

# Modeling heat flow across material interfaces and cracks using the material point method

John A. Nairn

Accepted: July 2018

**Abstract** Heat conduction through an object with material interfaces or cracks is influenced by heat flow across those discontinuities. This paper presents a numerical particle method for modeling such heat flow coupled to computational mechanics all within the material point method (MPM). In brief, MPM models contact and cracks by extrapolating multiple velocity fields to a grid. To model interfacial heat flow, MPM should similarly extrapolate multiple temperature fields. Interfaces nodes that “see” more than one temperature field modify their heat flow to reflect interfacial physics. For example, interfaces in contact may transfer heat by perfect conduction while separated interfaces may block heat flow or cause reduced heat flow by convection. After some validation examples, two real-world examples consider cooling an ingot within a crucible where cooling causes the ingot to lose contact with the crucible walls and thermal imaging of cracks within an opaque solid.

**Keywords** Heat conduction, material point method, explicit cracks, material contact

## 1 Introduction

A straightforward addition to particle-based, material point method (MPM) modeling is to couple the mechanics analysis to thermal conduction. Coupling is done by tracking particle temperature, extrapolating temperature to a background grid, and solving the heat flow equation on the grid along with the momentum equation. Having such a feature allows MPM to model heating mechanisms caused by physical phenomena such as

volumetric expansion or compression, plastic, viscoelastic, or damage energy dissipation, or contact with friction and to fully track thermodynamic state variables (*e.g.*, entropy, enthalpy, and free energy).

Standard MPM conduction methods extrapolate particle temperature to a single temperature field, thereby solving for a global temperature field. When a problem has material interfaces or cracks, however, this single-temperature-field approach cannot model the influence of those discontinuities on heat flow. This situation is analogous to MPM contact methods. When an MPM code uses a single velocity field, the calculations can model only stick contact. Extending MPM to model contact by friction [2] or an imperfect interface [10, 11] requires use of multimaterial velocity fields. Nodes that “see” only a single material proceed by conventional MPM while “interface” nodes that see multiple materials adjust nodal momenta to reflect some modeled contact mechanics.

This paper adopts a similar approach for heat flow calculations by extrapolating multiple temperature fields. Single field nodes proceed by conventional conduction methods while interface nodes adjust nodal heat flows to model realistic heat flow across an interface. The “Heat flow equations” section first describes global conduction analysis and then describes how to modify that analysis for multiple temperature fields. The methods are applied both to multimaterial MPM and to MPM with explicit cracks [9]. The methods were verified by simple calculations across a contact interface. Applications for the method are demonstrated with two real-world examples for cooling an ingot in contact with a crucible where thermal shrinkage may cause the ingot to lose contact with the crucible walls and thermal imaging of opaque objects with internal cracks. Although MPM provides no specific advantages over other nu-

merical methods (such as finite element analysis) for solution of heat transport equations, MPM may have advantages for modeling explicit cracks [9] and complex contact mechanics [11, 14]. The methods in the paper allow MPM simulations with cracks and interfaces to account for effects of those discontinuities on heat transport.

## 2 Heat flow equations

### 2.1 Global analysis

A common addition to MPM codes is a global heat conduction option that ignores material interfaces and cracks. Despite its use in many MPM codes, a full generalized interpolation material point (GIMP) [3] derivation is rarely in the literature [15]; it is given here. The heat conduction equation is

$$\rho C \frac{\partial T}{\partial t} + \nabla \cdot \mathbf{q}(\mathbf{x}) = \dot{q}_s(\mathbf{x}) \quad (1)$$

where  $\rho$  is density,  $C$  is heat capacity (per unit mass),  $T$  is temperature,  $\mathbf{q}$  is heat flux (per unit area), and  $\dot{q}_s$  is volumetric heat source rate. For heat conduction, heat flux is  $\mathbf{q} = -\mathbf{k}\nabla T$  where  $\mathbf{k}$  is the thermal conductivity tensor. Solving this equation in the MPM weak form gives

$$\int_V \left( \rho C \frac{\partial T}{\partial t} + \nabla \cdot \mathbf{q} - \dot{q}_s(\mathbf{x}) \right) w(\mathbf{x}) dV = 0 \quad (2)$$

where  $V$  is total volume and  $w(\mathbf{x})$  is an arbitrary weighting function. Using the vector identity:

$$w(\mathbf{x})\nabla \cdot \mathbf{q}(\mathbf{x}) = \nabla \cdot (w(\mathbf{x})\mathbf{q}(\mathbf{x})) - \nabla w(\mathbf{x}) \cdot \mathbf{q}(\mathbf{x}) \quad (3)$$

and the divergence theorem, the weak form equation becomes:

$$\int_V \left( w(\mathbf{x})\rho(\mathbf{x})C(\mathbf{x}) \frac{\partial T}{\partial t} - \nabla w(\mathbf{x}) \cdot \mathbf{q}(\mathbf{x}) - \dot{q}_s(\mathbf{x})w(\mathbf{x}) \right) dV + \int_{\delta V} w(\mathbf{x})\mathbf{q}(\mathbf{x}) \cdot \hat{n} dS = 0 \quad (4)$$

where  $\delta V$  is the border of  $V$  and  $\hat{n}$  is a surface normal vector. By GIMP methods [3], particle quantities are expanded in a particle basis to get:

$$\rho(\mathbf{x})C(\mathbf{x}) \frac{\partial T}{\partial t} = \sum_p \rho_p C_p \frac{\partial T_p}{\partial t} \chi_p(\mathbf{x}) \quad (5)$$

$$\dot{q}_s(\mathbf{x}) = \sum_p \dot{q}_{s,p} \chi_p(\mathbf{x}) \quad (6)$$

$$\mathbf{q}(\mathbf{x}) = \sum_p \mathbf{q}_p \chi_p(\mathbf{x}) \quad (7)$$

where subscript  $p$  denotes a particle property (note:  $C_p$  is particle heat capacity and *not* constant-pressure

heat capacity) and  $\chi_p(\mathbf{x})$  is a particle basis function for particle  $p$  (which is typically 1 within the particle's domain and zero elsewhere [3]). Next expand the weight function and its gradient using standard, isoparametric grid shape functions,  $N_i(\mathbf{x})$ :

$$w(\mathbf{x}) = \sum_i w_i N_i(\mathbf{x}) \text{ and } \nabla w(\mathbf{x}) = \sum_i w_i \nabla N_i(\mathbf{x}) \quad (8)$$

where  $w_i$  are nodal values of  $w(\mathbf{x})$  on the grid. After substituting all expansions, the weak form equation becomes

$$\begin{aligned} & \sum_i \int_{\delta V} w_i N_i(\mathbf{x}) \mathbf{q}(\mathbf{x}) \cdot \hat{n} dS = \\ & + \int_V \left\{ \sum_i \sum_p [w_i \nabla N_i(\mathbf{x}) \cdot \mathbf{q}_p \chi_p(\mathbf{x})] \right. \\ & + \sum_i \sum_p \dot{q}_{s,p} \chi_p(\mathbf{x}) w_i N_i(\mathbf{x}) \\ & \left. - \sum_i \sum_p \chi_p(\mathbf{x}) w_i N_i(\mathbf{x}) \rho_p C_p \frac{\partial T_p}{\partial t} \right\} dV \quad (9) \end{aligned}$$

Exploiting the fact that  $w(\mathbf{x})$  is arbitrary, this equation transforms to a system of equations for node  $i$ :

$$\begin{aligned} \sum_p M_p C_p \frac{\partial T_p}{\partial t} S_{pi} = \sum_p V_p \mathbf{q}_p \cdot \mathbf{G}_{pi} + \sum_p V_p \dot{q}_{s,p} S_{pi} \\ - \int_{\delta V} N_i(\mathbf{x}) \mathbf{q}(\mathbf{x}) \cdot \hat{n} dS \quad (10) \end{aligned}$$

where  $M_p$  and  $V_p$  are particle mass and volume and  $S_{pi}$  and  $\mathbf{G}_{pi}$  are GIMP shape functions [3]:

$$S_{pi} = \frac{1}{V_p} \int_V \chi_p(\mathbf{x}) N_i(\mathbf{x}) dV \quad (11)$$

$$\mathbf{G}_{pi} = \frac{1}{V_p} \int_V \chi_p(\mathbf{x}) \nabla N_i(\mathbf{x}) dV \quad (12)$$

Allowing particle volume and heat capacity to change each time step (superscripted with  $(n)$ ), the thermal energy on node  $i$  (with SI units J) can be defined as:

$$\tau_{Ti}^{(n)} = \sum_p M_p C_p^{(n)} T_p^{(n)} S_{ip}^{(n)} \quad (13)$$

The MPM thermal conduction equation becomes:

$$\frac{d\tau_{Ti}^{(n)}}{dt} = q_i^{(n)} + q_{i,q}^{(n)} \quad (14)$$

$$\text{where } q_i^{(n)} = \sum_p V_p^{(n)} \left( \mathbf{q}_p \cdot \mathbf{G}_{ip}^{(n)} + \dot{q}_{s,p} S_{ip}^{(n)} \right) \quad (15)$$

$$q_{i,q}^{(n)} = - \int_{\delta V} N_i(\mathbf{x}) \mathbf{q}(\mathbf{x}) \cdot \hat{n} dS \quad (16)$$

are nodal thermal energy flows (with SI units J/sec or Watts).  $q_i^{(n)}$  is internal thermal flow while  $q_{i,q}^{(n)}$  is thermal flow at boundaries due to flux boundary conditions. Unlike the corresponding MPM momentum calculation

for force that depends on tracked particle stress, particle heat flux is not tracked on the particle; instead it is calculated on each time step using shape function gradients:

$$\mathbf{q}_p = -\mathbf{k}_p^{(n)} \nabla T_p^{(n)} = -\mathbf{k}_p^{(n)} \sum_i T_i^{(n)} \mathbf{G}_{ip}^{(n)} \quad (17)$$

where  $\mathbf{k}_p$  is particle thermal conductivity and nodal temperature,  $T_i^{(n)}$ , is found from:

$$T_i^{(n)} = \frac{T_{Ti}^{(n)}}{c_i^{(n)}} \quad \text{where} \quad c_i^{(n)} = \sum_p M_p C_p^{(n)} S_{ip}^{(n)} \quad (18)$$

is nodal heat capacity (SI units J/K) extrapolated from particles. The temperature update on each node is

$$T_i^{(n+1)} = T_i^{(n)} + \frac{q_i^{(n)} + q_{i,q}^{(n)}}{c_i^{(n)}} \Delta t = T_i^{(n)} + v_{Ti}^{(n)} \Delta t \quad (19)$$

where nodal temperature “velocity” (SI units K/sec) is

$$v_{Ti}^{(n)} = \frac{q_i^{(n)} + q_{i,q}^{(n)}}{c_i^{(n)}} \quad (20)$$

Once temperature velocities are found, each particle’s temperature updates by:

$$T_p^{(n+1)} = T_p^{(n)} + v_{Tp}^{(n)} \Delta t \quad (21)$$

where  $v_{Tp}^{(n)}$  is temperature velocity extrapolated to the particle. Two options are possible. The first extrapolates the temperature velocity using

$$v_{Tp}^{(n)} = \sum_i v_{Ti}^{(n)} S_{pi}^{(n)} \quad (22)$$

while the second extrapolates heat flux and divides that result by particle heat capacity:

$$v_{Tp}^{(n)*} = \frac{1}{\rho_p C_p} \sum_i \frac{q_i^{(n)} + q_{i,q}^{(n)}}{v_i^{(n)}} S_{pi}^{(n)} \quad (23)$$

where

$$v_i^{(n)} = \sum_p V_p S_{ip}^{(n)} \quad (24)$$

is volume extrapolated to the node. These two methods are identical when all particles have the same mass, density, and heat capacity, but would differ in composite materials and when modeling phase transitions where heat capacity varies in the melting region. In general, the results seem similar, but unpublished results with phase transitions suggest the first method (extrapolation of temperature rate) is more stable. This paper uses that method (and actually either method would work because all examples used particles with the same mass, density, and heat capacity). Note that particle

temperature must be updated by extrapolating temperature *velocity* instead of temperature, to the particle. A temperature extrapolation leads to artificial conduction, even if the material’s conductivity is zero [4].

The conduction solution is easily coupled to standard MPM mechanics analysis. Coupling occurs two ways - through thermal expansion or through heat generated by various mechanisms. Thermal expansion is coupled by evaluating temperature increment on each particle for input to constitutive law calculations that account for thermal expansion. Because of the way particle temperatures update, some simulations exhibit variations in  $T_p^{(n)}$  within a cell. Despite these variations, the temperature field on the grid remains smooth and is a better description of the current temperature field. As a consequence, temperature increment on each particle is best calculated from

$$\Delta T_p = T_{g \rightarrow p}^{(n+1)} - T_{g \rightarrow p}^{(n)} \quad \text{where} \quad T_{g \rightarrow p}^{(n)} = \sum_i T_i^{(n)} S_{pi}^{(n)} \quad (25)$$

rather than from  $T_p^{(n)}$ . To make this calculation possible, particles should track both  $T_p^{(n)}$  and  $T_{g \rightarrow p}^{(n)}$ .  $T_p^{(n)}$  is used in conduction equations while  $T_{g \rightarrow p}^{(n)}$  is used to find  $\Delta T_p$  or to implement any other feature that depends on particle temperature.

Heat generated by constitutive laws may be adiabatic heating (*e.g.*, adiabatic deformation of an elastic material changes temperature by  $dT_{ad} = (\mathbf{M} \cdot d\boldsymbol{\varepsilon})/\rho$  where  $\mathbf{M}$  is the stress-temperature tensor and  $d\boldsymbol{\varepsilon}$  is the strain increment [5]) or any energy that is dissipated as heat (*e.g.*, plastic or viscoelastic energy dissipation). These heat terms can be converted to temperature changes (depending on particle’s current heat capacity), accumulated during constitutive law calculations as  $dT_{p,ad}^{(n)}$ , and then added to particle temperature during MPM particle updates. In effect, this approach is solving an adiabatic particle heat equation:

$$\rho_p C_p \frac{dT_p^{(n)}}{dt} = \dot{q}_s \quad (26)$$

with solution

$$T_p^{(n+1)} = T_p^{(n)} + \frac{\dot{q}_s \Delta t}{\rho_p C_p} = T_p^{(n)} + dT_{p,ad}^{(n)} \quad (27)$$

Other heat generation might be frictional contact at material interfaces and cracks [14] or crack tip heating during crack propagation. These heats are often found on the grid and implemented by adding to  $q_i^{(n)}$  in conduction calculations

$$q_i^{(n)}(\text{total}) = q_i^{(n)} + \sum_j q_{j,i} \quad (28)$$

where the sum is over events  $j$  that each generates heat  $q_{j,i}$  on node  $i$  (SI units Watts).

The above derivation is for Cartesian coordinates. Extension to axisymmetric MPM is trivial and details are provided in Ref. [15]. The only changes needed for axisymmetry are to use axisymmetric  $S_{pi}$  and  $G_{pi}$  shape functions and to use particle mass ( $M_p$ ) and volume ( $V_p$ ) on per-radian basis.

## 2.2 Heat flow across material interfaces

Extending MPM to handle heat flow at material interfaces requires each material to extrapolate to independent temperature fields on the nodes and then to implement contact heat flow physics at all interface nodes (defined as nodes that “see” temperature fields from more than one material). We define material specific quantities from sums that include only particles of material  $\alpha$ :

$$\tau_{Ti,\alpha}^{(n)} = \sum_{p \in \alpha} M_p C_p^{(n)} T_p^{(n)} S_{ip}^{(n)} \quad (29)$$

$$c_{i,\alpha}^{(n)} = \sum_{p \in \alpha} M_p C_p^{(n)} S_{ip}^{(n)} \quad (30)$$

$$q_{i,\alpha}^{(n)} = \sum_{p \in \alpha} V_p^{(n)} \left( \mathbf{q}_p^{(n)} \cdot \mathbf{G}_{ip}^{(n)} + \dot{q}_{s,p}^{(n)} S_{ip}^{(n)} \right) \quad (31)$$

where (importantly)  $\mathbf{q}_p^{(n)}$  is heat flow that would occur on particle  $p$  in the absence of interactions with other materials. It is calculated from the temperature field for the material type of particle  $p$ :

$$\mathbf{q}_{p \in \alpha}^{(n)} = -\mathbf{k}_p^{(n)} \sum_i T_{i,\alpha}^{(n)} \mathbf{G}_{ip}^{(n)} = -\mathbf{k}_p^{(n)} \sum_i \frac{\tau_{Ti,\alpha}^{(n)}}{c_{i,\alpha}^{(n)}} \mathbf{G}_{ip}^{(n)} \quad (32)$$

For any number of materials, global and material extrapolations are related by:

$$\tau_{Ti}^{(n)} = \sum_{\alpha} \tau_{Ti,\alpha}^{(n)} \quad \text{and} \quad c_i^{(n)} = \sum_{\alpha} c_{i,\alpha}^{(n)} \quad (33)$$

In contrast, global and material heat flows may differ ( $q_i^{(n)} \neq \sum_{\alpha} q_{i,\alpha}^{(n)}$ ) because  $q_i^{(n)}$  and  $q_{i,\alpha}^{(n)}$ 's are based on different particle heat flows ( $\mathbf{q}_p^{(n)}$  vs.  $\mathbf{q}_p^{(n)}$ ) that are calculated from different temperature field gradients.

The governing equation on node  $i$  for material  $\alpha$  is

$$\frac{d\tau_{Ti,\alpha}^{(n)}}{dt} = q_{i,\alpha}^{(n)} + \Delta q_{i,\alpha}^{(n)} + \phi_{i,\alpha}^{(n)} q_{i,q}^{(n)} \quad (34)$$

where  $\Delta q_{i,\alpha}^{(n)}$  is any additional heat flow that must be added to account for material interfaces at node  $i$  and  $\phi_{i,\alpha}^{(n)}$  is the fraction of any heat flux boundary condition on node  $i$  that must be applied to material  $\alpha$ . The temperature update for particle  $p$  of material  $\alpha$  using

temperature rates for material  $\alpha$  is:

$$T_{p \in \alpha}^{(n+1)} = T_p^{(n)} + \Delta t \sum_i \frac{q_{i,\alpha}^{(n)} + \Delta q_{i,\alpha}^{(n)} + \phi_{i,\alpha}^{(n)} q_{i,q}^{(n)}}{c_{i,\alpha}^{(n)}} S_{pi} \quad (35)$$

and the back extrapolation to find temperature increment is

$$T_{g \rightarrow p \in \alpha}^{(n)} = \sum_i T_{i,\alpha}^{(n)} S_{pi}^{(n)} \quad (36)$$

The implementation of MPM heat conduction analysis that accounts for material interfaces is now reduced to determining  $\Delta q_{i,\alpha}^{(n)}$  and  $\phi_{i,\alpha}^{(n)}$  at each interface node.

## 2.3 Contact calculations

When MPM calculations are in multimaterial mode [2, 7, 11], the standard approach to contact is to adjust momenta on all nodes in contact. In brief, whenever a node has more than one material, mechanics calculations determine if they are in contact [11]. If they are in contact, momenta are changed to reflect contact physics (*e.g.*, frictional contact [2, 7] or an imperfect interface [11]). When this mechanics analysis is coupled to heat flow analysis, these contact calculations should also calculate  $\Delta q_{i,\alpha}^{(n)}$  and  $\phi_{i,\alpha}^{(n)}$  on the same contact nodes. Furthermore, while contact forces can be skipped when interface are separated, heat flow calculations are needed for interfaces both in contact and separated. Determination of contact or separation can be used to implement different heat flow characteristics for each state. For example, interfaces in contact might transfer heat by conduction, while separated interfaces might use convection.

To start, consider perfect conduction at an interface such that multimaterial conduction analysis should revert to the global analysis that ignores interfaces. In this limit, the particle updates must be identical. Equating Eq. (21) to Eq. (35) and solving for perfect conduction heat flow change,  $\Delta q_{i,\alpha}^{*(n)}$ , gives

$$\Delta q_{i,\alpha}^{*(n)} = \frac{c_{i,\alpha}^{(n)}}{c_i^{(n)}} q_i^{(n)} - q_{i,\alpha}^{(n)} + \left( \frac{c_{i,\alpha}^{(n)}}{c_i^{(n)}} - \phi_{i,\alpha} \right) q_{i,q}^{(n)} \quad (37)$$

If we assume  $\phi_{i,\alpha}^{(n)} = c_{i,\alpha}^{(n)}/c_i^{(n)}$  (*i.e.*, assume that externally applied flux is logically spread over available materials according to their heat capacity fraction on the node), this term becomes:

$$\Delta q_{i,\alpha}^{*(n)} = \phi_{i,\alpha} q_i^{(n)} - q_{i,\alpha}^{(n)} \quad (38)$$

Next consider an interface with exactly two materials –  $\alpha$  and  $\beta$  – having contact by convection with

convection coefficient  $h$  (in  $W/(m^2-K)$ ). The convection heat flow applied to each material is:

$$\Delta q_{i,\alpha}^{(n)} = ha_i \left( T_{i,\beta}^{(n)} - T_{i,\alpha}^{(n)} \right) \text{ and } \Delta q_{i,\beta}^{(n)} = -\Delta q_{i,\alpha}^{(n)} \quad (39)$$

where  $a_i$  is interfacial contact area on the grid for node  $i$ . In other words, heat flow induced by convection at an interface is added to each material and total extra heat flow is zero.

The contact area can be calculated by the same methods used when implementing MPM contact methods that depend on contact area as described in (author?) [11]; the needed area is:

$$a_i = \frac{\sqrt{2(v_{i,\alpha}^{(n)} + v_{i,\beta}^{(n)}) \min(v_{i,\alpha}^{(n)}, v_{i,\beta}^{(n)})}}{t_\perp} \quad (40)$$

where

$$v_{i,\alpha}^{(n)} = \sum_{p \in \alpha} V_p^{(n)} S_{ip}^{(n)} \quad (41)$$

is material volume extrapolated to the grid and  $t_\perp$  is an effective thickness of the contacting volume. For a regular grid with equal element sides,  $\Delta x = \Delta y = \Delta z$ ,  $t_\perp = \Delta x$  or is equal to the constant cell size. For elements with rectangular elements,  $t_\perp$  is needed to account for interfaces oriented in different grid directions. More explanation and a method for finding  $t_\perp$  are given in Ref. [11]. Importantly,  $a_i$  is an *effective* contact area that reduces to an area as a function of distance from node  $i$  to the interface. This scaling is crucial for grid independence of contact results as demonstrated in the ‘‘Results and discussion’’ section.

A robust numerical implementation must allow for three or more materials on a single node. To handle this situation, replace material  $\beta$  with a virtual material that lumps all materials besides material  $\alpha$  or  $\tau_{T_{i,\beta}}^{(n)} = \tau_{T_i}^{(n)} - \tau_{T_{i,\alpha}}^{(n)}$ ,  $c_{i,\beta}^{(n)} = c_i^{(n)} - c_{i,\alpha}^{(n)}$ ,  $T_{i,\beta}^{(n)} = \tau_{T_{i,\beta}}^{(n)} / c_{i,\beta}^{(n)}$ , and  $v_{i,\beta}^{(n)} = v_i^{(n)} - v_{i,\alpha}^{(n)}$ . Substituting into Eq. (39) and eliminating all  $\beta$  terms, a general convection flow added to material  $\alpha$  becomes

$$\Delta q_{i,\alpha}^{(n)} = ha_i \frac{T_i^{(n)} - T_{i,\alpha}^{(n)}}{1 - \phi_{i,\alpha}} \quad (42)$$

This change is applied (individually) to each material on the node. For two materials, this approach reduces exactly to Eq. (39) and total added heat flow is zero. For more than two materials, it gives a reasonable result although the sum of  $\Delta q_{i,\alpha}^{(n)}$  may not be zero.

The full contact heat flow algorithm is:

1. After updating momenta on the nodes, evaluate the velocity fields at each node with more than one material, determine whether or not they are in contact

[11], use the contact state to select contact properties (*e.g.*, assume heat flow by convection with convection coefficient  $h$  or equilibrated heat flow by conduction), and then repeat following steps 2 and 3 for each material ( $\alpha$ ) at the node ( $i$ ).

2. Find  $\Delta q_{i,\alpha}^{*(n)}$  for heat flow under the perfect conduction limit.
3. If the contact state is using equilibrated conduction, set  $\Delta q_{i,\alpha}^{(n)} = \Delta q_{i,\alpha}^{*(n)}$ . If instead it is using convection, find  $a_i$  and  $\Delta q_{i,\alpha}^{(n)}$  by Eq. (42). This heat flow, however, must not exceed conduction heat flow. If  $|\Delta q_{i,\alpha}^{(n)}| < |\Delta q_{i,\alpha}^{*(n)}|$ , then use  $\Delta q_{i,\alpha}^{(n)}$ ; otherwise set  $\Delta q_{i,\alpha}^{(n)} = \Delta q_{i,\alpha}^{*(n)}$ .
4. If mechanical analysis determines interfaces in contact by friction, the frictional sliding can be converted to heat by adding a calculated  $q_{friction}$  to  $q_{i,q}$  (*i.e.*, by adding  $\phi_{i,\alpha} q_{friction}$  to heat flow in each material’s temperature field).

Note that calculation of  $\Delta q_{i,\alpha}^{(n)}$  requires knowledge of both  $q_i^{(n)}$ , which depends on  $\mathbf{q}_p^{(n)}$ , and  $q_{i,\alpha}^{(n)}$ , which depends on  $\mathbf{q}_p^{(n)}$ . As a consequence, MPM code to implement interfacial heat flow must extrapolate both global and material temperature fields to calculate both  $\mathbf{q}_p^{(n)}$  and  $\mathbf{q}_p^{(n)}$  on each particle.

## 2.4 Heat flow across cracks

Analysis of heat flow across cracks is nearly identical to heat flow at material interfaces except separate temperature fields for each material are replaced by separate temperature fields for each side of the crack. In MPM with cracks (which is called CRAMP [9]), each particle node pair is assigned a crack velocity field,  $v(p, i) = 1$  or 0 depending whether a line from the particle to the node crosses a crack (1) or does not cross a crack (0). For a single crack, each node will have at most two velocity fields (0 and 1). The CRAMP method can be extended to handle two interacting cracks by allowing up to four velocity fields on each node (or  $v(p, i) = 0$  to 3) corresponding to lines that cross no cracks (0), one crack (1), a second crack (2), or both cracks (3) [13].

When modeling heat flow across cracks, the crack velocity field specific thermal terms become:

$$\tau_{T_{i,j}}^{(n)} = \sum_p M_p C_p^{(n)} T_p^{(n)} S_{ip}^{(n)} \delta_{j,v(p,i)} \quad (43)$$

$$c_{i,j}^{(n)} = \sum_p M_p C_p^{(n)} S_{ip}^{(n)} \delta_{j,v(p,i)} \quad (44)$$

$$q_{i,j}^{(n)} = \sum_p V_p^{(n)} \left( \mathbf{q}_p^{(n)} \cdot \mathbf{G}_{ip}^{(n)} + q_{s,p}^{(n)} S_{ip}^{(n)} \right) \delta_{j,v(p,i)} \quad (45)$$

$$v_{i,j}^{(n)} = \sum_p V_p^{(n)} S_{ip}^{(n)} \delta_{j,v(p,i)} \quad (46)$$

where  $j$  is crack temperature field  $j$  on node  $i$  and  $\delta$  is the Kronecker delta. The crack heat flow term,  $q_{i,j}^{(n)}$ , includes yet another particle heat flow,  $\mathbf{q}_p''^{(n)}$ , that is calculated each time step using

$$\mathbf{q}_p''^{(n)} = -\mathbf{k}_p^{(n)} \sum_i \frac{\tau_{T_{i,v(p,i)}}^{(n)}}{c_{i,v(p,i)}^{(n)}} \mathbf{G}_{ip}^{(n)} \quad (47)$$

In other words, it extrapolates temperature gradient from each node using the temperature appropriate for that particle-node pair.

The crack contact calculations determine if crack surfaces are in contact and heat flow calculations are applied to each crack temperature field instead of each material temperature field. Crack heat flow calculations are identical to material calculations in section 2.3 except they use crack terms instead of material terms. The particle updates in Eqs. (35) and (36) replace  $\alpha$  with the appropriate velocity field  $v(p,i)$ . To implement these calculations, the extrapolations must evaluate both  $\mathbf{q}_p^{(n)}$  and  $\mathbf{q}_p''^{(n)}$  on each particle.

## 2.5 Combining cracks and material interfaces

To combine heat flow across both cracks and material interfaces, temperature fields have to be arranged in a hierarchical structure. The approach used here was to allow each node to have multiple crack velocity fields (up to two for a single crack or up to four to handle two interacting cracks). Each crack velocity field may have 1 to  $m$  material temperature fields, where  $m$  is the number of materials in the simulation.

In this arrangement, an analysis may encounter three types of nodes requiring additional heat flow calculations — nodes with multiple materials within a single crack velocity field, nodes with multiple crack velocity fields each having only a single material, and nodes with multiple crack velocity fields containing multiple materials. The first two are handled exactly as described in sections 2.2 and 2.4. The last one requires special treatment:

1. Heat flow at material interfaces are handled first. Because materials are within a crack velocity field, these calculations replace global values ( $T_i^{(n)}$ ,  $c_i^{(n)}$ , and  $q_i^{(n)}$ ) with the corresponding crack temperature field values ( $T_{i,j}^{(n)}$ ,  $c_{i,j}^{(n)}$ , and  $q_{i,j}^{(n)}$ ) and then proceed as described in section 2.2.
2. Heat flow at cracks are handled second. The change to the crack temperature field heat flow is calculated exactly as described in section 2.4. When done, however,  $c_{i,\alpha}^{(n)} \Delta q_{i,j}^{(n)} / c_{i,j}^{(n)}$  is added to each material temperature field (*i.e.*, the added heat flow is spread

over material temperature fields according to their thermal mass fraction within the crack velocity field).

For all interface nodes, particle updates use the material temperature field (see Eq. (35)). For this update to work on nodes having multiple crack velocity fields with only a single material, the total crack heat flow in such fields,  $j$ , should be copied to that one material,  $\alpha$  (*i.e.*,  $q_{i,\alpha}^{(n)} = q_{i,j}^{(n)} + \Delta q_{i,j}^{(n)}$ ). Finally, for MPM code to implement combined interfacial and crack heat flow, it must extrapolate global, material, and crack temperature fields to calculate  $\mathbf{q}_p^{(n)}$ ,  $\mathbf{q}_p'^{(n)}$ , and  $\mathbf{q}_p''^{(n)}$  on each particle.

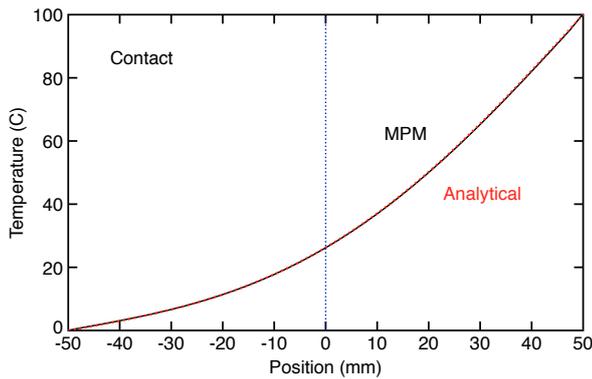
## 3 Results and discussion

The algorithm presented is fully 3D and could be added to any code that models 3D contact mechanics and/or 3D cracks. The verification examples given here were all 2D or axisymmetric. The first problem was a 2D simulation, but models a 1D problem. Consider a strip of length  $L = 100$  mm from  $x_0 = -50$  mm to  $x_1 = +50$  mm and width 20 mm with MPM background cell size of  $2.5 \times 2.5$  mm (and, like all simulations in this paper, four particles per cell). The strip was comprised of two separate isotropic materials, but with identical (arbitrarily-selected) thermal properties:  $k = 2000$  W/(m·K),  $C = 1000$  J/(kg·K), and  $\rho = 1$  g/cm<sup>3</sup>. All calculations were done in the MPM code OSParticulas [12], which fully couples mechanical and thermal calculations. Because thermal conduction is typically much slower than stress waves, these calculations used a low-modulus, zero-expansion material ( $E = 0.01$  MPa,  $\nu = 0.33$ , and thermal expansion coefficient = 0) to allow a larger time step for explicit integrations.

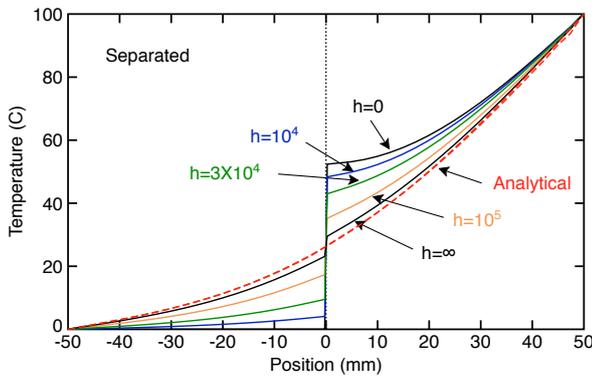
At time zero, all particles were set to temperature  $T_0 = 0^\circ\text{C}$  and boundary conditions set  $T = T_0 = 0^\circ\text{C}$  at  $x_0$  and  $T = T_1 = 100^\circ\text{C}$  at  $x_1$ . The interface was at  $x = 0$  and was modeled as conduction when in contact, but convection (with convection coefficient  $h$ ) when separated. To induce contact or separation, the two materials were either pushed together or pulled apart by 0.5 mm before the heat flow reached the interface. A Fourier series solution to this problem for conduction in the absence of an interface is [6]:

$$T(x,t) = T_0 + \Delta T \left( \xi + 2 \sum_n \frac{(-1)^n}{n\pi} e^{-\lambda_n^2 t} \sin(n\pi\xi) \right) \quad (48)$$

where  $\Delta T = T_1 - T_0$ ,  $\xi = (x - x_0)/L$ , and  $\lambda_n = (n\pi/L)\sqrt{k/(\rho C)}$ .



**Fig. 1** Temperature profile in a bar with two materials in contact at  $x = 0$  mm. All material points started at  $T = 0^\circ\text{C}$  and the results are after 500 ms with  $T$  set to  $0^\circ\text{C}$  at  $x = -50$  mm and  $100^\circ\text{C}$  at  $x = +50$  mm. The solid black line is the MPM result. The dotted red line is the analytical solution.



**Fig. 2** Temperature profile in a bar with two materials separated by 0.5 mm at  $x = 0$  mm. All material points started at  $T = 0^\circ\text{C}$  and results are after 500 ms with  $T$  set to  $0^\circ\text{C}$  at  $x = -50$  mm and  $100^\circ\text{C}$  at  $x = +50$  mm. The solid lines are MPM results for various values of interfacial convection coefficient  $h$  (in  $\text{W}/(\text{m}^2\text{-K})$ ). The dashed red line is the analytical solution for perfect conduction.

Figure 1 shows results after 500 ms with the interface pushed into contact. Because contact was modeled as conduction, the results should reduce to perfect conduction. The MPM solution (solid black line) is nearly identical to the analytical solution (dotted red line). Figure 2 shows the analogous results when the interface was separated. This case developed a temperature discontinuity at the interface. When  $h = 0$  no heat is transferred across the interface and all temperature rise was to the right of the interface ( $x > 0$ ). The left side material correctly stayed exactly at zero. As  $h$  increased, the discontinuity got smaller and the results approached the analytical solution for perfect conduction.

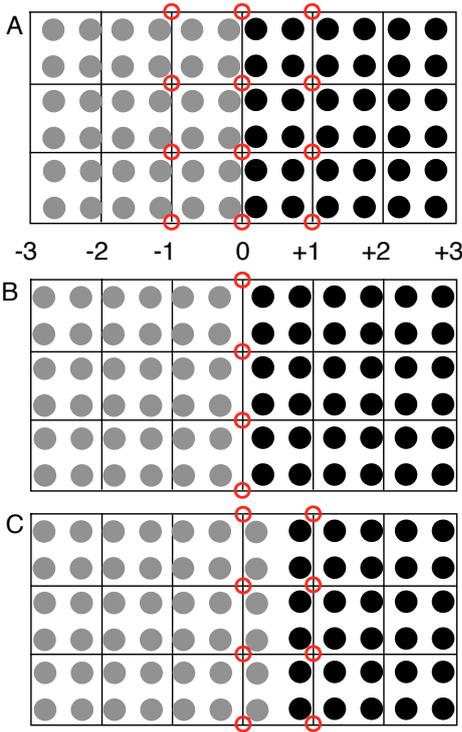
Notice that the  $h = \infty$  (which used very large  $h$ ) does not exactly match the analytical solution, which

is understood by looking at expected interface nodes. Figure 3 shows material points around an interface near grid line 0 when in contact (Fig. 3A) or separated (Fig. 3B). The circled nodes show all nodes that would be interface nodes when using GIMP shape functions (and non-GIMP methods used in early MPM papers [16] should be avoided because they activate fewer nodes). When in contact, the nodes at -1, 0 and +1 are all interface nodes that adjust their heat flow and this arrangement is able to exactly revert to perfect conduction for large  $h$ . The nodes at  $\pm 2$  are not interface nodes, but they are sufficiently far from the interface that global ( $\mathbf{q}_p^{(n)}$ ) and material ( $\mathbf{q}_p'^{(n)}$ ) heat flows would be identical. Thus, when in contact all nodes can exactly recover perfect conduction for large  $h$ .

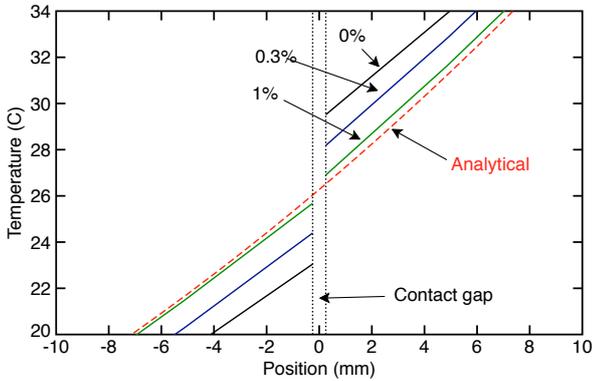
In contrast, when separated (Fig. 3B), only the nodes at 0 are interface nodes. Although those nodes can revert to conduction for very high  $h$ , the nodes at  $\pm 1$  have only a single material field. Because those nodes are close to an interface, they will have a slight discrepancy between their global ( $\mathbf{q}_p^{(n)}$ ) and material ( $\mathbf{q}_p'^{(n)}$ ) heat flows (due to use of  $T_{i,\alpha}^{(n)}$  instead of  $T_i^{(n)}$ , which differ near an interface). Because their heat flow must be based on the material temperature field (to be correct for lower  $h$ ), these nodes cannot recover the perfect conduction limit at high  $h$ .

Correcting the analysis to revert to perfect conduction limit at high  $h$  is relatively unimportant because 1) the discrepancy from perfect conduction is typically small and 2) when  $h$  is sufficiently high, the simulation can simply be run using standard conduction that ignores interfaces. Nevertheless, a simple correction seems to work. Realizing that heat flow at single-material nodes  $\pm 1$  cell from an interface may have slightly reduced heat flow, one option is to compensate for that by allowing the  $\Delta q_{i,\alpha}^{(n)}$  on interface nodes to slightly exceed the conduction limit  $\Delta q_{i,\alpha}^{*(n)}$ . Figure 4 magnifies the discontinuity region for very high  $h$  as a function of the excess heat flow allowed over conduction. As the allowed excess increased to 1% the simulations got very close to the analytical solution. Increasing beyond 1% did not improve the agreement and by 4%, the excess heat flow caused a temperature instability.

Although allowing 1% excess allowed high- $h$  simulations to approach perfect conduction, what are the consequences on that excess heat flow when  $h$  is lower? Figure 5 shows the temperature discontinuity at the interface in the separated 1D bar example as a function of  $h$  for either 1% excess heat flow (solid line) or 0% excess heat flow (dashed line). For low  $h$ , the added excess had no effect (because convection flow is always less than the conduction limit), but at high  $h$ , the 1% excess allowed the discontinuity to approach the cor-

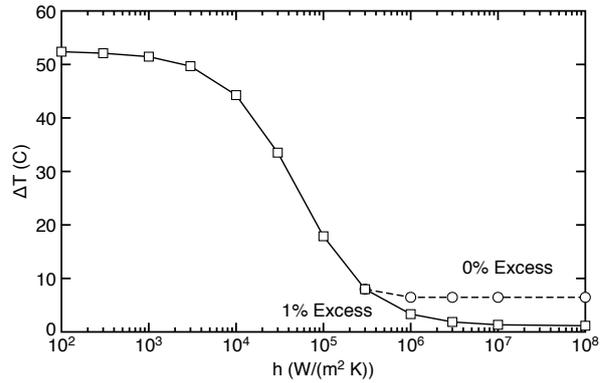


**Fig. 3** Material points (solid circles) on top of a background grid for cases when the material interface between “black” and “gray” materials is A. in contact. B. separated. C. separated but the interface is near the midpoint of background cells. The red empty circles indicate interface nodes in the MPM calculations that can adjust heat flow to account for the interface.



**Fig. 4** Temperature profile in a bar using conditions identical to Fig. 2 for very high  $h$  (effectively infinite) as a function of the excess heat flow allowed at interface nodes. The plot magnifies the region near the interface. The dashed red line is the analytical solution for perfect conduction at the interface.

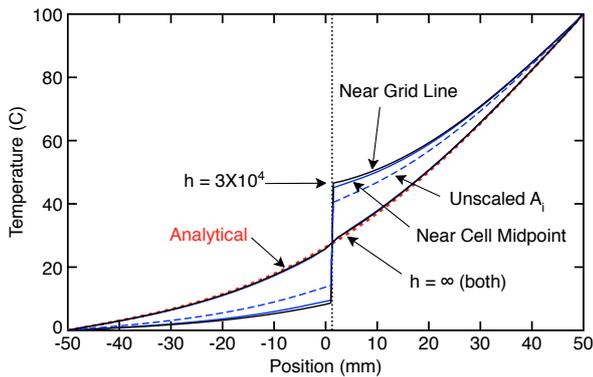
rect limit of zero. In brief, it appears reasonable to always allow 1% excess heat flow. Making this excess a simulation parameter would allow it to be adjusted for different problems if needed. The proper excess would be determined by comparing high- $h$  simulation results to a simulation that ignores interfaces.



**Fig. 5** Temperature discontinuity at the material interface for a bar using conditions identical to Fig. 2 as a function of  $h$  and for 0% (dashed line) or 1% (solid line) excess heat flow allowed at interface nodes.

All previous examples had the interface close to a grid line in the background grid, but it is important to verify heat flow calculations are independent of interface location within that grid. Figure 3C shows an interface through midpoints of background cells. The circled nodes show that twice as many interface nodes are present for such an interface compared to an interface near a grid line (*c.f.*, Fig. 3B). If contact heat flow calculations do not account for total number of interface nodes, these two interface locations would give different results. This issue is identical to contact force calculations in MPM [11] and solved by proper choice of contact area  $a_i$  in Eq. (41), which should be read as an *effective* contact area. When the interface is near a grid line (Fig. 3B),  $v_{i,\alpha}^{(n)} = v_{i,\beta}^{(n)} \approx v_i^{(n)}/2$ , where  $v_i^{(n)}$  is total volume extrapolated to node  $i$ . The *effective* contact area reduces to  $a_i = v_i^{(n)}/t_\perp$ , which is equal to the physical contact area. When the interface is through midpoints of background cells (Fig. 3C),  $v_{i,\alpha}^{(n)} \approx 7v_i^{(n)}/8$  and  $v_{i,\beta}^{(n)} \approx v_i^{(n)}/8$ , where material  $\alpha$  is the one that surrounds the node [11]. The resulting *effective* contact area,  $a_i = v_i^{(n)}/(2t_\perp)$ , now correctly assigns half the contact heat flow to each interface node in Fig. 3C where the interface is affecting twice as many nodes.

Figure 6 compares results for an interface near a grid line to one through midpoints of background cells for  $h = 3 \times 10^4 W/(m^2 K)$ . The results are nearly identical. The dashed line, however, shows results for an interface through midpoints but using the physical contact area ( $\approx v_i^{(n)}/t_\perp$ ) instead of the *effective* contact area  $a_i$ . Because the uncorrected area is about twice  $a_i$ , the added heat flow is two times too high leading to a temperature distribution corresponding to doubling of  $h$ . For realistic simulations, especially when interfaces move during calculations, grid independence requires use of  $a_i$ . The overlapping,  $h = \infty$  curves show that the results are



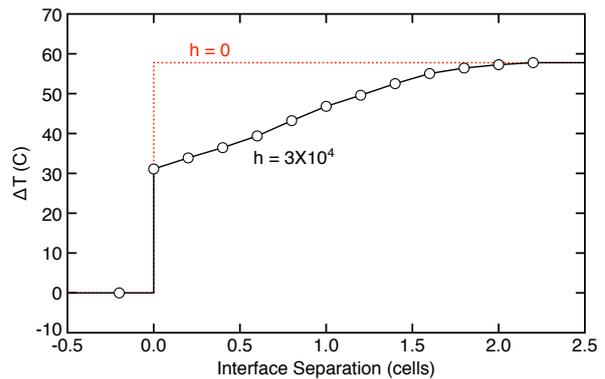
**Fig. 6** Temperature profile in a bar using conditions identical to Fig. 2 for  $h = 3 \times 10^4$  W/(m<sup>2</sup>-K) and  $h = \infty$  for simulations with the interface near a grid line or through midpoints of background cells. The dashed blue line is for an interface through midpoints of cells but using *physical* instead of *effective* contact area. The dotted red line is the analytical solution for perfect conduction at the interface.

independent of interface location and are identical to the analytical solution. Although less correction was needed when the interface was near the midpoints of cells (because of the higher number of interface nodes), each calculation agreed well with the perfect conduction limit by allowing the recommended 1% excess heat flow.

To validate heat flow analysis across cracks, simulations in Fig. 2 were repeated with the material interface being replaced by a crack within a single material or by a crack part way along a material interface in multimaterial mode. The first tested heat flow at cracks instead of material interfaces while the second tested problems that combine cracks and multimaterial mode. These new results were identical to the results in Fig. 2 (therefore no new plot is shown).

Figure 7 shows temperature discontinuity at an interface for the simulation in Fig. 2 with  $h = 3 \times 10^4$  W/(m<sup>2</sup>-K), but now as a function of interfacial separation. For interfaces in contact (separation < 0), the modeling assumed conduction and  $\Delta T$  was always zero. For separated interfaces,  $\Delta T$  increased as a function of separation and approached the  $h = 0$  limit (dotted red line) for separation of more than two cells in the background grid. A separation of more than two cells corresponds to materials that are sufficiently far apart that the simulation no longer has interface nodes. When there are no interface nodes, no calculations are done to adjust heat flow and thus heat transfer across the gap drops to zero. In other words, a separation of more than two cells results in numerical separation and thermal isolation of the materials.

Because of numerical separation, the modeling described here works best for interfaces and cracks with small separations (less than the background cell size).



**Fig. 7** Temperature discontinuity at a material interface for a bar using conditions identical to Fig. 2 with  $h = 3 \times 10^4$  W/(m<sup>2</sup>-K) as a function of separation between the two materials. The dotted red line is temperature discontinuity when  $h = 0$ .

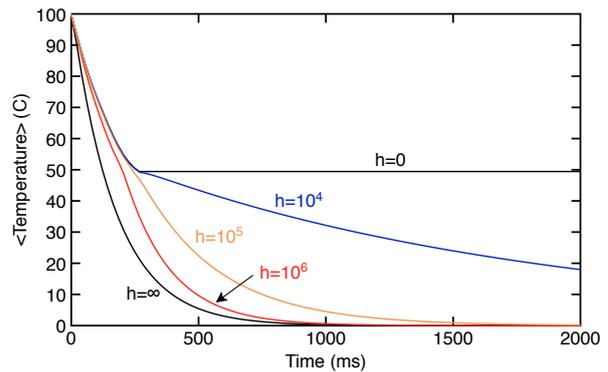
While it is physically reasonable for *effective* convection coefficient  $h$  to decrease as separation increases, the separation dependence apparent in Fig. 7 is controlled by background grid cell size and not physics of heat transport. This property could be changed by allowing the interfacial heat flow (see Eq. (42)) to be any function of  $T_i^{(n)}$ ,  $T_{i,\alpha}^{(n)}$ , and  $\delta$  (where  $\delta$  is interfacial opening displacement available in both contact [11] and explicit crack [9, 10] calculations). For example, decreased heat flow as a function of separation could be modeled by allowing  $h$  to decrease as separation increases. But no function can compensate for loss of thermal contact after “numerical separation.” Such wide separations become a problem of modeling heat flow through voids rather than across interfaces. Void modeling will require new methods. One approach might be to fill voids with a gas having different thermal conductivity properties and then modeling contact between gas and solid at material interface by the methods presented here.

Two final examples considered real-world examples that illustrate problems where interfaces or cracks might affect heat flow. Vacuum arc remelting (VAR) is a technologically-important metals process [8]. In brief, a cylindrical ingot is remelted by electrical currents into a cylindrical crucible. If the process can be sufficiently controlled, the re-solidified ingot can have fewer impurities and better properties. Because VAR is an important and expensive process, any modeling methods that can help control it would be beneficial. Full VAR modeling requires many features such as phase transitions, recrystallization kinetics, Lorentz forces induced from electric currents, and more. A possible MPM approach to this problem will be in a future publication. This paper considers only a VAR-related example for cooling a cylindrical ingot within a crucible under con-

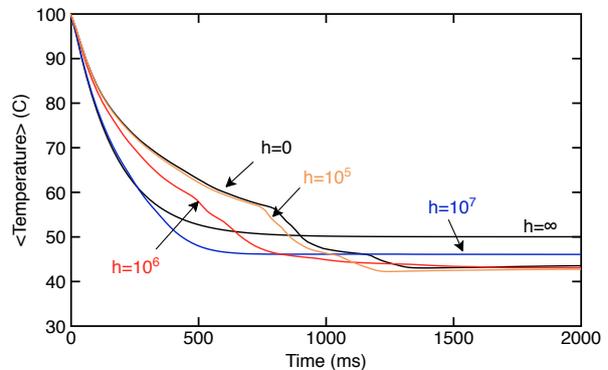
ditions where thermal contraction of the ingot causes it to separate from the crucible walls. Because VAR is conducted under vacuum, such separation is likely to affect cooling efficiency.

Consider a cylindrical ingot (radius 40 mm; length 100 mm) fully surrounded by a crucible. Mechanical and thermal properties of ingot ( $i$ ) and crucible ( $c$ ) were set to  $E_i = E_c = 1$  MPa,  $\nu_i = \nu_c = 0.33$ ,  $\alpha_i = 40 \times 10^{-6}$  C $^{-1}$ ,  $\alpha_c = 0$  C $^{-1}$ ,  $k_i = 10000$  W/(m·K),  $k_c = 50000$  W/(m·K),  $C_i = C_c = 1000$  J/(kg·K), and  $\rho_i = \rho_c = 1$  g/cm $^3$ . Most importantly, the ingot's thermal expansion coefficient was higher than the crucible's such that the ingot will pull away from the crucible as it cools. The axisymmetric simulation used  $2.5 \times 2.5$  mm cells, started with all material points at 50°C, and gradually heated them to 100°C. The heating was done to induce interfacial pressure such that all surfaces start in full contact. Gradual heating (relative to the material's wave speed) was done to minimize dynamic stress effects. After reaching 100°C, two different cooling methods were used. First, the entire crucible was immersed in a thermal bath at 0°C and crucible outer surfaces were modeled by heat flux boundary conditions on surface particles of  $q^{(n)} = h_{surf}(T - T_{res})$  (see Eq. (16)) where  $h_{surf} = 10^5$  W/(m $^2$ ·K),  $T$  is particle temperature, and  $T_{res} = 0$ °C is the reservoir temperature. The average temperatures in the ingot as a function of time after reaching 100°C for various values of  $h$  across separated ingot/crucible interfaces are plotted in Fig. 8. At first, all cooling was close to the perfect conduction limit ( $h = \infty$  curve), but lower  $h$  gave slightly slower cooling because thermal gradients induced some loss of contact at the ingot/crucible interface. After the average temperature dropped below the initial temperature of 50°C, the entire ingot separated from the crucible and the cooling rate significantly slowed. For  $h = 0$ , heat transfer stopped and therefore cooling stopped. As  $h$  increased, cooling continued but cooling rates depended strongly on  $h$ .

Figure 9 shows corresponding results where the bottom half of the crucible was immersed in a 0°C reservoir while the top half remained in a 100°C reservoir. The perfect condition limit ( $h = \infty$  curve) gave the expected result that average temperature approached 50°C with a steady state temperature gradient from cold side to hot side. The results with interfacial heat transfer, however, gave non-intuitive results. The cooling curves show regions of reduced cooling rates and regions of higher cooling rates. These regions are determined by which parts of the interface were in contact. When the hotter walls on the top half lost contact, the cooling rate was higher, but when the colder walls on the bottom half lost contact, the cooling rate slowed.



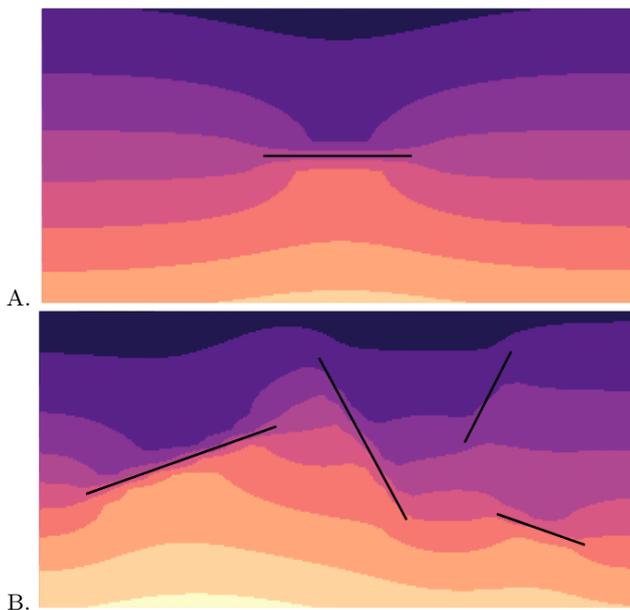
**Fig. 8** Average temperature in an ingot initially heated from 50 to 100°C followed by immersing the crucible in a 0°C reservoir as a function of cooling time for various values of  $h$  (in W/(m $^2$ ·K)).



**Fig. 9** Average temperature in an ingot initially heated from 50 to 100°C followed by the bottom half of the crucible being immersed in a 0°C reservoir while the top half remained in a 100°C reservoir as a function of cooling time for various values of  $h$  (in W/(m $^2$ ·K)).

Transitions between slow and fast cooling indicate dynamic motion of the ingot within the crucible. The role of dynamic contact between ingot and crucible on heat flow would be difficult to model by anything other than numerical methods such as those proposed here.

The second real-world example modeled cracks. One method for detecting internal cracks in solid, opaque objects is by thermal imaging [1]. In brief, an object is heated by various methods and the surface temperature is imaged using an infrared camera. If cracks affect heat flow, the number and location of cracks will alter the temperature distribution. The heat-flow methods described here could help to interpret such experiments. Figure 10 shows two examples of the effect of cracks on temperature distributions during heat flow. The  $100 \times 50$  mm isotropic material ( $E = 1$  MPa,  $\nu = 0.33$ ,  $\alpha = 0$ ,  $k = 10000$  W/(m·K),  $C = 1000$  J/(kg·K), and  $\rho = 1$  g/cm $^3$ , thickness = 1 mm) started at 0°C. The bottom edge was heated using constant heat-flux boundary condition of  $q^{(n)} = 10^7$  W/m $^2$ , the top edge

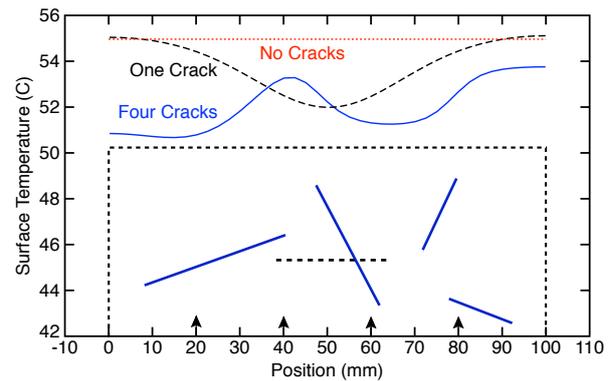


**Fig. 10** Temperature distribution in an object with A. one crack or B. four cracks after 500 ms of heat flux applied to the bottom surface. The image lightness indicates temperature from 40°C (dark) to 110°C (light) and 7°C per contour.

was cooled by convection using  $q^{(n)} = h_{surf}(T - T_{res})$  where  $h_{surf} = 10^5 \text{ W}/(\text{m}^2 \cdot \text{K})$  and  $T_{res} = 0^\circ\text{C}$ , and the sides had zero heat flux. To maximize crack effects, convective heat flow across cracks surfaces used  $h = 0$ . The temperature distribution was evaluated at the end of a 500-ms heating period. The MPM model used  $2.5 \times 2.5 \text{ mm}$  cells.

The first simulation had a single crack in the middle of the object, which, as shown in Fig. 10A, influenced the temperature field. Thermal imaging can only observe surface temperature. Figure 11 shows a drop on the top surface temperature above the crack compared to temperature distribution that would occur with no cracks. The length of the crack could be estimated from this drop, but the details would depend on distance of the crack to the surface and on the convection coefficient for heat flow across the crack surface.

The second simulation used four arbitrarily sized and placed cracks. The temperature field was significantly altered (see Fig. 10B) and the surface temperature showed highly convoluted effects of cracks on surface temperature distribution. For example, the temperature above the last crack on the right was higher than average. That crack was too far below the surface to have continued effect on surface temperature. It is unlikely the surface temperature alone can uniquely locate internal cracks, but modeling coupled with experiments could lead to derivation of a distribution of *equivalent* cracks that are consistent with observed surface temperature variations.



**Fig. 11** Surface temperature from the thermal fields in Fig. 10 for one crack (dashed black line), four cracks (solid blue line), or no crack (dotted red line).

## 4 Conclusions

This paper describes a new MPM feature for modeling heat flow across material interfaces, cracks planes, or both. The method works best for interfaces with small separations (less than one cell size in the background grid). The method has potential for analysis of real-world problems that couple mechanical response with heat flow across interfaces or cracks where temperature changes may cause interfaces or crack planes to dynamically lose or regain contact.

## Compliance with ethical standards

**Funding:** This work was made possible by the endowment for the Richardson Chair in Wood Science and Forest Products. The author also thanks Rigel Woodside, Paul King, and Kevin Gartner for helpful discussions.

**Conflict of interest:** The corresponding author states that he has no conflicts of interest.

## References

1. Almond, D.P., Peng, W.: Thermal imaging of composites. *Journal of Microscopy* **201**(2), 163–170 (2001)
2. 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 material. *Computer Modeling in Engineering & Sciences* **2**, 509–522 (2001)
3. Bardenhagen, S.G., Kober, E.M.: The generalized interpolation material point method. *Computer*

- Modeling in Engineering & Sciences **5**, 477–496 (2004)
4. Brackbill, J., Kothe, D., Ruppel, H.: FLIP: A low-dissipation, particle-in-cell method for fluid flow. *Computer Physics Communications* **48**(1), 25 – 38 (1988)
  5. Carlson, D.E.: Linear thermoelasticity. In: C. Truesdell (ed.) *Mechanics of Solids*, vol. II, pp. 297–345. Springer-Verlag, New York (1984)
  6. Guenther, R.B., Lee, J.W.: *Partial Differential Equations of Mathematical Physics and Integral Equations*. Dover Books, New York (1996)
  7. 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)
  8. Maurer, G.E.: Primary and secondary melt processing — superalloys. In: J.K. Tien, T. Caulfield (eds.) *Superalloys, Supercomposites and Superceramics*, pp. 49–97. Academic Press, New York (1989)
  9. Nairn, J.A.: Material point method calculations with explicit cracks. *Computer Modeling in Engineering & Sciences* **4**, 649–664 (2003)
  10. Nairn, J.A.: Numerical implementation of imperfect interfaces. *Computational Materials Science* **40**, 525–536 (2007)
  11. 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)
  12. Nairn, J.A.: Material point method (NairnMPM) and finite element analysis (NairnFEA) open-source software. <http://osupdocs.forestry.oregonstate.edu> (2017)
  13. Nairn, J.A., Aimene, Y.: Modeling interacting and propagating cracks in the material point method: Application to simulation of hydraulic fractures interacting with natural fractures (2017). In preparation
  14. Nairn, J.A., Bardenhagen, S.G., Smith, G.S.: Generalized contact and improved frictional heating in the material point method. *Computer Particle Mechanics* **5**(2), 285–296 (2018). DOI 10.1007/s40571-017-0168-1
  15. Nairn, J.A., Guilkey, J.E.: Axisymmetric form of the generalized interpolation material point method. *Int. J. for Numerical Methods in Engineering* **101**, 127–147 (2015)
  16. Sulsky, D., Chen, Z., Schreyer, H.L.: A particle method for history-dependent materials. *Comput. Methods Appl. Mech. Engrg.* **118**, 179–186 (1994)