have tensile adhesion strength, N_a , as well. Such a law is easily implemented, but requires extra calculation when surfaces are detected as being separated ($N < 0$ or $\delta_n > 0$). Under simple friction, such interfaces move freely, but when adhesion is present, another criterion is needed to decide if they should continue to stick or should move freely. One option is an elliptical failure criterion:

$$\left(\frac{S_{stick}}{S_a} \right)^2 + \left(\frac{N}{N_a} \right)^2 > 1 \quad (25)$$

If this criterion is met, the interfacial adhesion has broken and surfaces move freely; if it is not met, the surfaces stick and move in the center-of-mass velocity field. When surfaces are in contact, they proceed as normal (e.g., using Eq. (24) and N_a has no effect.)

Notice that unless modified, the MPM adhesive contact methods described here cannot model history-dependent adhesion. In other words it cannot model contact that sticks until adhesion is broken and thereafter slides with some frictional sliding law. The problem is that contacting nodes are calculated on each time step and extrapolated nodal quantities contain no information about whether the contact on the previous time step overcame adhesion (or even if the contact on previous time step was for the same two materials). As a consequence, adhesion contact described above is reversible. If the surfaces overcome adhesion they will slide, but if the contact forces drop, they will re-adhere. This behavior is realistic for some contact, such as interfaces with interaction forces, but may not be appropriate for others, such as breaking of bonds at interfaces.

2.10 Frictional Heating

Whenever friction is modeled, it might be important to model heating caused by the friction. The total work

done by friction forces on node i in one time step is

$$\Delta W_{i,F} = \int_0^{\Delta t} \mathbf{F}(t) \cdot \Delta \mathbf{v}'_i(t) dt \quad (26)$$

where $\mathbf{F}(t)$ is the frictional sliding force and $\Delta \mathbf{v}'_i(t)$ is the *final* relative sliding velocity. For first order heating calculation, we assume $\mathbf{F}(t) = S_{slide} A_c \hat{\mathbf{t}}$ and $\Delta \mathbf{v}'_i(t) = (d_t - S_{slide} A_c \Delta t) \hat{\mathbf{t}} / m_{i,red}$ (as derived above) and both are constant over the time step. The frictional work is

$$\Delta W_{i,F} = W_1 = \frac{S_{slide} A_c \Delta t}{m_{i,red}} (d_t - S_{slide} A_c \Delta t) \quad (27)$$

Notice that $\Delta W_{i,F}$ is always greater than zero because d_t is always greater than $S_{slide} A_c \Delta t$ during sliding. This expression involves only $S_{slide} A_c \Delta t = \Delta \mathbf{p}'_{i,a} \cdot \hat{\mathbf{t}}$ and d_t . Therefore, whenever the frictional law is independent of A_c , the frictional work result is as well.

In an attempt to improve convergence of the frictional heating term, we added an acceleration term to account for changes in sliding velocity during the time step. The frictional force becomes

$$\mathbf{F}(t) = S_{slide} (\Delta \mathbf{v}'_i(t)) A_c \hat{\mathbf{t}}$$

and the relative sliding velocity becomes

$$\Delta \mathbf{v}'_i(t) = (\Delta \mathbf{v}'_i \cdot \hat{\mathbf{t}} - \Delta \mathbf{a}'_i(t - \Delta t) \cdot \hat{\mathbf{t}}) \hat{\mathbf{t}} \quad (28)$$

where $\Delta \mathbf{a}'_i$ is the relative acceleration of the sliding surfaces, which can be found from

$$\Delta \mathbf{a}'_i = \frac{\mathbf{f}'_{i,b}}{m_{i,b}} - \frac{\mathbf{f}'_{i,a}}{m_{i,a}} = \frac{\Delta \mathbf{f}_{i,a} - \frac{\Delta \mathbf{p}'_{i,a}}{\Delta t}}{m_{i,red}} \quad (29)$$

where

$$\Delta \mathbf{f}_{i,a} = \frac{m_{i,a} \mathbf{f}_{i,b} - m_{i,b} \mathbf{f}_{i,a}}{m_{i,a} + m_{i,b}} \quad (30)$$

If we ignore velocity dependence of S_{slide} (which is correct for velocity-independent friction laws such as Coulomb friction and would be difficult to implement when S_{slide} requires a numerical solution in other laws) and assume the sliding direction is constant (i.e., $\hat{\mathbf{t}}$ remains constant), the friction work is:

$$\Delta W_{i,F} = \int_{t_s}^{\Delta t} S_{slide} A_c \left[\frac{(d_t - S_{slide} A_c \Delta t)}{m_{i,red}} + \frac{(f_t - S_{slide} A_c)(t - \Delta t)}{m_{i,red}} \right] dt \quad (31)$$

where t_s is the time when sliding starts in the time step and $f_t = \Delta \mathbf{f}_{i,a} \cdot \hat{\mathbf{t}}$. $\Delta W_{i,F}$ is only calculated when surfaces are determined to be sliding at the end of the time step, but the sliding may have started during the

time step. Extrapolating to the start of the time step, the relative velocity in the sliding direction was

$$|\mathbf{v}'_i(0)| = \frac{d_t - f_t \Delta t}{m_{i,red}} = W_1 - W_2 \quad (32)$$

where

$$W_2 = \frac{S_{slide} A_c \Delta t}{m_{i,red}} (f_t - S_{slide} A_c) \Delta t \quad (33)$$

Thus if $W_2 < W_1$, the velocity at the start of the time step was positive, which means the full integral is used ($t_s = 0$); it evaluates to

$$\Delta W_{i,F} = W_1 - \frac{W_2}{2} \quad (34)$$

The first term recovers “first order” frictional work (see Eq. (31)) while the second adds acceleration for “second order” frictional work. If $W_2 > W_1$, however, it means the sliding started in this time step. Solving for the time when relative velocity is zero can be cast as

$$t_s = \Delta t \left(1 - \frac{W_1}{W_2} \right) \quad (35)$$

Integrating from t_s to Δt , frictional work evaluates to

$$\Delta W_{i,F} = \frac{W_1^2}{2W_2} \quad (36)$$

Notice that addition of acceleration effects involves only the minor extra calculations to find f_t followed by substitution into Eq. (34) or (36). This information is available in any MPM code implementing contact.

As shown below, this new “second order” heating analysis improves convergence of frictional heating calculations, but it does not provide second-order temporal convergence of frictional work results. The integrands of heating integrals come from standard MPM calculations. Because MPM calculations have less than quadratic temporal convergence, frictional work will be less too. If future MPM methods add quadratic convergence, the “second order” method here should extend that convergence to frictional work as well.

2.11 Injecting Heat to MPM Calculations

Friction heating is calculated on the nodes and therefore will not affect MPM particles unless those nodes communicate to temperature on those particles. An appropriate way to inject heat is to run MPM calculations with coupled conduction. Although few papers discuss coupling heat conduction to MPM calculations, the task

is easy and available in most MPM codes. One brief account shows that a generalized MPM conduction analysis reduces to solving the following equation on the nodes [14]:

$$M_i \frac{dT_i}{dt} = f_i^{(cond)} \quad (37)$$

where M_i is nodal thermal mass and T_i is nodal temperature extrapolated to the nodes weighted by the thermal mass:

$$M_i = \sum_p m_p C_v^{(p)} S_{ip}, \quad T_i = \frac{1}{M_i} \sum_p m_p C_v^{(p)} T_p S_{ip} \quad (38)$$

where $C_v^{(p)}$ is particle heat capacity. In ordinary calculations, the conduction force, $f_i^{(cond)}$ would include heat conduction (due to thermal gradients), heat sources, and boundary heat fluxes. Frictional heating is implemented simply by adding $\Delta W_{i,F}/\Delta t$ to $f_i^{(cond)}$ for any node with frictional contact. The heating is added once per contact node and not once per material velocity field.

2.12 Other Issues

The above analysis assumes only two materials contact at each node and MPM works best in such situations. If more than two materials interact with the same node, it becomes infeasible to resolve all possible contact situations needed to correctly find contacting normals at each surface affecting that node. Because three or more materials on a single node are inevitable in large, multimaterial simulations, it is important to have an approximate algorithm to handle them. A reasonable approach was proposed in Ref. [13]:

1. For each node with more than two materials, loop over each material on that node.
2. For each material a , lump all other materials into a virtual second material, b , with lumped mass, momentum, and any other quantities needed for contact calculations.
3. Use the previous two-material MPM contact methods to find $\Delta p'_{i,a}$ but use that result only to change momentum and force for material a .
4. Add frictional heating for material a , which is discussed next.

A complication appears for frictional heating, which is added once for a contacting pair. By noting that for a multimaterial node, the reduced mass can be written as:

$$\frac{1}{m_{i,red}} = \sum_j \frac{1}{m_{i,j}} \quad (39)$$

the total frictional work can be revised to be

$$\Delta W_{i,F} = \sum_j \frac{m_{i,red} \Delta W_{i,F}^{(j)}}{m_{i,j}} \quad (40)$$

where $\Delta W_{i,F}^{(j)}$ is friction heating calculated for material j contacting the other lumped materials. Thus total frictional heating is converted to a sum over each material with summand added in step 4 above. This approach reduces to exact results for two contacting materials because $\Delta W_{i,F}^{(a)} = \Delta W_{i,F}^{(b)}$. For more than two materials, it gives a reasonable approximation to total frictional heat.

The extension to contact of material a with a rigid material b is also given in Ref. [13]. In brief, because rigid materials effectively have infinite mass, the contact calculations will reduce to $\mathbf{v}_i^{(c)} = \mathbf{v}_{i,b}$ and $m_{i,red} = m_{i,a}$; the rest of the analysis, including addition of frictional heating, is the same.

A related contact problem in MPM is contact between crack surfaces. In MPM, explicit cracks can be modeled using the CRAMP algorithm [11] for **CRACKS** in the **MPM**. In this algorithm, each node near a crack is divided into separate velocity fields corresponding to particles above the crack and particles below the crack. Like multimaterial mode MPM, these nodes can implement contact or imperfect interfaces on crack surfaces [12]. The implementation details are essentially identical to the methods given here and therefore not repeated. The only significant difference is finding the contacting normal. The CRAMP algorithm includes a discretization of the crack path (in 2D) or crack surface (in 3D [7]) and this geometry can be used to find normal vector for contacting surfaces. Because of this difference, contact between crack surfaces can be more accurate than contact between two materials. For example, an alternative to modeling multimaterial contacting problems is to use single-material MPM but insert crack surfaces between all contact surfaces. This approach, which is analogous to finite element method of pre-defining master and slave surfaces, is limited to relatively simple problems with prior knowledge of all contact situations. Contact in multimaterial MPM does not have this limitation.

3 Results and Discussion

For validation, we selected the non-trivial (but still simple) sliding block problem shown in Fig. 2. The normal force is ramped up to a constant normal stress of $N = F_n/(lb)$ from 0 to t_1 (which was done to minimize dynamic effects). The tangential force is then ramped from t_2 to t_f until tangential contact stress,

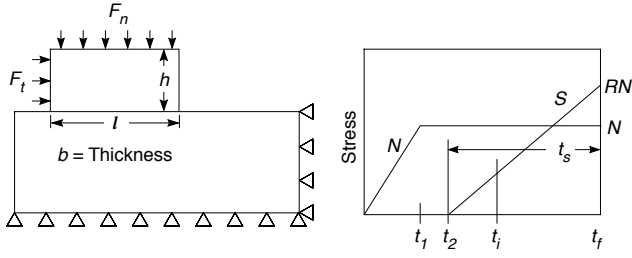


Fig. 2 The sliding block problem. The dimensions used were $l = 40$ mm, $h = 15$ mm, and $b = 1$ mm. The supporting base was 140×25 mm (and supported by fixed displacement boundary conditions), the sliding block was offset 10 mm from base's edge, and the background grid used 5 mm cells.

$S = F_t/(lb)$, reaches R times the normal stress. F_t and F_n were spread out equally as point loads to the particles along the respective edges. We assumed a friction law of:

$$S_{slide} = (\mu_d + k v(t))N + S_a \quad (41)$$

where $\mu_d = 0.2$ is the dynamic coefficient of friction, $k v(t)$ implements a velocity dependent coefficient of friction, and S_a is an adhesion term. A rigid-body, analytical solution for both block velocity and for total friction work is in the appendix. The MPM model, however, used nonrigid materials with $E = 2500$ MPa, $\nu = 0.33$, $\rho = 1$ g/cm³, and $C_v = 200$ J/(kg-K). All calculations used $t_2 = t_1 = 0.25t_f$ and $R = 1$. The total normal force, F_n , was adjusted such that the maximum velocity, v_{max} , at the end of an 85 mm sliding zone would always be less than 2% of the material's tensile wave speed (or $v_{max} = 31.6$ m/sec) for contact modeled by Coulomb friction only ($k = S_a = 0$). The same conditions were then used for other contact laws as well to enhance comparisons. Finally, this simple multimaterial MPM model had only one interface, which allowed specifying the contact normal as $\hat{n} = (0, 1)$. Specifying the normal allowed us to validate friction laws without concern about whether or not multimaterial MPM was correctly finding the normal vector.

Figure 3 compares MPM results for average velocity of the sliding block particles compared to solution for a sliding rigid block using three different friction laws. The three friction laws are simple Coulomb friction (with $\mu = 0.2$), friction when the coefficient of friction depends linearly on velocity (with $\mu = 0.2$ and $k v_{max} = 0.5$), and friction with adhesion (with $\mu = 0.2$ and $S_a = 0.3N$). The solid lines are MPM results and the dotted lines are the rigid body solutions (given in the appendix). For all friction laws, the MPM results agree nearly exactly with the rigid body solution.

The simulations in Fig. 3 used a regular, rectangular grid with the horizontal surface along a grid line.

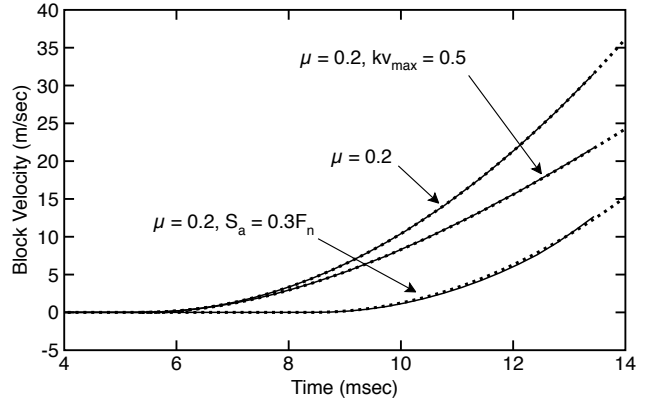


Fig. 3 Average velocity of the particles in the sliding block as a function of time for three different friction laws. The solid lines are MPM results and the dotted lines are theoretical results for sliding of a rigid block.

To test if our algorithm can handle angled surfaces, we re-ran the Coulomb friction example for surfaces from 0° to 45° and the results were essentially identical. We conclude that MPM can handle contact on arbitrary surfaces. Needless to say, modeling of contact on curved surfaces, must have a background ground that can resolve curvatures and normals of the contacting surfaces. We were able to model all straight, angled surfaces using the same 5 mm cells that worked for horizontal surfaces in Fig. 3.

Using simple Coulomb friction with a higher static coefficient of friction ($\mu_d = 0.4$ and $\mu_s = 0.2$) did not agree with rigid body theory because sliding started earlier than predicted by static conditions. The simulated onset of sliding was not improved by changing resolution, time step, or sliding rate. The question remains — why does the proposed MPM method for friction handle coefficient of friction that increases linearly with velocity but cannot handle one that drops instantly from static to dynamic conditions? The linearly-increasing function is smooth (leading to continuous sliding acceleration) and the coefficient of friction increases as velocity increases. In contrast, the static/dynamic law has a discontinuity in coefficient of friction (leading to discontinuous sliding acceleration) and the coefficient of friction decreases with velocity. To test which of these two differences causes the simple implementation to fail, we implemented a new law having a “smooth” transition from μ_s to μ_d :

$$S_{slide} = \frac{\mu_s v_{1/2} + \mu_d |\Delta \mathbf{v}'_i|}{v_{1/2} + |\Delta \mathbf{v}'_i|} N \quad (42)$$

By this law, the coefficient of friction smoothly transitions from μ_s at zero sliding velocity to μ_d at high sliding velocity. It is half way from μ_s to μ_d at sliding velocity of $v_{1/2}$, which is an adjustable parameter. The

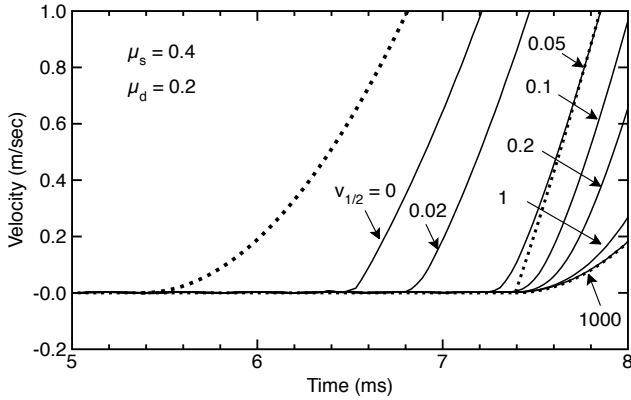


Fig. 4 Average velocity of the particles in the sliding block as a function of time for a smooth transition from $\mu_s = 0.4$ to $\mu_d = 0.2$ using the law in Eq. (42) as a function of $v_{1/2}$ (in m/sec). The solid lines are MPM results. The three dotted lines are theoretical results for sliding of a rigid block for $\mu_s = \mu_d = 0.2$, $\mu_s = 0.4$ and $\mu_d = 0.2$, and $\mu_s = \mu_d = 0.4$, respectively (from left to right).

reason for choosing this law is because S_{slide} can be found analytically after substituting into Eq. (21) for velocity to get:

$$\frac{S_{slide}}{N} = \frac{\frac{D^*}{N^*} + \mu_d}{2} - \sqrt{\left(\frac{\frac{D^*}{N^*} - \mu_d}{2}\right)^2 - \frac{(\mu_s - \mu_d)k^*}{N^*}} \quad (43)$$

where $D^* = d_t + v_{1/2}m_{i,red}$, $N^* = NA_c\Delta t$, and $k^* = v_{1/2}m_{i,red}$. This law is applied only when $d_t > -\mu_s d_n$, otherwise the surfaces stick due to static friction.

Simulation results as a function of $v_{1/2}$ for the law in Eq. (42) are given in Fig. 4. Besides having two coefficients of friction, $\mu_s = 0.4$ and $\mu_d = 0.2$, all other parameters are the same as for Fig. 2. The three dotted lines are rigid-body solutions. The one in the middle is the expected rigid sliding result with static and dynamic coefficients of friction. Notice the kink in the velocity curve due to discontinuity in sliding acceleration at the onset of sliding. The dotted lines on the left and right edges are simple Coulomb friction laws for $\mu_s = \mu_d = 0.2$ or 0.4 , respectively, and given for reference. The solid lines are MPM simulation results. For the smooth transition law in Eq. (42), the simulation correctly matches the $\mu_s = \mu_d = 0.4$ as $v_{1/2}$ becomes large (≥ 1000 m/sec). As $v_{1/2}$ is reduced, the results transition toward the static/dynamic predictions and matches them very well for $v_{1/2} = 0.05$ m/sec. If $v_{1/2}$ is made too small, however, the sliding starts earlier than expected. After starting, the sliding moves with the predicated dynamic coefficient of friction, but an offset between simulation and prediction persists. Apparently, MPM can handle a decreasing coefficient of friction, but

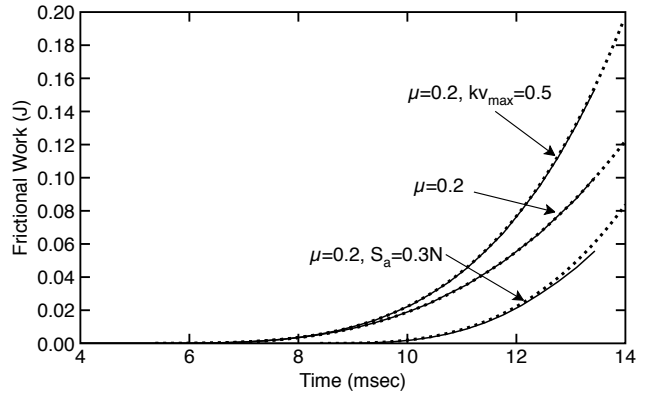


Fig. 5 Total work of friction in the sliding block problem as a function of time for three different friction laws. The solid lines are MPM results and the dotted lines are theoretical result for sliding of a rigid block.

if it changes too abruptly (*e.g.*, $v_{1/2} < 0.05$ m/sec) the simulation results may give different results for onset of sliding than expected by simple stick conditions. We speculate that one or a few nodes commence sliding early due to dynamic fluctuations in forces. If the law has an abrupt change in coefficient of friction, the resulting discontinuity in acceleration acts as an impact pulse that causes neighboring nodes to commence sliding as well. Once most nodes are sliding, the simulation proceeds with the dynamic coefficient of friction. The only solution is to limit the rate of change in coefficient of friction using a sufficiently smooth function. Although the law in Eq. (42) is one potential option, it is likely that the choice of $v_{1/2} = 0.05$ m/sec may need to change for other problems.

Figure 5 gives the results for the total work of friction (*i.e.*, frictional heating) corresponding to the sliding results in Fig. 3. The dotted lines are predictions given by equations in the appendix; the solid lines are MPM results. The MPM results and predictions agree well. When MPM simulations were run using either first order or second order analysis, the results were indistinguishable. These simple friction laws are continuously sliding with acceleration that is monotonically increasing with time. These specific conditions apparently led to well-controlled simulation with $f_t \approx S_{slide}A_c$ and therefore $W_2 \ll W_1$. As a result, the inclusion of second order acceleration term in heating effects was negligible.

Although second-order frictional work calculations were negligible for the simple problem in Fig. 2, the term becomes more important under conditions of non-continuous sliding, acceleration that changes sign, and high contact forces. These conditions describe the rubbing problem in Fig. 6. Here the block was loaded slowly with F_n until stress reached 100 MPa and then the top 5/6 of the left and right edges were moved by rigid

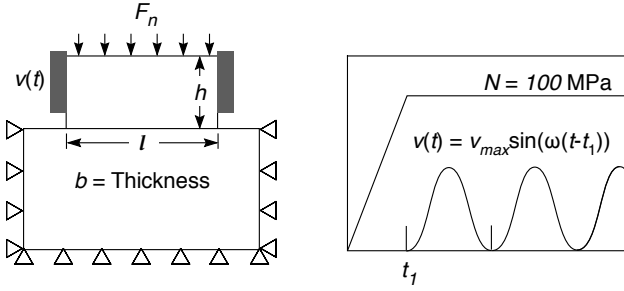


Fig. 6 The oscillating block problem. The dimensions used were $l = 40$ mm, $h = 15$ mm, and $b = 1$ mm. The supporting base was 60×25 mm (and supported by fixed displacement boundary conditions), the sliding block was offset 10 mm from base's edge, and the background grid used 2.5 mm cells.

blocks moving with sinusoidal velocity

$$v(t) = v_{max} \sin \omega(t - t_1) \quad \text{for } t > t_1 \quad (44)$$

where $\omega = 2v_{max}/x_{max}$. With this velocity, the block is rubbing with an amplitude of x_{max} reaching a maximum velocity of v_{max} . We set $x_{max} = 0.1$ mm and v_{max} to 0.01% of the material's wave speed. For the same material properties used in Fig. 2, these resulted in $v_{max} = 0.158$ m/sec and period of the oscillation of $2\pi/\omega = 1.987$ msec. The contact law was simple Coulomb friction with $\mu = 0.5$.

This oscillating block problem starts in stick conditions. Once the velocity increases enough, it starts to slide, but when the velocity drops again, it will re-enter stick conditions. We simulated one stick-slide-stick interval. During the sliding phase, frictional heating occurs and we monitored the total heat generated by that sliding. In explicit computational mechanics, the Courant-Friedrichs-Lewy condition (CFL condition) is a necessary condition for convergence. It states that the time step must satisfy:

$$\Delta t = C \frac{\Delta x}{v} \quad (45)$$

where Δx is minimum grid spacing (2.5 mm), v is bulk wave speed of the material (1924 m/sec), and C is the CFL factor which must be less than or equal to one. Although standard MPM calculations converge with $C \leq 1$, we found that the oscillating block problem required a smaller time step for accurate frictional heating results. Figure 7 plots total frictional work as a function of C calculated by either first order heating, Eq. (31), or second order heating, Eq. (34) or (36). Although we do not have an analytical solution for expected heat generation, the two methods should converge to the same results as C approaches zero, and indeed they do. The second order heat analysis, however, converges faster. A second order heating analysis gets good results for

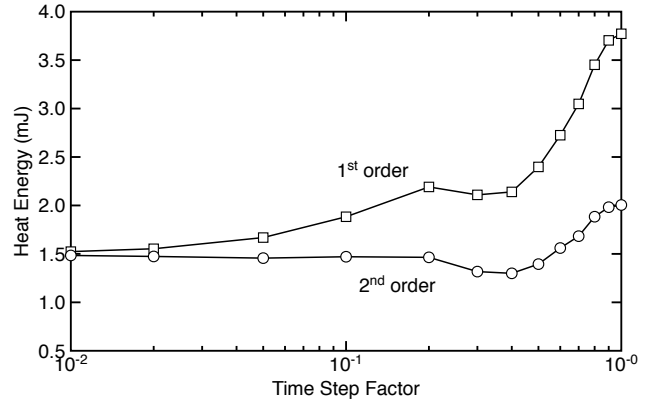


Fig. 7 Total work of friction in the oscillating block problem as a function of the CFL time step factor. The two curves are for frictional heat found by first or second order methods.

$C < 0.5$ while accurate results using first order heating would require the time step to be 10 times smaller.

For an important practical problem, we looked at high-rate loading of PBX. A concern in these materials is that frictional heating during rapid loading might cause enough heating to cause detonation [6]. A 2D simulation for the idealized geometry we used is shown in Fig. 8. The PBX material is confined on three walls and the fourth wall was pushed at 3.07 m/sec (which was 1% of the bulk wave speed for the HMX particles in the PBX). The total modeled area was $240 \times 240 \mu\text{m}$ meaning the loading strain rate was $2.6 \times 10^4 \text{ sec}^{-1}$. The explosive particles were HMX particles and represented by elliptical particles shown in various shades of gray. The remaining space in white was filled with a plastic binder. Both materials were modeled with a Mie-Grüneisen pressure response [18] and an elastic shear response. The HMX added elastic-plastic yielding in shear. The material properties are given in Table 1. The background mesh had $4 \times 4 \mu\text{m}$ cells each filled with four $2 \times 2 \mu\text{m}$ particles. As indicated by the shades of gray for HMX particles, the particles were modeled as different materials, albeit with identical properties. This approach allows MPM to recognize them as different materials and therefore implement frictional contact between particles or between particles and binder. The locations where three materials meet require approximate methods for handling more than two materials on a node. These different contacting surfaces could be modeled with different friction laws, but for this example all contact used Coulomb friction with $\mu = 0.3$. Unlike previous sliding simulations, it was not possible to specify the contacting normals in the PBX simulations. We instead found the normal using the maximum gradient approach (MVG) and used the displacement option when calculating surface separation, δ_n (Eqs. (9) and

Table 1 Material properties used for the PBX simulations of HMX in a plastic binder. The material models used a Mie-Grüneisen pressure response [18] and elastic shear response. The HMX material had elastic-plastic yielding at the given yield stress.

Property	HMX	Binder
C_0 (m/sec)	3070	1222
S_1	1.79	3.4463
S_2	—	-3.447
γ_0	0.7	0.583
Density (g/cm ³)	1.891	0.935
Shear Modulus (GPa)	4.81	1.0
Yield Stress (MPa)	260	—
C_v (J/(kg-K))	950	2280

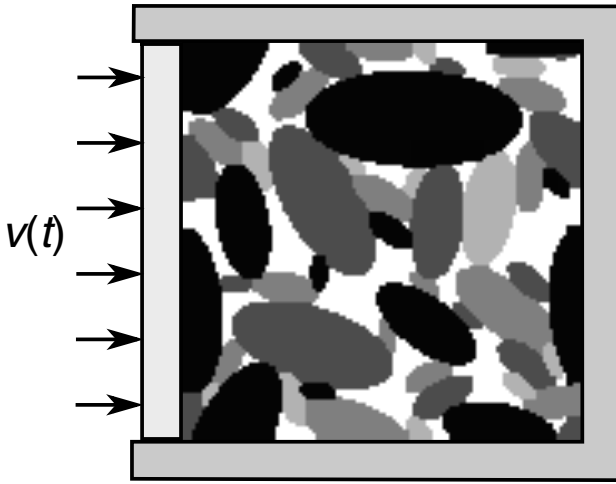


Fig. 8 The rapidly loaded PBX problem. The dimensions used were $240 \times 240 \mu\text{m}$. The gray elliptical blocks are HMX material and the white regions binder material. The rectangles on the edges are rigid materials; three walls were fixed and the fourth wall was moved at a constant displacement rate of $v(t) = 3.07 \text{ m/sec}$. The background grid used $4 \mu\text{m}$ cells.

(10)). The MPM model was constructed by reading an image of the PBX material, which was stored in a BMP files with 120×120 resolution and converting each pixel in the image to a material point in the simulation based on gray value in the image.

By running simulations with randomly generated structures, we could look at role of that structure in friction heating. Here we kept the structure constant to look at the difference between first and second order heating calculations — the results are in Fig. 9. The specimen was loaded for $15.8 \mu\text{s}$ corresponding to a final strain of 20%. The curves plot the total friction work during the loading as a function of the CFL factor for either first order heating or second order heating. Again, the second order heating converges faster meaning that accurate simulations are possible with a larger time step. This simulation is likely similar to the oscil-

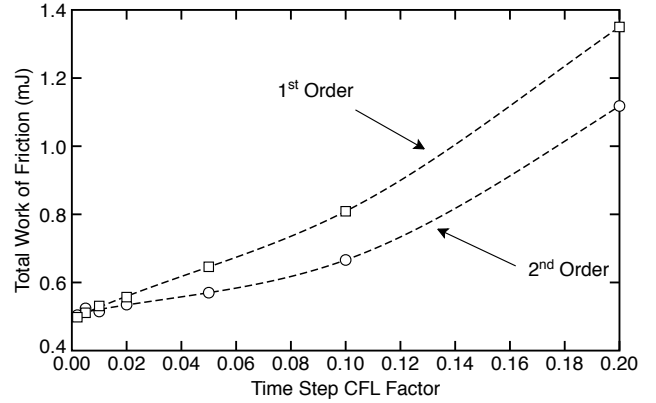


Fig. 9 Total work of friction for a specific PBX specimen loaded at a constant rate for $15.8 \mu\text{s}$. The two curves are for frictional heat found by first or second order methods.

lating block problem. During rapid loading, the specimen will encounter stress waves reflecting between the walls. The stress waves are likely to cause slip and stick conditions. Furthermore, the large strains lead to high contact forces. In other words, the use of second order heating is most noticeable for problems involving oscillations and high contact forces.

4 Conclusions

We have generalized previous MPM methods for contact to handle any friction law. Depending on the form of the friction law, the contact calculations may require extra work. For example, a friction law with an adhesion term will depend on a calculated contact area, A_c , at each node. Implementation of velocity-dependent friction laws needs to be derived from the final sliding velocity. This complication means that most friction laws will require numerical methods to find sliding force. All laws tested agreed very well with predictions in simple problems.

We derived first and second order approaches to frictional heating based on treating the sliding velocity as constant or on including effects of sliding acceleration and the possibility of slip and stick within a time step. For many problems, the second order heating terms are negligible, but for at least one real-world problem (high-rate loading of PBX), the addition of second order heating improved temporal convergence. Because the numerical effort needed to include second order heating is negligible compared to savings in temporal convergence, our recommendation is to use second order heating for all MPM contact simulations with frictional heating.

Compliance with Ethical Standards

Funding: This work was funded by a Small Business Technology Transfer (STTR) contract with Eglin Air Base Contract #FA8651-15-M-0298.

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

References

1. Akhavan, J.: The Chemistry of Explosives. Royal Society of Chemistry (2011)
2. Bardenhagen, S.G., Brackbill, J.U., Sulsky, D.: The material point method for granular materials. *Computer Methods in Applied Mechanics and Engineering* **187**, 529–541 (2000)
3. Bardenhagen, S.G., Brackbill, J.U., Sulsky, D.L.: Shear deformation in granular material. In: *Proceedings of the 11th International Detonation Symposium*, pp. 528–536. Snowmass Village, Colorado (1998)
4. Bardenhagen, S.G., Guilkey, J.E., Roessig, K.M., Brackbill, J.U., Witzel, W.M., Foster, J.C.: An improved contact algorithm for the material point method and application to stress propagation in granular materials. *Computer Modeling in Engineering & Sciences* **2**, 509–522 (2001)
5. Bardenhagen, S.G., Kober, E.M.: The generalized interpolation material point method. *Computer Modeling in Engineering & Sciences* **5**, 477–496 (2004)
6. Field, J.E.: Hot spot ignition mechanisms for explosives. *Accounts of Chemical Research* **25**(11), 489–496 (1992). DOI 10.1021/ar00023a002. URL <http://dx.doi.org/10.1021/ar00023a002>
7. Guo, Y., Nairn, J.A.: Three-dimensional dynamic fracture analysis in the material point method. *Computer Modeling in Engineering & Sciences* **16**, 141–156 (2006)
8. Homel, M., Herbold, E.B.: Field-gradient partitioning for fracture and frictional contact in the material point method. *Int. J. for Numerical Methods in Engineering* **109**(7), 1013–1044 (2016)
9. 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://dx.doi.org/10.1002/nme.2981>
10. Lemiale, V., Hurmane, A., Nairn, J.A.: Material point method simulation of equal channel angular pressing involving large plastic strain and contact through sharp corners. *Computer Modeling in Eng. & Sci.* **70**(1), 41–66 (2010)
11. Nairn, J.A.: Material point method calculations with explicit cracks. *Computer Modeling in Engineering & Sciences* **4**, 649–664 (2003)
12. Nairn, J.A.: Numerical implementation of imperfect interfaces. *Computational Materials Science* **40**, 525–536 (2007)
13. Nairn, J.A.: Modeling of imperfect interfaces in the material point method using multimaterial methods. *Computer Modeling in Eng. & Sci.* **92**(3), 271–299 (2013)
14. Nairn, J.A., Guilkey, J.E.: Axisymmetric form of the generalized interpolation material point method. *Int. J. for Numerical Methods in Engineering* **submitted** (2013)
15. Pan, X.F., Xu, A.G., Zhang, G.C., Zhang, P., Zhu, J.S., Ma, S., Zhang, X.: Three-Dimensional Multi-mesh Material Point Method for Solving Collision Problems. *Communications in Theoretical Physics* **49**, 1129–1138 (2008). DOI 10.1088/0253-6102/49/5/09
16. Sewell, T.D., Menikoff, R., Bedrov, D., Smith, G.D.: A molecular dynamics simulation study of elastic properties of hmx. *J. Chem. Phys.* **119**, 7417–7426 (2003)
17. Sulsky, D., Chen, Z., Schreyer, H.L.: A particle method for history-dependent materials. *Comput. Methods Appl. Mech. Engrg.* **118**, 179–186 (1994)
18. Wilkens, M.L.: *Computer Simulation of Dynamic Phenomena*. Springer-Verlag, New York (1999)
19. Williams, J.G., Patel, Y., Blackman, B.R.K.: A fracture mechanics analysis of cutting and machining. *Engineering Fracture Mechanics* **77**(2), 293–308 (2010)

Appendix

For the sliding block problem in Fig. 2 the sliding traction for $t > t_2$ is:

$$S = \frac{RN(t - t_2)}{t_s} \quad (46)$$

where $t_s = t_f - t_2$ is total time for increasing tangential force. The sliding will start when this traction reaches the zero-velocity sliding force using the static coefficient of friction ($\mu_s N + S_a$), which occurs when

$$t_i = t_2 + \frac{1}{R} \left(\mu_s + \frac{S_a}{N} \right) t_s \quad (47)$$

Once sliding begins, the total force in the x direction is $F_t - lb((\mu_d + k v(t))N + S_a)$, which leads to an accel-

eration of:

$$a(t) = \frac{dv(t)}{dt} = \alpha(t - t_i) + \beta - \gamma v(t) \quad (48)$$

where

$$\alpha = \frac{RN}{\rho h t_s}, \quad \beta = \frac{(\mu_s - \mu_d)N}{\rho h}, \quad \text{and} \quad \gamma = \frac{kN}{\rho h} \quad (49)$$

Solving the above differential equation for $v(t)$, the sliding velocity is

$$v(t) = \begin{cases} \frac{1}{2}\alpha T^2 + \beta T & \text{for } k = 0 \\ \psi T + \phi(1 - e^{-\gamma T}) & \text{for } k \neq 0 \end{cases} \quad (50)$$

where

$$T = t - t_i, \quad \psi = \frac{\alpha}{\gamma}, \quad \text{and} \quad \phi = \frac{\beta - \psi}{\gamma} \quad (51)$$

The total friction work is

$$Q(T) = lb \int_0^T ((\mu_d + k v(t))N + S_a) v(t) dt \quad (52)$$

For $k = 0$, the result is

$$\frac{Q(T)}{lb} = (\mu_d N + S_a) \left[\frac{1}{6}\alpha T^3 + \frac{1}{2}\beta T^2 \right] \quad (53)$$

For $k \neq 0$, the result is

$$\begin{aligned} \frac{Q(T)}{lb} = & (\mu_d N + S_a) \left[\frac{1}{2}\phi T^2 + \psi \left(T - \frac{1 - e^{-\gamma T}}{\gamma} \right) \right] \\ & + kN \left[\frac{1}{3}\psi^2 T^3 + \phi^2 \left(T - \frac{3 - 4e^{-\gamma T} + e^{-2\gamma T}}{2\gamma} \right) \right. \\ & \left. + \psi\phi \left(T^2 - \frac{2}{\gamma^2} \left(1 - (1 + T\gamma)e^{-\gamma T} \right) \right) \right] \end{aligned} \quad (54)$$